Metody projektowania i wdrażania systemów informatycznych

Sprawozdanie z projektu

Kampa Agnieszka

WCY21KA1S1

80452

Temat: Aplikacja do wykonania ataku na algorytm XTR dla użytku firmy zajmującej się badaniem bezpieczeństwa informacji

1. Opis fragmentu organizacji

Fragmentem firmy zajmującym się badaniem algorytmów kryptograficznych jest Laboratorium Badawcze Kryptologii.

Laboratorium Badawcze Kryptologii stanowi kluczowy element struktury firmy zajmującej się badaniem bezpieczeństwa informacji. Jego głównym zadaniem jest analiza, ocena i rozwijanie nowych metod kryptograficznych oraz badanie ich odporności na różnego rodzaju ataki. Laboratorium to składa się z zespołu wykwalifikowanych kryptologów, matematycznych analityków oraz inżynierów oprogramowania, którzy wspólnie pracują nad identyfikacją potencjalnych słabości w algorytmach kryptograficznych i opracowaniem skutecznych strategii ochrony przed nimi.

LBK odpowiada za szczegółową analizę różnorodnych algorytmów kryptograficznych stosowanych w branży. W ramach swoich działań zespół przeprowadza testy mające na celu ocenę bezpieczeństwa algorytmów oraz identyfikację możliwych wektorów ataków. W tym kontekście szczególną uwagę zwraca się na nowoczesne algorytmy i protokoły, które mogą być używane w przyszłości, a także na ocenie obecnych standardów kryptograficznych pod kątem ich odporności na ataki kwantowe.

1. Struktura organizacyjna LBK

a) Główny Kryptolog

- Zespół Kryptologów

b) Główny Inżynier Oprogramowania

- Zespół Inżynierów Oprogramowania

c) Analityk ds. Bezpieczeństwa

d) Koordynator Projektów

e) Asystent Badawczy

1. Opis problemów i wyzwań

* Manualne przeprowadzanie ataków:

Przeprowadzanie kryptoanalizy ręcznie jest niezwykle czasochłonne i wymaga znacznych zasobów ludzkich. Każdy krok procesu musi być wykonany ręcznie, co może prowadzić do długotrwałych i nieefektywnych procedur. Kryptolodzy muszą ręcznie kodować różne warianty ataków, testować je i analizować wyniki. Bez automatyzacji, każdy drobny błąd może wymagać powtórzenia całego procesu, co dodatkowo zwiększa czas i koszty. Taka sytuacja jest nieefektywna i może prowadzić do opóźnień w projekcie oraz ograniczenia zakresu testów, które można przeprowadzić w dostępnym czasie.

* Ograniczona dokładność wyników:

Bez zaawansowanego narzędzia do automatyzacji ataków, analizy mogą być mniej dokładne. Ręczne testowanie i analiza mogą nie wychwycić subtelnych słabości w algorytmie XTR. Dedykowana aplikacja mogłaby systematycznie i precyzyjnie przeprowadzać setki, jeśli nie tysiące, testów w krótkim czasie, co jest niemal niemożliwe do osiągnięcia ręcznie. Dokładność wyników jest kluczowa dla zrozumienia bezpieczeństwa algorytmu oraz dla rozwoju skutecznych strategii kryptoanalizy. Brak narzędzia może prowadzić do niekompletnych lub niedokładnych wniosków, co z kolei wpływa na jakość raportów i rekomendacji dostarczanych klientom.

* Brak możliwości skalowania testów:

Skalowanie testów kryptograficznych jest kluczowe dla oceny odporności algorytmów na różne scenariusze ataków. Manualne przeprowadzanie testów jest nie tylko czasochłonne, ale także trudno skalowalne. Bez automatyzacji, testowanie algorytmu XTR na większych instancjach czy w różnych konfiguracjach staje się praktycznie niemożliwe. Dedykowana aplikacja umożliwia uruchamianie wielu testów równolegle i w różnych warunkach, co pozwala na uzyskanie bardziej reprezentatywnych wyników i lepsze zrozumienie potencjalnych zagrożeń. Brak możliwości skalowania ogranicza zespół do małych, ręcznie zarządzanych testów, co nie odzwierciedla rzeczywistych warunków i zagrożeń.

1. Wizja rozwiązania

Aby skutecznie przeciwdziałać problemom związanym z niemożnością przeprowadzania testów na dużą skalę bez dedykowanej aplikacji, można stworzyć innowacyjną aplikację wspierającą zespół badawczy w analizie algorytmu XTR. Można zawrzeć w niej następujące rozwiązania:

* Automatyzacja ataków: Aplikacja będzie umożliwiała automatyczne generowanie i przeprowadzanie różnorodnych ataków na algorytm XTR. Dzięki temu zespół będzie mógł szybko i skutecznie testować różne scenariusze ataków, co pozwoli na pełniejsze zrozumienie jego słabości.
* Skalowalność: Aplikacja będzie zoptymalizowana pod kątem wydajności i skalowalności, co umożliwi przeprowadzanie testów na dużą skalę. Dzięki temu zespół będzie mógł łatwo przetestować algorytm XTR na różnych instancjach i w różnych warunkach, co pozwoli na uzyskanie bardziej reprezentatywnych wyników.
* Integracja z symulatorami kwantowymi: W celu przeprowadzania testów wyżarzania kwantowego, aplikacja będzie integrowana z symulatorami kwantowymi.
* Monitoring postępu testów: Aplikacja będzie zapewniała narzędzia do monitorowania postępu testów i analizy wyników.
* Dokumentacja i raportowanie: Aplikacja będzie automatycznie generować raporty z przeprowadzonych testów oraz dokumentować wszystkie kroki analizy. Dzięki temu zespół będzie miał pełną kontrolę nad procesem testowania i łatwy dostęp do wyników.

