KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu
Analiza złośliwego oprogramowania [S2Inf1E-CYB>MSA]

Przedmiot

Kierunek studiów
Informatyka/Computing

Rok/Semestr
2/3

Studia w zakresie (specjalność)
Cyberbezpieczeństwo

Profil studiów
ogólnoakademicki

Poziom studiów
drugiego stopnia

Język oferowanego przedmiotu
angielski

Forma studiów
stacjonarne

Wymagalność
obligatoryjny

Liczba godzin

Wykład
15

Laboratorium
30

Inne (np. online)
0

Ćwiczenia
0

Projekty/seminaria
0

Liczba punktów ECTS

4,00

Koordynatorzy

dr hab. inż. Piotr Zwierzykowski prof. PP
piotr.zwierzykowski@put.poznan.pl

mgr inż. Błażej Nowak
blazej.nowak@put.poznan.pl

Wykładowcy

Wymagania wstępne

Student rozpoczynający ten przedmiot powinien mieć podstawową wiedzę z zakresu sieci komputerowych, algorytmów kryptograficznych i systemów operacyjnych Windows i Linux. Powinien również posiadać umiejętność pozyskiwania informacji ze wskazanych źródeł oraz mieć gotowość do podjęcia współpracy w ramach zespołu.

Cel przedmiotu

Przekazanie studentom wiedzy z zakresu szeroko rozumianej analizy złośliwego oprogramowania, w tym metod i narzędzi wykorzystywanych analizy statycznej i dynamicznej takiego oprogramowania oraz elementów inżynierii wstecznej. W ramach realizacji przedmiotu zostaną omówione metody wybrane metody statycznej i dynamicznej analizy złośliwego oprogramowania oraz wykorzystywanej w tym celu inżynierii wstecznej. W ramach ćwiczeń laboratoryjnych student zapozna się w praktyce z narzędziami umożliwiającymi wykrycie złośliwego oprogramowania

Przedmiotowe efekty uczenia się

Wiedza:
ma uporządkowaną i podbudowaną teoretycznie wiedzę ogólną związaną z kluczowymi zagadnieniami z zakresu analizy złośliwego oprogramowania.
ma zaawansowaną wiedzę szczegółową dotyczącą wybranych zagadnień z zakresu szeroko rozumianej analizy złośliwego oprogramowanie oraz metod i narzędzi wykorzystywanych do analizy statycznej i dynamicznej oraz inżynierii wstecznej.
ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach informatyki i telekomunikacji w zakresie wykrywania , analizy statycznej i dynamicznej złośliwego oprogramowania.
ma zaawansowaną i szczegółową wiedzę o procesach zachodzących systemach wykorzystywanych do dynamicznej analizy złośliwego oprogramowania.

Umiejętności:
potrafi pozyskiwać informacje na temat metod statycznej i dynamicznej analizy złośliwego oprogramowania. pozyskane informacje (w języku polskim i angielskim) potrafi integrować i poddawać krytycznej ocenie.
potrafi wykorzystać metody eksperymentalne do formułowania i rozwiązywania zadań inżynierskich i prostych problemów badawczych w obszarze analizy złośliwego oprogramowania.
potrafi integrować wiedzę z różnych obszarów informatyki i telekomunikacji przy formułowaniu i rozwiązywaniu zadań inżynierskich związanych z wykrywaniem i analizą złośliwego oprogramowania.
potrafi ocenić przydatność i możliwość wykorzystania nowych rozwiązań sprzętowych i programowych służących do rozwiązywania zadań inżynierskich, polegających na budowie bezpiecznych systemów przesyłania danych.

Kompetencje społeczne:
rozumie, że w zakresie bezpieczeństwa teleinformatycznego wiedza i umiejętności bardzo szybko stają się przestarzałe.
rozumie znaczenie wykorzystywania najnowszej wiedzy z zakresu bezpieczeństwa teleinformatycznego w rozwiązywaniu problemów badawczych i praktycznych.
ma świadomość konieczności profesjonalnego podejścia do rozwiązywanych problemów bezpieczeństwa teleinformatycznego i podejmowania odpowiedzialności za proponowane przez siebie projekty.

Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Wiedza nabyta w ramach wykładu jest weryfikowana na kolokwium ustnym i/lub pisemnym.
Zagadnienia zaliczeniowe, na podstawie których opracowywane są pytania, przesyłane są studentom drogą mailową z wykorzystaniem systemu uczelnianej poczty elektronicznej, lub umieszczane w kursie przedmiotowym w uczelnianym systemie zdalnego nauczania.
Kolokwium ustne i/lub pisemne obejmuje od 3 do 5 pytań, na które oczekuje się odpowiedzi opisowej. Każda odpowiedź na pytanie jest oceniana w skali od 0 do 5 punktów. Każde pytanie jest równo punktowane. Próg zaliczeniowy: 50% punktów.
W przypadku kolokwium ustnego studenci losują pytania ze zbioru 30 pytań. W przypadku kolokwium pisemnego pytania są zadawane przez prowadzącego.
Umiejętności nabyte w ramach zajęć laboratoryjnych weryfikowane są na bieżąco. Na każdych zajęciach laboratoryjnych oceniana jest poprawność wykonania ćwiczeń w skali od 2 do 5. Ocena końcowa jest średnią ocen uzyskanych z poszczególnych zajęć laboratoryjnych. Ocena końcowa jest średnią ocen uzyskanych z poszczególnych zajęć laboratoryjnych.

Treści programowe

Tematyka wykładów:
- Introduction to Malware Analysis
- Classification of Malware
- Static Analysis of Malware
- Dynamic Analysis of Malware
- Reverse Engineering in Malware Analysis
- Static and Dynamic Reversing
- Malware Functionalities and Persistence
- Malware Obfuscation and Evasion Techniques

Tematyka laboratoriów:
- Basic Static Analysis Techniques,
- Malware Analysis in Virtual Machines,
- Basic Dynamic Analysis Techniques,
- IDA - Interactive Disassembler,
- C Constructions in ASM,
- Malware Analysis in Windows OS,
- OllyDbg - Interactive Debugger,
- Network Signatures,
- Encryption,
- C++ Analysis.

Tematyka zajęć

Wprowadzenie do analizy złośliwego oprogramowania
Analiza złośliwego oprogramowania polega na badaniu i zrozumieniu sposobu działania, celów oraz skutków działania złośliwego kodu. Celem jest identyfikacja, klasyfikacja, oraz opracowanie metod detekcji i neutralizacji zagrożeń. Analiza może być prowadzona na różnych poziomach, od powierzchownej identyfikacji zagrożeń, po szczegółową dekompilację i badanie kodu.

Klasyfikacja złośliwego oprogramowania
Złośliwe oprogramowanie można klasyfikować według różnych kryteriów, takich jak metoda infekcji, typ działania, czy cel ataku.

Inżynieria wsteczna w analizie złośliwego oprogramowania
Inżynieria wsteczna polega na dekompilacji i analizie złośliwego kodu w celu zrozumienia jego działania.

Funkcjonalności i trwałość złośliwego oprogramowania
Złośliwe oprogramowanie często zawiera różnorodne funkcjonalności mające na celu trwałe zainstalowanie się w systemie oraz wykonywanie złośliwych działań.

Techniki zaciemniania i unikania wykrycia
Złośliwe oprogramowanie często stosuje różne techniki zaciemniania (obfuscation) i unikania wykrycia (evasion) w celu utrudnienia analizy oraz wykrycia przez systemy bezpieczeństwa.

Podstawowe techniki analizy statycznej
Analiza statyczna polega na badaniu kodu programu bez jego uruchamiania. W tym celu analizujemy pliki binarne, kod źródłowy (jeśli jest dostępny), i inne zasoby, takie jak biblioteki lub pliki konfiguracyjne. Narzędzia często używane w analizie statycznej to deasembler, narzędzia do analizy binarnej, oraz narzędzia do analizy kodu źródłowego.

Analiza złośliwego oprogramowania w maszynach wirtualnych
Korzystanie z maszyn wirtualnych (VM) pozwala na bezpieczne uruchamianie i analizowanie złośliwego oprogramowania bez ryzyka infekcji systemu głównego. Popularne rozwiązania to VMware, VirtualBox oraz Hyper-V. Analiza w VM umożliwia tworzenie izolowanego środowiska, w którym można monitorować zachowanie podejrzanego oprogramowania, śledzić zmiany w systemie plików, rejestrze systemowym oraz ruchu sieciowym.

Podstawowe techniki analizy dynamicznej
Analiza dynamiczna polega na badaniu zachowania programu podczas jego działania. Obejmuje to uruchamianie programu w kontrolowanym środowisku i monitorowanie jego aktywności, takiej jak komunikacja sieciowa, operacje na plikach, oraz interakcje z systemem operacyjnym. Narzędzia używane w analizie dynamicznej to m.in. Process Monitor, Process Explorer, Wireshark oraz narzędzia do śledzenia systemu plików i rejestru.

