Symulacja w Badaniach i Rozwoju
Vol. 7, No. 3-4/2016

Rafał KASPRZYK, Artur STACHURSKI
Wojskowa Akademia Techniczna
ul. Gen. S. Kaliskiego 2, 00-908 Warszawa
E–mail: rafal.kasprzyk@wat.edu.pl, artur.stachurski@wat.edu.pl

Koncepcja procesu zarządzania podatnościami systemów teleinformatycznych

1 Wstęp

Wraz z postępem techniki komputerowej, rozwojem Internetu i rozpowszechnieniem się usług sieciowych wciąż pojawiają się nowe zagrożenia i metody ataków na elementy cyberprzestrzeni. Spośród grona bardzo spektakularnych ataków przynoszących ogromne straty wymienić można między innymi:

- cyberataki przeprowadzone na infrastrukturę rządową Estonii (2007 r.) [4];
- ataki sabotażowe skierowane przeciwko irańskiemu programowi nuklearnemu przy wykorzystaniu zaawansowanego oprogramowania złośliwego: *Stuxnet* (2010 r.) [13], *Duqu* (2011 r.) [1], *Flame* (2012 r.) [3];
- cyberataki typu ransomware *WannaCry* i *Petya* (2017 r.), m.in. na instytucje rządowe, banki i firmy prywatne, przeprowadzane w celu wyłudzenia pieniędzy [23].

Szeroko rozumiane cyberbezpieczeństwo zależy tak od możliwości zarządzania ryzykiem naruszania bezpieczeństwa systemów teleinformatycznych, jak również od zarządzania incydentami w systemach teleinformatycznych. Zapewnienie odpowiedniego poziomu cyberbezpieczeństwa jest procesem cyklicznym, który wymusza konieczność ciągłej identyfikacji cyberzagrożeń i podatności, analizowania incydentów oraz wdrażania metod minimalizowania ryzyka potencjalnego ataku w kolejnym cyklu. Aby proces ten był efektywny, niezbędne jest posiadanie narzędzi, które automatyzują wybrane czynności związane z zarządzaniem bezpieczeństwem systemów teleinformatycznych.

Artykuł przedstawia koncepcję procesu zarządzania podatnościami, będącego elementem procesu zarządzania ryzykiem naruszania bezpieczeństwa systemów teleinformatycznych. Jednym z istotnych założeń proponowanego rozwiązania jest wykorzystanie wybranych standardów z rodziny SCAP (ang. *Security Content Automation Protocol*), które pozwalają zautomatyzować proces zarządzania bezpieczeństwem systemu teleinformatycznego.

2 Klasyfikacja podatności systemów teleinformatycznych

Podatność można rozumieć jako wadę systemu, która pozwala atakującemu naruszyć jego bezpieczeństwo, a co za tym idzie, wykorzystać go do wielu celów, zazwyczaj powodujących negatywne skutki dla właściciela systemu [8]. Obecnie w literaturze wyróżnić można wiele klasyfikacji podatności. Wśród nich najczęściej spotykany jest podział związany ze stopniem krytyczności podatności (ang. *vulnerability severity*) [11]. Podział ten wskazuje między innymi, że pewien podzbiór podatności może być

stosunkowo łatwy do wykorzystania przez atakującego i dać mu wiele możliwości naruszania bezpieczeństwa systemu. Z drugiej strony jednak możemy się spotkać z podatnościami, których próba wykorzystania w celu przeprowadzenia ataku na system będzie wymagać od atakującego wiele wysiłku, np. stosowania technik inżynierii społecznej.

Amerykański instytut NIST (*National Institute of Standards and Technology*), zajmujący się między innymi badaniem podatności systemów teleinformatycznych na zagrożenia, wyodrębnił trzy zasadnicze obszary podatności [8]:

- *podatności związane z lukami w oprogramowaniu* – spowodowane błędem w projektowaniu lub implementacji oprogramowania, zarówno niezamierzonym, jak i zamierzonym;
- *podatności związane z konfiguracją systemu* – niedoskonałości konfiguracji systemu, poprzez które atakujący może uzyskać do niego dostęp;
- *podatności związane z niepoprawnym użytkowaniem systemu* – niewłaściwe wykorzystanie funkcji oprogamowania, prowadzące do naruszenia bezpieczeństwa systemu, np. odebranie za pomocą komunikatora zainfekowanego pliku przesłanego przez atakującego.

3 Standaryzacja bezpieczeństwa systemów teleinformatycznych

Projektując systemy teleinformatyczne, wiele organizacji decyduje się na tworzenie autorskich rozwiązań związanych z zarządzaniem bezpieczeństwem systemów teleinformatycznych, bazujących najczęściej na sprawdzonych i rozpoznawalnych standardach. Jednym z podejść jest wykorzystanie standardów z rodziny SCAP.

