

Budowa kalkulatora CWSS

Podstawy projektowania bezpiecznego oprogramowania

Zespół:

K4B1S1

Chodubska Justyna

Jochem Maria

Kowalewska Agata

Kępa Karolina

Wilk Katarzyna

Spis treści

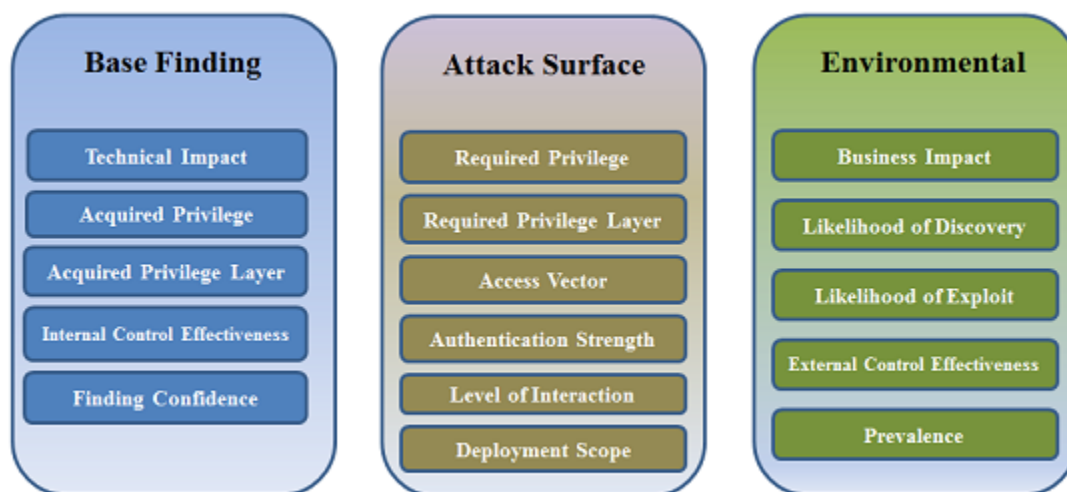
1	Wstęp	3
2	Implementacja	3
2.1	Mechanizm programu	4
2.1.1	Algorithms	4
2.1.2	CWSS.....	4
2.1.3	DATA.....	4
2.1.4	GUI.....	4
2.1.5	MyExeption	4
2.1.6	ReadString	4
2.1.7	Vector	5
2.2	Enum.....	5
3	Dane testowe	5
4	Konfiguracja środowiska	5
5	Instrukcja użycia programu	6
6	Przeprowadzone eksperymenty.....	8
7	Wnioski i możliwość rozwoju aplikacji	9
8	Bibliografia.....	9

1 Wstęp

CWSS (Common Weakness Scoring System), który jest wzorem pozwalającym na wyliczenie bezwzględnej wagi błędu na podstawie jego cech systematycznych. Na końcową wagę CWSS mają wpływ trzy wartości pośrednie:

1. Miara podstawowa (Base Subscore) wynikająca z tych cech błędu, które wspólne dla wszystkich podatnych implementacji i niezmiennych w czasie (np. możliwość zdalnego wykorzystania błędu, brak konieczności uwierzytelnienia).
2. Miara obszaru ataku (Attack Surface Subscore) biorąca pod uwagę wysiłek jaki atakujący musi włożyć w celu wykorzystania słabości.
3. Miara środowiskowa (Environmental Subscore) uwzględniająca lokalną specyfikę w konkretnym systemie teleinformatycznym. Miara ta jest ustalana indywidualnie przez każdą organizację.


CWSS jest stosowany m.in. w katalogach błędów oraz przez producentów komercyjnych skanerów podatności. Poniżej jest przedstawiony zawartość odpowiednich wektorów.



Rys. 1 Części wektora zawierające odpowiednie komponenty

2 Implementacja

W projekcie została wykorzystana JAVA 8 z dodatkowymi pluginami do grafiki zawierające klasę SWTResourceManager. Dany program składa się z 3 paczek. Jedna zawiera główne funkcję odpowiedzialne za mechanizm wyliczający. Kolejna paczka zawiera enumy, które reprezentują odpowiednie składowe wektora. Ostatnia paczka jest odpowiedzialna za grafikę programu, która została specjalnie doimportowana.

 Wojskowa Akademia Techniczna <small>im. Jerozolimskiego</small>	Zespół:	K4B1S1
	Chodubska Justyna Jochem Maria Kępka Karolina Kowalewska Agata Wilk Katarzyna	Strona 3 z 9

2.1 Mechanizm programu

Paczka CWSS zawiera mechanizm programu. Cały mechanizm składa się z 7 klas, które zostały poniżej opisane.