1. Interesariusze

* Zespoły badawcze ds. kryptografii: Zespoły zajmujące się badaniami kryptograficznymi będą zainteresowane aplikacją, ponieważ umożliwia ona skuteczne przeprowadzanie testów na algorytm XTR i wyżarzanie kwantowe, co może prowadzić do odkrycia nowych słabości i zagrożeń.
* Firmy zajmujące się bezpieczeństwem informacji: Firmy oferujące usługi z zakresu bezpieczeństwa informacji mogą być zainteresowane aplikacją jako narzędziem wspierającym ich działalność.

* Organizacje rządowe i instytucje państwowe: Organizacje rządowe i instytucje państwowe odpowiedzialne za kwestie bezpieczeństwa narodowego mogą być zainteresowane aplikacją jako narzędziem wspierającym ich działania w zakresie kryptografii. Skuteczne testowanie algorytmów kryptograficznych może przyczynić się do wzmocnienia bezpieczeństwa komunikacji i danych wrażliwych.
* Przedsiębiorstwa z sektora IT: Przedsiębiorstwa z sektora IT, w tym dostawcy usług chmurowych, dostawcy oprogramowania, dostawcy usług sieciowych itp., mogą być zainteresowane aplikacją jako narzędziem wspierającym ich prace nad zapewnieniem bezpieczeństwa swoich produktów i usług.
* Uczelnie i instytucje akademickie: Uczelnie i instytucje akademickie prowadzące badania z zakresu kryptografii mogą być zainteresowane aplikacją jako narzędziem wspierającym ich prace badawcze. Aplikacja może być wykorzystywana do celów naukowych oraz w ramach programów dydaktycznych.

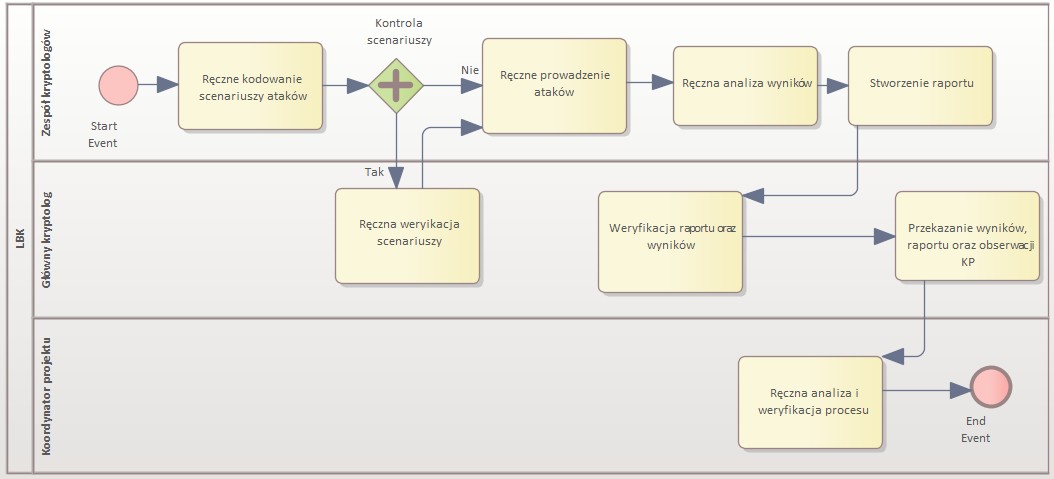
**+DIAGRAM KLAS/OBIEKTOW**

1. Diagram AS-IS

Aktualnie atak na algorytm wykonuje się ręcznie co jest przyczyną wielu błędów oraz niedopatrzeń.

Ręczne tworzenie oprogramowania czy raportów jest niezwykle czasochłonne. Oprócz tego w celu zachowania wysokiej jakości wykonywanego ataku potrzebna jest duża ilość zasobów ludzkich.

Diagram:



1. Diagram TO-BE

Wdrożenie aplikacji do automatycznego przeprowadzania ataku pozwoli na oszczędność zasobów ludzkich, czasu oraz zminimalizuję liczbę potencjalnych pomyłek i przeoczeń.