SCAP (ang. *The Security Content Automation Protocol*) to metoda standaryzacji procesów i środków wykorzystywanych do sformalizowanego, jednolitego opisu różnych aspektów bezpieczeństwa systemów teleinformatycznych: polityki bezpieczeństwa, konfiguracji, podatności, listy oprogramowania złośliwego oraz reguł automatycznych testów. Obecna wersja standardu SCAP posiada numer 1.2 i została wydana we wrześniu 2011 roku, w publikacji NIST o numerze SP800-126r2. W skład standardu wchodzi 11 specyfikacji [2, 15]:

- Języki – pozwalają na sformalizowaną reprezentację polityk bezpieczeństwa, mechanizmów testów bezpieczeństwa oraz wyników tych testów:
  - OVAL (*Open Vulnerability and Assessment Language*),
  - XCCDF (*Extensible Configuration Checklist Description Format*),
  - OCIL (*Open Checklist Interactive Language*).
- Formaty raportów– pozwalają na sformalizowany opis danych przetworzonych przez systemy zarządzające procesami bezpieczeństwa:
  - ARF (*Asset Reporting Format*),
  - AI (*Asset Identification*).
- Nazewnictwo, słowniki – określają standardowy format nazw i słowników artefaktów związanych z bezpieczeństwem teleinformatycznym:
  - CPE (*Common Platform Enumeration*),

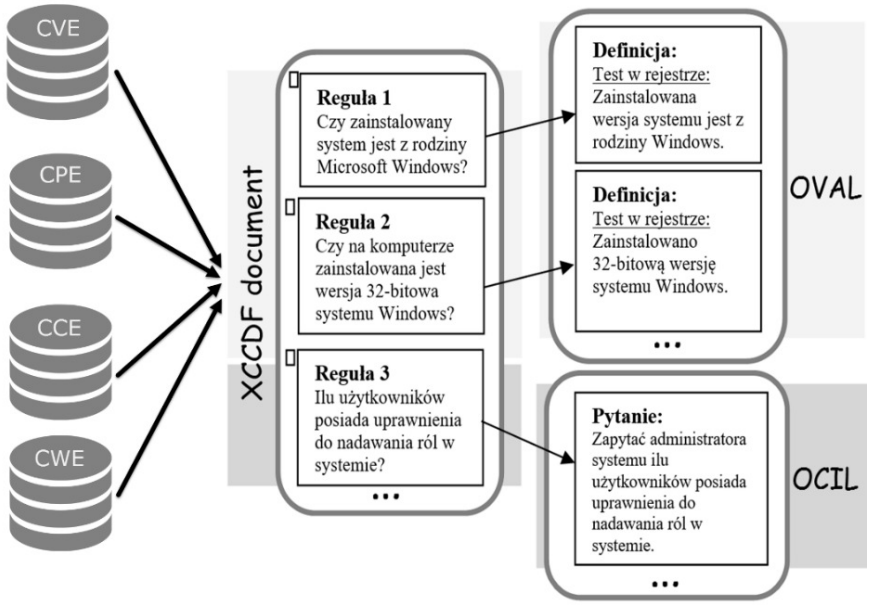
  - CCE (*Common Configuration Enumeration*),
  - CVE (*Common Vulnerabilities and Exposures*).
- Skale i mechanizmy porównawcze – opisują sposób ewaluacji cech charakterystycznych podatności, umożliwiający ocenę ryzyka naruszenia bezpieczeństwa z wykorzystaniem danej podatności:
  - CVSS (*Common Vulnerability Scoring System*),
  - CCSS (*Common Configuration Scoring System*).
- Integralność – opisuje model zapewniania integralności danych wyrażonych zgodnie z wyżej opisanymi notacjami:
  - TMSAD (*Trust Model for Security Automation Data*).

Prezentowana koncepcja, poza wymienionymi standardami z rodziny SCAP, wykorzystuje następujące specyfikacje: słownik CWE (ang. *Common Weakness Enumeration*) oraz systemy scoringowe CWSS (ang. *Common Weakness Scoring System*) i CMSS (ang. *Common Misuse Scoring System*). Wybrane standardy wchodzące w skład zaproponowanej koncepcji procesu zarządzania podatnościami systemów teleinformatycznych oraz przykład ich użycia zostały opisane w kolejnych rozdziałach. Wykorzystanie wspomnianych standardów ma na celu przede wszystkim umożliwienie wykorzystania istniejących źródeł danych o zidentyfikowanych podatnościach oraz ich efektywną wymianę z sojuszniczymi systemami do zarządzania bezpieczeństwem teleinformatycznym.