2.1.1 Algorithms

Metody:

- **calc** – metoda licząca Score i trzy składowe tej wartości.

2.1.2 CWSS

Metoda:

- **Main** – klasa główna uruchamiająca aplikację

2.1.3 DATA

Metody:

- **getLiteral** – zwraca wartość literową
- **setLiteral** – ustawia wartość literową
- **getValue** – zwraca wartość liczbową
- **setValue** – ustawia wartość liczbową
- **getName** – zwraca składowe wektora
- **setLiteral** – ustawia składowe wektora

2.1.4 GUI

Klasa zawierająca wygląd aplikacji.

- **show** – uruchamiająca okno aplikacji
- **open** – otwiera okno aplikacji
- **createContents** – zawiera panele do wpisywania wartości
- **setSingleValue** – ustawia wartości dla składowych wektora
- **setResult** – ustawia wynik
- **clearResult** – czyści okno wypisujące wynik
- **setError** – ustawia błąd


2.1.5 MyException

Klasa rozszerzająca klasę Exception. Informująca o nieoczekiwanym zachowaniu programu

2.1.6 ReadString

Klasa parsująca danego wektora

- **firstSplit** – pierwsze dzielenie podanego wektora według zmiennej „/”
- **secondSplit** – drugie dzielenie podanego wektora według zmiennej „:”
- **thirdSplit** – trzecie dzielenie podanego wektora według zmiennej „,”
- **getSolution** – zwraca poprawnie podzielany dane w stringu
- **readError** – klasa wywołująca wyjątek
- **endSolution** – sprawdza poprawność zawierania zmiennych, metoda sprawdzająca poprawność zawartości podziału
- **checkLength** – sprawdza długość wektora
- **controlLength** – sprawdza poprawność wektora
- **getVector** – zwraca odpowiednie podzielony wektor w danych „Data”

 Wojskowa Akademia Techniczna im. Jarosława Dąbrowskiego	Zespół:	K4B1S1
	Chodubska Justyna Jochem Maria Kępka Karolina Kowalewska Agata Wilk Katarzyna	Strona 4 z 9

2.1.7 Vector

- `creat` – porównuje enumy i wybiera odpowiedni wartość słowną, która jest wyświetlana w panelu głównym
- `get` – zwraca wartość liczbową
- `setLiteral` – ustawia składowe wektora
- `setNumerals` – wkłada wartości do mapy

2.2 Enum

Paczka enumów zawiera 16 enumów tyle ile jest składowych wektora. Następnie są one porównywane i wybieranie z nich odpowiednie wartości.

3 Dane testowe

Dane użyte w celu przetestowania aplikacji zostały wybrane z dokumentacji CWSS. Zostanie podany wektor


„`TI:M,0.6/AP:A,1.0/AL:A,1.0/IC:N,1.0/FC:T,1.0/RP:RU,0.7/RL:A,1.0/AV:I,1.0/AS:W,0.9/IN:A,1.0/SC:NA,1.0/BI:L,0.3/DI:NA,1.0/EX:NA,1.0/EC:N,1.0/RE:NA,1.0/P:NA,1.0`”

oraz wektor:

`TI:H,0.9/AP:A,1.0/AL:A,1.0/IC:N,1.0/FC:T,1.0/RP:L,0.9/RL:A,1.0/AV:I,1.0/AS:N,1.0/IN:T,0.9/SC:A,1.0/BI:C,0.9/DI:H,1.0/EX:H,1.0/EC:N,1.0/P:NA,1.0`


4 Konfiguracja środowiska

Dany projekt został wykonany w języku *JAVA 8* pod środowiskiem do kompilacji *Eclipse*. Dodatkowo należy doinstalować pluginy zawierające bibliotekę *SWTResourceManager*. W tym celu w *Eclipse* należy wybrać opcję *Help* a następnie *Install New Software*. Kolejnym krokiem jest wybór wersji *Eclipsa*, na której użytkownik pracuje następnie wyszukać takich paczek jak *Swing Designer*, *SWT Designer*, *SWT Designer Core*, *SWT Designer SWT_AWT Support*, *WindowsBuilderCore*, *WindowsBuilder GroupeLayout Support* . Po wyborze wskazanych paczek wybrać opcję *Next* i potwierdzić opcję wskazując opcję *Finish*. Należy poczekać na doinstalowanie pluginów i utworzyć projekt *WindowsBuilder* na koniec zaimportować projekt. Program jest gotowy do użycia.