IDA - Interaktywny deasembler
IDA (Interactive Disassembler) to jedno z najpotężniejszych narzędzi do analizy statycznej oprogramowania. Pozwala na deasemblację kodu binarnego do czytelnej formy assemblerowej, co umożliwia analizę struktury programu oraz jego logiki. IDA oferuje interaktywne środowisko z możliwością

dodawania komentarzy, etykiet oraz analizy przepływu kontrolnego.

Konstrukcje języka C w ASM
Zrozumienie, jak konstrukcje języka C (takie jak pętle, instrukcje warunkowe, funkcje) są przekształcane na kod assemblerowy przez kompilator, jest kluczowe w analizie kodu niskopoziomowego. Pozwala to na łatwiejsze śledzenie logiki programu i identyfikowanie kluczowych fragmentów kodu, które mogą odpowiadać za złośliwe działania.

Analiza złośliwego oprogramowania w systemie Windows
System Windows jest najczęściej atakowanym systemem operacyjnym, dlatego zrozumienie jego architektury oraz mechanizmów bezpieczeństwa jest kluczowe. Analiza złośliwego oprogramowania w Windows obejmuje badanie zmian w rejestrze, śledzenie procesów i usług systemowych, analizę plików wykonywalnych oraz monitorowanie interakcji z systemem operacyjnym.

OllyDbg - Interaktywny debugger
OllyDbg to popularne narzędzie do analizy dynamicznej, które pozwala na interaktywne debugowanie aplikacji. Dzięki OllyDbg można wykonywać krok po kroku instrukcje programu, monitorować zmienne, śledzić przepływ kontrolny oraz identyfikować błędy i złośliwe zachowania w kodzie. Jest szczególnie użyteczne do analizy programów bez dostępu do kodu źródłowego.

Sygnatury sieciowe
Sygnatury sieciowe to wzorce używane do identyfikacji złośliwego ruchu sieciowego. Mogą to być specyficzne sekwencje bajtów, adresy IP, domeny, czy też specyficzne schematy komunikacji używane przez złośliwe oprogramowanie. Narzędzia do analizy ruchu sieciowego, takie jak Wireshark lub systemy IDS (Intrusion Detection System), wykorzystują te sygnatury do wykrywania i blokowania podejrzanego ruchu.

Szyfrowanie
Szyfrowanie jest często używane przez złośliwe oprogramowanie do ukrywania swoich operacji lub do zabezpieczania komunikacji z serwerami kontrolującymi. Zrozumienie technik szyfrowania, takich jak AES, RSA, oraz umiejętność ich dekodowania, jest kluczowe w analizie złośliwego oprogramowania. Często analiza kluczy szyfrujących lub metod szyfrowania może prowadzić do odkrycia informacji o funkcjonowaniu malware'u.

Analiza C++
Analiza programów napisanych w C++ jest bardziej skomplikowana niż tych napisanych w C, ze względu na złożoność języka oraz użycie obiektowości. Obejmuje to zrozumienie konstrukcji takich jak klasy, dziedziczenie, polimorfizm, oraz zarządzanie pamięcią. Deasemblacja kodu C++ oraz analiza jego struktury wymaga zaawansowanych narzędzi i technik, takich jak analiza wirtualnych tablic, identyfikacja metod wirtualnych oraz rekonstrukcja struktury klas.

Metody dydaktyczne

Wykład: prezentacja multimedialna, ilustrowana przykładami podawanymi na tablicy.
Laboratorium: ćwiczenia praktyczne w grupach lub indywidualne, z wykorzystaniem środowisk i narzędzi testowych.

Literatura

Podstawowa
D. Barker: Malware Analysis Techniques, Packt>, 2021
Uzupełniająca
1. Alexey Kleymenov, Amr Thabet: Mastering Malware Analysis,Packt>, 2019
2. Reginald Wong: Mastering Reverse Engineering, Packet>, 2018
3. K.A. Monnappa: Learning Malware Analysis, Pack>, 2018
4. M. Skikorski, A. Honing: Practical Malware Analysis: The Hands-On Guide to Dissecting Malicious Software, No Starch Press; 1st edition , 2012
5.O. Or-Meir, Nir Nissim, Yuval Elovici, and Lior Rokach: Dynamic Malware Analysis in the Modern Era—A State of the Art Survey, ACM Computing Surveys, Vol. 52 Issue 5, October 2019, Article No.: 88, pp 1–48, 10.1145.3329786

Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	100	4,00
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	45	2,00
Praca własna studenta (studia literaturowe, przygotowanie do zajęć laboratoryjnych/ćwiczeń, przygotowanie do kolokwiów/egzaminu, wykonanie projektu)	55	2,00