4 Proces identyfikacji podatności w systemach teleinformatycznych

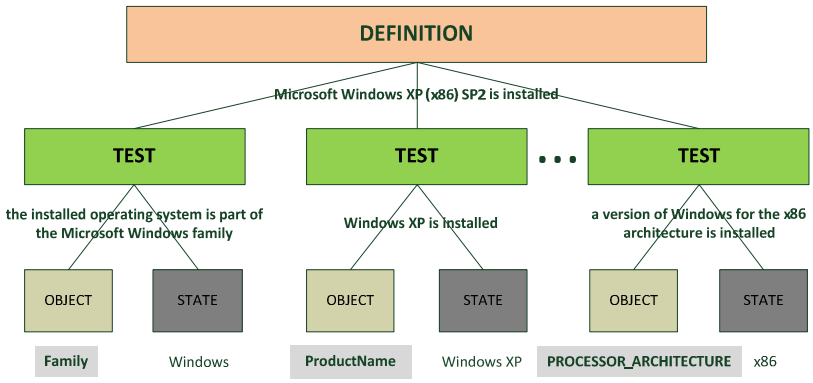
Zastosowanie standardów z rodziny SCAP pozwala zautomatyzować proces weryfikacji założeń polityki bezpieczeństwa, co jest szczególnie istotne dla systemów teleinformatycznych składających się z dużej liczby serwerów i stacji klienckich. Proces weryfikacji opiera się na tzw. kwestionariuszach bezpieczeństwa (ang. *security checklists*) umożliwiających wykrycie występujących odchyleń od założonej polityki bezpieczeństwa [19]. Kwestionariusze bezpieczeństwa tworzone są za pomocą języka XCCDF, bazującego na XML, z wykorzystaniem słowników CVE, CCE, CPE, CWE [19]. Kwestionariusze bezpieczeństwa są podstawą do opracowania, a następnie wykonania dwóch rodzajów testów (Rys. 1):

- ***Testy manualne*** – zdefiniowane w języku OCIL, polegające na przeprowadzeniu ankiety, wymagającej zazwyczaj interakcji z człowiekiem lub wglądu do danych na temat systemu teleinformatycznego;.
- ***Testy automatyczne*** – zdefiniowane w języku OVAL, wykonywane przy użyciu dedykowanego oprogramowania.



Rys. 1. XCCDF - przykładowy przypadek użycia; źródło: opracowanie własne
Fig. 1. Sample XCCDF use case scenario; source: own elaboration

Standard OVAL bazuje na otwartych źródłowych zasobach definiujących konfiguracje uznane za poprawne, co jest podstawą weryfikacji stanu badanego systemu teleinformatycznego. Definicje testów w języku OVAL wskazują ścieżkę do weryfikowanego elementu systemu (np. ścieżkę w rejestrze, ścieżkę do pliku itp.) oraz jego wartość „wzorcową". Testy sprawdzają między innymi wersje zainstalowanych aplikacji, bibliotek, jak i obecność wymaganych łatek, w wyniku czego możliwe jest stwierdzenie podatności systemu na dany rodzaj zagrożenia lub ataku. Przeprowadzenie testów polega na porównaniu wartości odczytanej z badanego systemu z wartością „wzorcową" i zwróceniu wyniku „prawda" lub „fałsz". W celu wykonania takiego badania wykorzystuje się dodatkowe oprogramowanie, które pozwala na interpretowanie testów w języku OVAL [9]. Przykładowe zobrazowanie definicji testu w języku OVAL, sprawdzającej, czy badany system to Windows XP, w architekturze 32-bitowej, zawierający dodatek SP2, zostało przedstawione na rysunku 2.

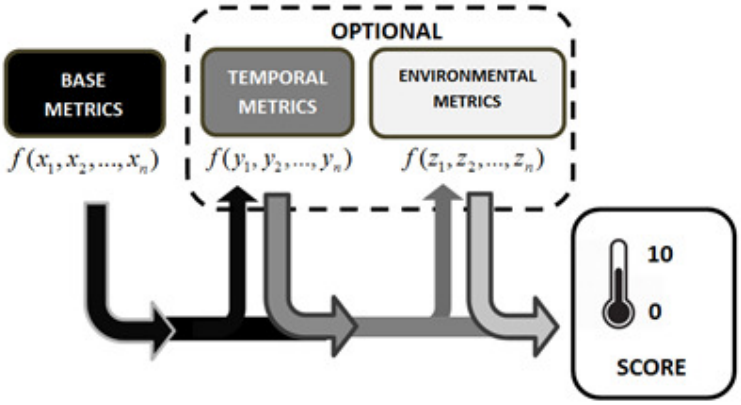


Rys. 2. Struktura przykładowej definicji testu OVAL; źródło: opracowanie własne
Fig. 2. A structure of sample OVAL definition; source: own elaboration

5 Metoda oceny krytyczności zidentyfikowanych podatności

Posiadając wyodrębniony zbiór potencjalnych podatności można określić ich krytyczność. Służą do tego następujące dedykowane mechanizmy i skale porównawcze, nazywane również systemami scoringowymi, które zintegrowane są z definicjami podatności występującymi w słownikach CVE, CWE oraz CCE:

- ***CVSS*** (*Common Vulnerability Scoring System*) – mechanizm przeznaczony do badania krytyczności podatności wynikających z luk w oprogramowaniu [8].
- ***CWSS*** (*Common Weakness Scoring System*) – mechanizm służący do badania krytyczności podatności wynikających z wad w oprogramowaniu. Koncepcyjnie zbliżony jest do CVSS, natomiast występują różnice w budowie oraz rodzaju przetwarzanych informacji o podatnościach [12].
- ***CCSS*** (*Common Configuration Scoring System*) – mechanizm służący do badania krytyczności podatności, wynikających z zastosowania niepoprawnej konfiguracji systemu [17].
- ***CMSS*** (*Common Misuse Scoring System*) – mechanizm służący do badania krytyczności podatności, pochodzących z niewłaściwego użytkowania oprogramowania [7].