 Wojskowa Akademia Techniczna im. Jerozolimskiego	Zespół:	K4B1S1
	Chodubska Justyna Jochem Maria Kępka Karolina Kowalewska Agata Wilk Katarzyna	Strona 5 z 9

5 Instrukcja użycia programu

Program należy uruchomić z klasy CWSS, metodą main. Po uruchomieniu pojawi się panel do wpisywania wektora, który ma być przeliczony.

 CWSS 1.1 Calculator

Sample:

TI:H,0.9/AP:A,1.0/AL:A,1.0/IC:N,1.0/FC:T,1.0/RP:L,0.9/RL:A,1.0/AV:I,1.0/AS:N,1.0/IN:T,0.9/SC:A,1.0/BI:C,0.9/DI:H,1.0/EX:H,1.0/EC:N,1.0/P:NA,1.0

Sprawdz wektor

TI

FC

AS

DI

AP

RP

IN

EX

AL

RL

SC

EC

IC

AV

BI

P

Results

Rysunek 1 Panel startowy aplikacji


Następnie w wyznaczone miejsce należy wpisać wektor według podanego wyżej przykładu i wskazać opcję *Sprawdz wektor*

Sample:

TI:H,0.9/AP:A,1.0/AL:A,1.0/IC:N,1.0/FC:T,1.0/RP:L,0.9/RL:A,1.0/AV:I,1.0/AS:N,1.0/IN:T,0.9/SC:A,1.0/BI:C,0.9/DI:H,1.0/EX:H,1.0/EC:N,1.0/P:NA,1.0

Sprawdz wektor

Rysunek 2 Panel główny z miejscem na wpisanie wektora i zaznaczona opcją „Sprawdz wektor”

 <p>Wojskowa Akademia Techniczna im. Jerozława Dąbrowskiego</p>	Zespół:	K4B1S1
	Chodubska Justyna Jochem Maria Kępka Karolina Kowalewska Agata Wilk Katarzyna	Strona 6 z 9

Po podaniu wektora parametry składowe zmieniają swoją wartość

TI	<input type="text" value="0.6"/>	FC	<input type="text" value="1.0"/>	AS	<input type="text" value="0.9"/>	DI	<input type="text" value="1.0"/>
AP	<input type="text" value="1.0"/>	RP	<input type="text" value="0.7"/>	IN	<input type="text" value="1.0"/>	EX	<input type="text" value="1.0"/>
AL	<input type="text" value="1.0"/>	RL	<input type="text" value="1.0"/>	SC	<input type="text" value="1.0"/>	EC	<input type="text" value="1.0"/>
IC	<input type="text" value="1.0"/>	AV	<input type="text" value="1.0"/>	BI	<input type="text" value="0.3"/>	P	<input type="text" value="1.0"/>

Rysunek 3 Zmiana parametrów składowych z wypisanego wcześniej wektora

W przypadku dobrze podanego wektora, zostanie policzony i wyświetlone wyniki dla danych wejściowych.

Results

AL: Application
IC: None
FC: Proven True
RP: Regular User
RL: Application
AV: Internet
AS: Weak
IN: Automated
SC: Not Applicable
BI: Low
DI: Not Applicable
EX: Not Applicable
EC: None
P: Not Applicable
=====

Base: 84.0
Ass: 0.935
Es: 0.65
Score: 51.05100000000001

Rysunek 4 Podsumowanie poprawnie podanego wektora

W przypadku źle podanego wektora co oznacza to, że użytkownik poda za krótki lub za długi wektor nie zostaną wyświetlone wyniki dla danych wejściowych tylko komunikat o błędzie.

CWSS 1.1 Calculator

Sample:
 TI:H,0.9/AP:A,1.0/AL:A,1.0/IC:N,1.0/FC:T,1.0/RL:A,1.0/AV:I,1.0/AS:N,1.0/IN:T,0.9/SC:A,1.0/BI:C,0.9/DI:H,1.0/EX:H,1.0/EC:N,1.0/P:NA,1.0

AV:I,1.0/AS:W,0.9/IN:A,1.0/SC:NA,1.0/BI:L,0.3/DI:NA,1.0/EX:NA,1.0/EC:N,1.0/RE:NA,1.0/P:NA,1.0

Sprawdz wektor

TI	<input type="text" value="0.6"/>	FC	<input type="text" value="1.0"/>	AS	<input type="text" value="0.9"/>	DI	<input type="text" value="1.0"/>
AP	<input type="text" value="1.0"/>	RP	<input type="text" value="0.7"/>	IN	<input type="text" value="1.0"/>	EX	<input type="text" value="1.0"/>
AL	<input type="text" value="1.0"/>	RL	<input type="text" value="1.0"/>	SC	<input type="text" value="1.0"/>	EC	<input type="text" value="1.0"/>
IC	<input type="text" value="1.0"/>	AV	<input type="text" value="1.0"/>	BI	<input type="text" value="0.3"/>	P	<input type="text" value="1.0"/>