Diagram:

Obraz zawierający tekst, zrzut ekranu, diagram, linia

Opis wygenerowany automatycznie

1. Wymagania funkcjonalne

|  |  |  |
| --- | --- | --- |
| Nr | Nazwa | Opis |
| WF01 | Automatyczne generowanie scenariuszy ataków | Opis: System powinien generować różnorodne scenariusze ataków na algorytm XTR na podstawie zdefiniowanych parametrów i reguł.  Zakres danych: Parametry ataków (np. typ ataku, intensywność, cel), reguły generowania.  Reguły walidacji: Parametry muszą być zgodne z zdefiniowanymi typami ataków; intensywność ataku nie może przekraczać maksymalnych wartości określonych w specyfikacji. |
| WF02 | Przeprowadzanie ataków | Opis: System powinien automatycznie przeprowadzać ataki na algorytm XTR zgodnie z wygenerowanymi scenariuszami.  Zakres danych: Scenariusze ataków, logi z przeprowadzonych ataków, wyniki ataków.  Reguły walidacji: Scenariusze muszą być poprawnie sformułowane; logi muszą zawierać pełną historię działań. |
| WF03 | Integracja z symulatorami kwantowymi | Opis: System powinien mieć możliwość integracji z symulatorami kwantowymi w celu przeprowadzania testów wyżarzania kwantowego.  Zakres danych: Interfejsy API symulatorów kwantowych, konfiguracje testów kwantowych.  Reguły walidacji: Integracje muszą być zgodne z dokumentacją API symulatorów; konfiguracje muszą być weryfikowane pod kątem zgodności z wymaganiami testów kwantowych. |
| WF04 | Monitorowanie postępu testów | Opis: System powinien zapewniać narzędzia do monitorowania w czasie rzeczywistym postępu testów kryptograficznych.  Zakres danych: Aktualny status testów, procent ukończenia, logi bieżących działań.  Reguły walidacji: Status musi być aktualizowany co najmniej raz na minutę; logi muszą być kompletne i czytelne. |
| WF05 | Automatyczna analiza wyników | Opis: System powinien automatycznie zbierać i analizować dane z przeprowadzonych testów, identyfikując potencjalne słabości algorytmu XTR.  Zakres danych: Wyniki testów, wykryte słabości, statystyki dotyczące ataków.  Reguły walidacji: Analizy muszą być oparte na poprawnych i kompletnych danych z testów; wykryte słabości muszą być klasyfikowane według zdefiniowanych kategorii. |
| WF06 | Generowanie raportów | Opis: System powinien automatycznie generować szczegółowe raporty z wynikami testów i analizą.  Zakres danych: Wyniki testów, analiza słabości, zalecenia.  Reguły walidacji: Raporty muszą zawierać wszystkie wymagane sekcje; format raportów musi być zgodny z ustalonym szablonem. |
| WF07 | Zarządzanie konfiguracją testów | Opis: System powinien umożliwiać definiowanie i zarządzanie różnymi konfiguracjami testów, w tym parametrami wejściowymi i warunkami brzegowymi.  Zakres danych: Konfiguracje testów, parametry wejściowe, warunki brzegowe.  Reguły walidacji: Konfiguracje muszą być zapisywane zgodnie z określonym formatem; parametry wejściowe muszą mieścić się w zdefiniowanych zakresach. |
| WF08 | Interfejs użytkownika | Opis: System powinien posiadać intuicyjny interfejs użytkownika umożliwiający łatwą nawigację i zarządzanie procesami testowymi.  Zakres danych: Elementy interfejsu, dostępne opcje, konfiguracje użytkownika.  Reguły walidacji: Interfejs musi być zgodny z zasadami użyteczności; wszystkie funkcje muszą być dostępne z poziomu interfejsu. |
| WF09 | Historia i archiwizacja testów | Opis: System powinien umożliwiać przechowywanie i przeglądanie historii przeprowadzonych testów oraz archiwizację wyników.  Zakres danych: Historia testów, wyniki testów, metadane testów.  Reguły walidacji: Dane historyczne muszą być przechowywane przez określony czas; archiwizacja musi być zgodna z polityką przechowywania danych. |
| WF10 | Powiadomienia i alerty | Opis: System powinien wysyłać powiadomienia i alerty w przypadku wykrycia krytycznych słabości algorytmu XTR lub zakończenia testów.  Zakres danych: Typy powiadomień, kryteria wyzwalania, kanały komunikacji.  Reguły walidacji: Powiadomienia muszą być wysyłane natychmiast po wykryciu krytycznej słabości; wszystkie powiadomienia muszą być rejestrowane. |
| WF11 | Integracja z istniejącymi narzędziami | Opis: System powinien umożliwiać integrację z istniejącymi narzędziami używanymi przez Laboratorium Badawcze Kryptologii, takimi jak narzędzia do analizy kryptograficznej.  Zakres danych: Interfejsy API narzędzi, konfiguracje integracji.  Reguły walidacji: Integracje muszą być zgodne z dokumentacją API; dane wymieniane między systemami muszą być kompletne i zgodne z oczekiwaniami. |

Obraz zawierający tekst, zrzut ekranu, Czcionka, menu

Opis wygenerowany automatycznie



1. Macierz wymagań i aktywności w Enterprise Architect