Rys. 3 Schemat procesu scoringu; źrodlo: opracowanie własne
Fig. 3. Scoring process scheme; source: own elaboration

Podatność, będąca przedmiotem badania, określona jest wektorem, którego składowe reprezentują jej charakterystyki – nazywane miarami (ang. *Metrics*) [8]. W standardach CVSS, CCSS oraz CMSS wyróżniamy trzy grupy miar: podstawowe (ang. *Base Metrics*), zmienne w czasie (ang. *Temporal Metrics*) oraz środowiskowe (ang. *Environmental Metrics*). Miary podstawowe określają charakterystyki badanej podatności, które nie powinny ulegać zmianom w czasie oraz nie są zależne od środowiska, w którym dana podatność występuje. Miary zmienne w czasie dotyczą charakterystyk podatności, które mogą się zmieniać wraz z upływem czasu, wskazujących na przykład poziom dojrzałości kodu podatności (ang. *Exploit Code Maturity*) czy poziom zaawansowania prac nad wykonaniem łatki dla podatności (ang. *Remediation Level*). Miary środowiskowe łączą w sobie te charakterystyki podatności z miar podstawowych oraz zmiennych w czasie, które zależne są od środowiska, w którym podatność ta występuje. Warto zwrócić uwagę na to, iż zarówno miary zmienne w czasie, jak i środowiskowe nie są obligatoryjne podczas scoringu, natomiast pozwalają na bardziej szczegółowy pomiar krytyczności podatności.
W standardzie CWSS zastosowano podobne podejście do budowy wektora podatności, jednakże jest on bardziej rozbudowany niż we wspomnianych standardach oraz można zauważyć różnice w podziale miar na grupy. Dodatkowo w niektórych przypadkach CWSS umożliwia nadanie domyślnych wartości lub pominięcie wybranych składowych wektora podatności [12].

Zasada działania wspomnianych systemów scoringowych jest dla każdego standardu analogiczna. W celu wykonania scoringu wartości składowych wektora podstawiane są do wzorów równań określonych przez standard dla każdej grupy miar, a zwracany wynik mieści się w przedziale od 0 do 10 dla CVSS, CCSS, CMSS lub od 0 do 100 w przypadku CWSS. Im mniejsza wartość z obliczeń zostanie zwrócona, tym w mniejszym stopniu krytyczna jest badana podatność. Schemat koncepcyjny badania krytyczności podatności przedstawiono na rysunku 3. Wektor podatności zapisuje się

jako jako ciąg znaków. Wzory funkcji $f(x_1, x_2, \ldots, x_n)$ wykorzystywane do wyliczenia krytyczności podatność oraz wartości składowych wektora podatności dla wszystkich wspomnianych systemów scoringowych zostały ustalone przez ekspertów, posiadających odpowiednią wiedzę o podatnościach oraz ich wpływie na bezpieczeństwo systemów teleinformatycznych. Schemat wyliczania wartości metryki podstawowej ze standardu CVSS 3.0 zaprezentowano na rysunku 4 na przykładzie podatności SQL Injection dla MySQL.

MySQL Stored SQL Injection (CVE-2013-0375)

MIARA	SKRÓT	Wartość (zm. lingwistyczna)	Wartość liczbowa
Attack Vector	AV	Network	0,85
Attack Complexity	AC	Low	0,77
Priviliges Required	PR	Low	0,68 *
User Interaction	UI	None	0,85
Scope	S	Changed	-
Confidentiality Impact	C	Low	0,22
Integrity Impact	I	Low	0,22
Availability Impact	A	None	0

* Gdy dla miary **Scope** ustalono wartość „Changed" miara **Priviliges Required** dla wartości „Low" przyjmuje wartość liczbową 0,68 (0,62 w p.p.)

WEKTOR PODATNOŚCI: AV:N/AC:L/PR:L/UI:N/S:C/C:L/I:L/A:N

$F(x_1, x_2, \ldots, x_n)$

BASE SCORE = 6,4

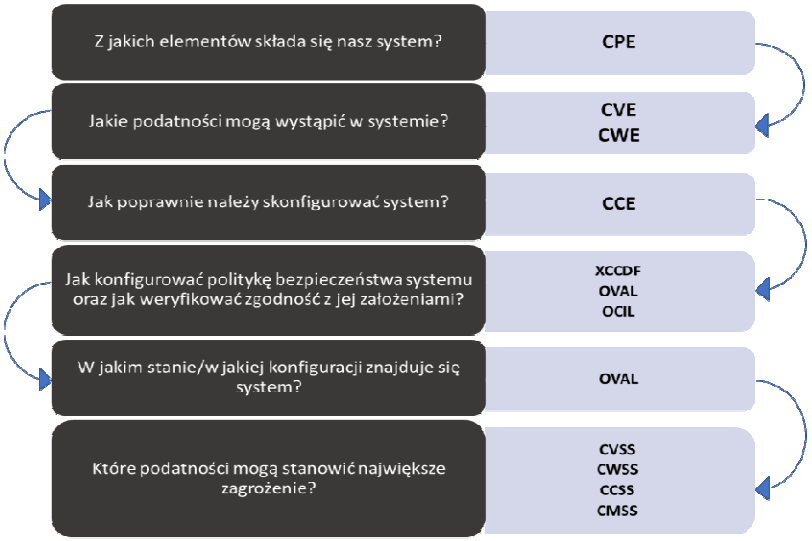
Rys. 4. Przykład wyliczenia metryki podstawowej w standardzie CVSS 3.0 dla podatności, pozwalającej na przeprowadzenie ataku SQL Injection w MySQL; źródło: opracowanie własne