Results

Podano złe dane - Wektor niepoprawny

Rysunek 5 Panel główny ze źle podanym wektorem

6 Przeprowadzone eksperymenty

Eksperymenty aplikacji zostały przeprowadzone na poprawnych wektorach oraz o niepoprawnej długości danych wejściowych. Zastosowaliśmy wyjątki w obu przypadkach jeśli użytkownik poda niewłaściwy wektor. Poniżej zostanie zaprezentowane poprawne działanie programu. W obu przypadkach kiedy wektor zostanie podany poprawnie oraz nie.

CWSS 1.1 Calculator

Sample:
 TI:H,0.9/AP:A,1.0/AL:A,1.0/IC:N,1.0/FC:T,1.0/RL:A,1.0/AV:I,1.0/AS:N,1.0/IN:T,0.9/SC:A,1.0/BI:C,0.9/DI:H,1.0/EX:H,1.0/EC:N,1.0/P:NA,1.0

AV:I,1.0/AS:W,0.9/IN:A,1.0/SC:NA,1.0/BI:L,0.3/DI:NA,1.0/EX:NA,1.0/EC:N,1.0/RE:NA,1.0/P:NA,1.0

Sprawdz wektor

TI	<input type="text" value="0.6"/>	FC	<input type="text" value="1.0"/>	AS	<input type="text" value="0.9"/>	DI	<input type="text" value="1.0"/>
AP	<input type="text" value="1.0"/>	RP	<input type="text" value="0.7"/>	IN	<input type="text" value="1.0"/>	EX	<input type="text" value="1.0"/>
AL	<input type="text" value="1.0"/>	RL	<input type="text" value="1.0"/>	SC	<input type="text" value="1.0"/>	EC	<input type="text" value="1.0"/>
IC	<input type="text" value="1.0"/>	AV	<input type="text" value="1.0"/>	BI	<input type="text" value="0.3"/>	P	<input type="text" value="1.0"/>

Results

Podano złe dane - Wektor niepoprawny

Rysunek 6 Panel główny ze źle podanym wektorem

CWSS 1.1 Calculator

Sample:

TI:H,0.9/AP:A,1.0/AL:A,1.0/IC:N,1.0/FC:T,1.0/RP:L,0.9/RL:A,1.0/AV:I,1.0/AS:N,1.0/IN:T,0.9/SC:A,1.0/BI:C,0.9/DI:H,1.0/EX:H,1.0/EC:N,1.0/P:NA,1.0

Sprawdz wektor

TI 0.6

FC 1.0

AS 0.9

DI 1.0

AP 1.0

RP 0.7

IN 1.0

EX 1.0

AL 1.0

RL 1.0

SC 1.0

EC 1.0

IC 1.0

AV 1.0

BI 0.3

P 1.0

Results

AL: Application
 IC: None
 FC: Proven True
 RP: Regular User
 RL: Application
 AV: Internet
 AS: Weak
 IN: Automated
 SC: Not Applicable
 BI: Low
 DI: Not Applicable
 EX: Not Applicable
 EC: None
 P: Not Applicable

=====

Base: 84.0
 Ass: 0.935
 Es: 0.65
 Score: 51.05100000000001

Rysunek 7 Panel główny z dobrze podanym wektorem

7 Wnioski i możliwość rozwoju aplikacji

Program został poprawnie zaimplementowany. Kalkulator pełni swoją rolę i prawidłowo liczy wartości wektora. Dodatkową możliwością rozwoju programu może być wprowadzenie zmiennych przez „klikanie” na odpowiednie składowe wektora w panelu a nie wpisywanie danych. Ponadto można rozszerzyć kalkulator o kalkulator do CVSS, w zależności jakiej informacji potrzebują użytkownik.


8 Bibliografia

Dokumentacja cwss:

https://cwe.mitre.org/cwss/cwss_v1.0.1.html

dołączenie SWTResourceManager:

<https://www.youtube.com/watch?v=OHbSx2iVuJE>

 <p>Wojskowa Akademia Techniczna im. Jerzego Dąbrowskiego</p>	Zespół:	K4B1S1
	Chodubska Justyna Jochem Maria Kępka Karolina Kowalewska Agata Wilk Katarzyna	Strona 9 z 9