Fig. 4. CVSS 3.0 Base metrics calculation for MySQL Stored Injection vulnerability; source: own elaboration

Organizacje wykorzystujące ilościowe metody badania podatności, w tym między innymi wspomniane skale i mechanizmy porównawcze, mogą dzięki nim uzyskać istotne z punktu widzenia bezpieczeństwa teleinformatycznego wymierne korzyści. Przykładowo będą w stanie w sposób wysoce prawdopodobny przewidzieć, w jaki sposób atakujący może probować naruszyć bezpieczeństwo ich systememów teleinformatycznych, a w konsekwencji będą mogli zminimalizować ryzyko ataku lub ograniczyć potencjalne skutki ataku, eliminując krytyczne podatności.

6 Koncepcja procesu zarządzania podatnościami

Przedstawiona poniżej koncepcja (Rys. 5) może przyczynić się do przeciwdziałania lub ograniczenia oddziaływania podatności, które mogłyby być wykorzystane przez atakującego do naruszenia bezpieczeństwa, a nawet bezpośrednio do przejęcia kontroli nad systemem teleinformatycznym. Procesowi zarządzania podatnościami w zaproponowanej koncepcji towarzyszy proces identyfikacji podatności na zagrożenia występujące w cyberprzestrzeni, realizowany przez instytucje zajmujące się szeroko rozumianym bezpieczeństwem w sieci. Dane o podatnościach systemów teleinformatycznych, a także inne artefakty związane z bezpieczeństwem systemów i oprogramowania gromadzone są w słownikach CPE, CVE, CWE oraz CCE.



Rys. 5. Schemat zaproponowanej koncepcji zarządzania podatnościami systemów teleinformatycznych; źródło: opracowanie własne

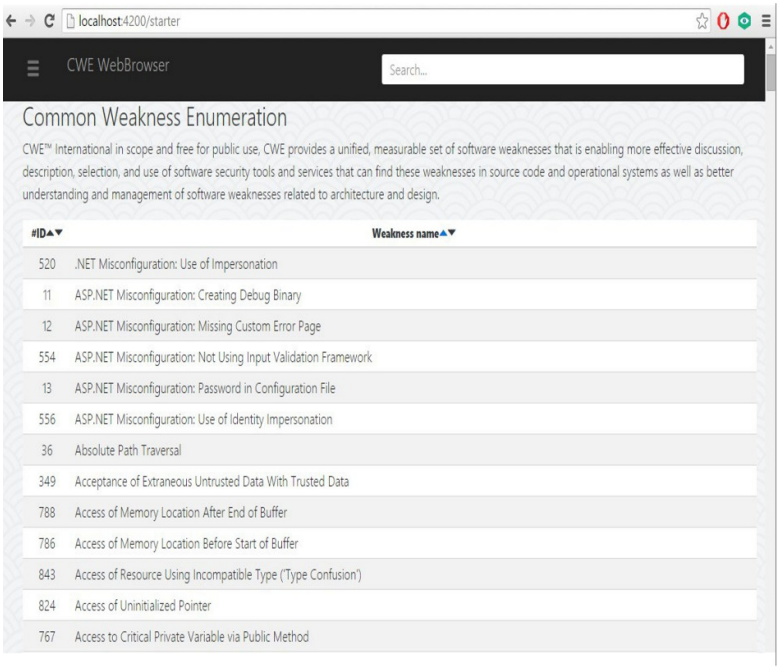
Fig. 5. A scheme of a proposed concept of vulnerability management process for IT systems; source: own elaboration

Rozpoczynając planowanie procesu zarządzania podatnościami, którego realizację przewiduje się na etapie tworzenia wymagań funkcjonalnych systemu, należy wyspecyfikować sprzęt, oprogramowanie, konfigurację systemu oraz inne atrybuty tworzące środowisko, w którym docelowy system będzie funkcjonować. Po ustaleniu, w jakim środowisku oraz w jakiej konfiguracji będzie wykorzystywany system, należy podjąć próbę ustalenia zorganizowanego zbioru reguł, czyli kwestionariuszy bezpieczeństwa (ang. *security checklists*) definiujących politykę bezpieczeństwa systemu. Ich zastosowanie pozwoli ustalić stan badanego systemu, między innymi wersje zastosowanego oprogramowania, konfigurację zabezpieczeń, dzięki czemu możliwe będzie oszacowanie zbioru jego potencjalnych podatności. W przypadku gdy zidentyfikowany został zbiór potencjalnych podatności, możliwe jest zbadanie ich stopnia krytyczności przy wykorzystaniu kalkulatorów do przeprowadzenia scoringu. Jak wspomniano w poprzednim rozdziale, dla każdego standardu z grupy skal i mechanizmów porównawczych wykorzystywane są ekspercko dobrane funkcje oraz wartości lingwistyczne składowych wektora podatności. Uzyskane w procesie scoringu wyniki pozwalają na ustalenie krytyczności podatności, a w konsekwencji na określenie kolejności ich usuwania. Dodatkowo wyniki scoringu są elementem wspierającym przeprowadzenie analizy ryzyka naruszenia bezpieczeństwa.

7 Przykład praktycznego wykorzystania opracowanej koncepcji

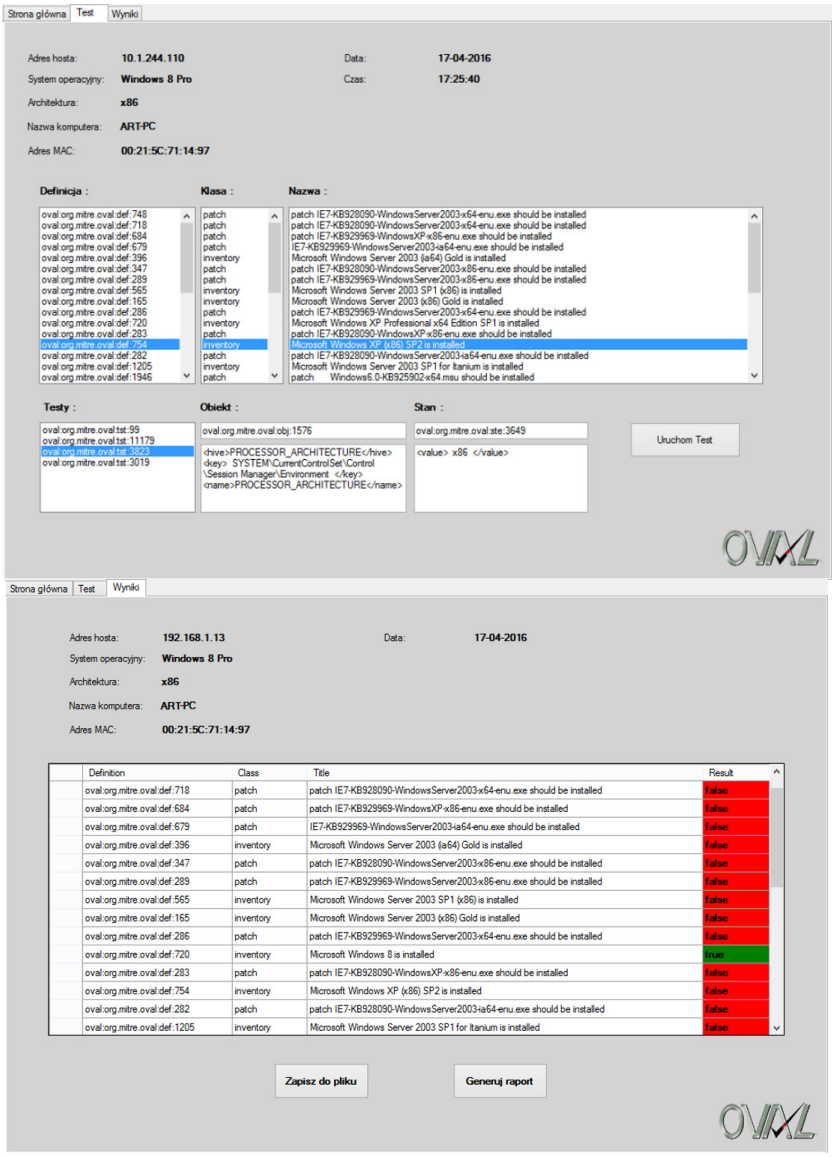
Z uwagi na uniwersalną konstrukcję standardu SCAP, zbudowanego w większości w oparciu o język XML, koncepcja może być zaimplementowana w wielu środowiskach programistycznych. Przykładem zbiorów danych udostępnianych

w plikach XML są słowniki CVE, CCE, CPE i CWE. Praca ze słownikami może być zatem realizowana poprzez implementację dedykowanych przeglądarek, a zaletą takiego podejścia jest możliwość wytworzenia narzędzi ułatwiających pracę z dużymi woluminami danych, pozwalających między innymi na filtrowanie, wyszukiwanie czy grupowanie. Przykład implementacji przeglądarki podatności dla standardu CWE, posiadającej rozbudowane mechanizmy przeglądania rekordów, zilustrowany został na rysunku 6. Rozwiązanie zostało zaimplementowane jako aplikacja internetowa wykonana w technologii Java, przy użyciu szkieletów aplikacyjnych Spring oraz Angular 2. Dzięki nim przeglądarka jest w pełni responsywna, a aplikacja skalowalna, przez co działa poprawnie również na urządzeniach mobilnych.



Rys. 6. Przeglądarka Common Weakness Enumeration; źródło: opracowanie własne
Fig. 6. Common Weakness Enumeration browser application; source: own elaboration

Aby zweryfikować stan badanego systemu teleinformatycznego zgodnie z zaprezentowaną koncepcją, która zakłada między innymi wykonanie testów automatycznych zdefiniowanych w języku OVAL, opracowano autorskie oprogramowania, zaprezentowane na rysunku 7. Jest to przykład aplikacji desktopowej posiadającej interfejs zakładkowy, pozwalający na sprawną nawigację pomiędzy jej funkcjonalnościami. Aplikacja umożliwia wczytanie oraz walidację pliku z definicjami testów w języku OVAL, następnie ich interpretację oraz wykonanie. Wyniki przeprowadzonych testów przedstawione są w tabeli i można je zapisać do pliku tekstowego, a także wyeksportować w formie predefiniowanego raportu.

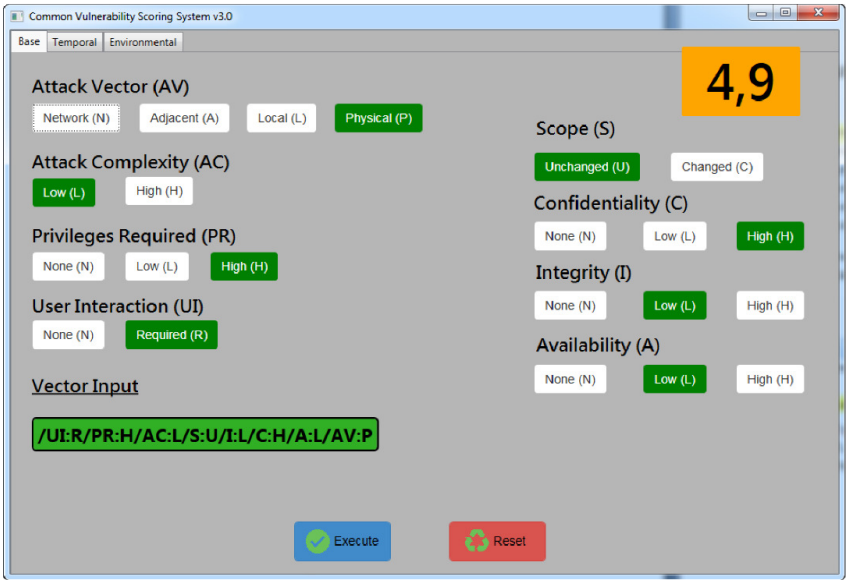


Rys. 7. Przykład oprogramowania wykonującego testy zdefiniowane w języku OVAL; źródło: opracowanie własne.

Fig. 7. Software for conducting OVAL tests; source: own elaboration.

Szacowanie krytyczności podatności za pomocą systemów scoringowych odbywa się przy wykorzystaniu oprogramowania umożliwiającego wprowadzanie lub odczyt wartości składowych wektora podatności, które następnie podstawiane są

do predefiniowanych funkcji dla każdego ze standardów. Przykładowy interfejs graficzny zaimplementowanego oprogramowania kalkulatora dla standardu CVSS 3.0 w języku C# został zaprezentowany na rysunku 8. Oprogramowanie pozwala na wczytywanie gotowego ciągu znaków zbudowanego w standardzie CVSS 3.0, dla każdej z trzech zdefiniowanych grup metryk. Uwzględniona została walidacja wprowadzonego wektora podatności oraz możliwość wprowadzania własnego wektora, w przypadku gdyby zaistniała potrzeba modyfikacji składowych podatności.



Rys. 8. Interfejs programu przeznaczonego do wyliczania stopnia zagrożenia podatności w standardzie CVSS 3.0; źródło: opracowanie własne

Fig. 8. An application for calculating vulnerability severity in CVSS 3.0; source: own elaboration

8 Podsumowanie

Podatności systemów teleinformatycznych mają istotny wpływ na szeroko rozumiane bezpieczeństwo elementów cyberprzestrzeni. Zaproponowana koncepcja ma na celu wsparcie procesu zarządzania podatnościami systemów teleinformatycznych na cyberzagrożenia. Wykorzystuje ona standardy z rodziny SCAP, które stosowane są obecnie w wielu komercyjnych i rządowych rozwiązaniach na świecie. Uzyskane wyniki mogą zostać wykorzystane w obszarze obronności i bezpieczeństwa państwa w zakresie osiągania nowych zdolności do działań w cyberprzestrzeni, w szczególności przez Siły Zbrojne RP.

Literatura

1. *Computer Fraud & Security*. Duqu: son of Stuxnet?, 2011
2. Frasunek P.: System zarządzania procesami bezpieczeństwa w administracji państwowej RP, *CIIP focus - informator o ochronie teleinformatycznej*, 2012
3. Higgins K. J.: Flame Gives Spyware A Next-Gen Update, *Information Week*, 2012
4. Kari A.: An Exceptional war That Ended in Victory for Estonia or an Ordinary e-Disturbance?, *Estonian Narratives of the Cyber-Attacks in 2007*, 2012
5. Kasprzyk R., Stachurski A.: A concept of standard-based vulnerability management automation for IT systems, *Computer Science and Mathematical Modelling*, ISSN 2450-0054, pp. 33-38, No. 3, Warszawa 2016
6. Kasprzyk R., Paź M., Tarapata Z.: Modelowanie i symulacja cyberzagrożeń typu botnet, PTSK, *Symulacja w Badaniach i Rozwoju*, ISSN 2081-6154, Kwartalnik Vol. 6 No. 2/2015, pp. 89-104, Warszawa 2015
7. LeMay E., Mell P., Scarfone K.: The Common Misuse Scoring System (CMSS): Metrics for Software Misuse Vulnerabilities, *NIST*, 2012
8. Mell P., Scarfone K.: A Complete Guide to the Common Vulnerability Scoring System, *NIST*, 2007
9. MITRE, *An Introduction to the OVAL™ Language*, The MITRE Corporation. MITRE, 2006
10. MITRE, *Common Configuration Enumeration - Standardized Identifiers for Security Configuration Issues and Exposures*, 2012
11. MITRE, „Common Vulnerabilities and Exposures. The Standard for Information Security Vulnerability Names
12. MITRE (2014), „Scoring CWE's", MITRE. https://mitre.org/cwss/cwss_v.1.0.1.html.
13. New Scientist. „Thanks, Stuxnet", 2011, wyd. 2806
14. NIST, „Common Platform Enumeration (CPE) ", http://scap.nist.gov/specifications/cpe/
15. NIST, „Security Content Automation Protocol (SCAP) Specifications", http://scap.nist.gov/revision/1.2/index.html
16. NIST, „The Security Content Automation Protocol (SCAP) ", http://scap.nist.gov/
17. Mell P., Scarfone K., „The Common Configuration Scoring System (CCSS): Metrics for Software Security Configuration Vulnerabilities", NIST, 2010
18. Casipa M., Scarfone C.,Waltermire D., „Specification for the Open Checklist Interactive Language (OCIL) Version 2.0", NIST, 2011
19. Scarfone K., Schmidt C., Waltermire D., Ziring N., „Specification For The Extensible Configuration Checklist Description Format (XCCDF) Version 1.2", NIST, 2011
20. Halbardier A., Quinn S., Scarfone K., Waltermire D., „The Technical Specification for the Security Content Automation Protocol (SCAP): SCAP Version 1.2", NIST, 2011
21. Zabielski M., Kasprzyk R., Tarapata Z.: „Metody wykrywania naruszeń prywatności w internetowych sieciach społecznych", PTSK, Symulacja w Badaniach i Rozwoju, ISSN 2081-6154, Kwartalnik Vol. 6 No. 2/2015, ss. 153-161, Warszawa 2015

22. Zabielski M., Tarapata Z., Kasprzyk R., Szkółka K.: „Profile Cloning Detection in Online Social Networks", Computer Science and Mathematical Modelling, ISSN 2450-0054, pp. 39-46, No. 3, Warszawa 2016
23. Zaufana Trzecia Strona (2017), „Masowa, niezwykle skuteczna kampania ransomware wyłącza całe firmy". https://zaufanatrzeciastrona.pl/post/masowa-niezwykle-skuteczna-kampania-ransomware-wylacza-cale-firmy/

Streszczenie

W pracy przedstawiono koncepcję zarządzania bezpieczeństwem systemów teleinformatycznych ze szczególnym naciskiem położonym na automatyzację zarządzania podatnościami na zagrożenia w cyberprzestrzeni. W zaproponowanym podejściu zakłada się wykorzystanie standardów z rodziny SCAP (ang. *The Security Content Automation Protocol*), pozwalających na efektywną wymianę informacji o charakterystykach podatności. Wśród nich wymienić należy słowniki przeznaczone do sformalizowanego opisu różnych aspektów bezpieczeństwa systemów, kwestionariusze do ewaluacji zgodności z założonymi politykami oraz systemy scoringowe pozwalające zbadać krytyczność podatności. W oparciu o założenia zaproponowanej koncepcji opracowano prototypy narzędzi mogących stanowić podstawę do budowy systemu zarządzania bezpieczeństwem.

Słowa kluczowe: cyberzagrożenia, podatności, SCAP

A concept of vulnerability management process for it systems

Summary

The paper presents the concept of security management process for IT systems, with particular emphasis on IT vulnerabilities management automation. The proposed approach involves the use of SCAP (*The Security Content Automation Protocol*) components which allow for an effective vulnerability data exchange. These include enumerations for providing unified naming conventions for IT security artifacts, security checklists for evaluation of the compliance with the developed security policy and scoring systems for measuring vulnerability serverity. On the basis of the assumptions of the proposed concept several mechanisms were implemented which can further be the basis of the complex system for security management.

Keywords: cyber threats, vulnerabilities, SCAP