Zestawienie OWASP Top 10 dla aplikacji opartych na dużych modelach językowych (LLM)

Wiosną 2023 roku rozpocząłem badania nad lukami w zabezpieczeniach charakterystycznymi dla dużych modeli językowych (LLM). W tamtym czasie istniało już sporo opracowań dotyczących bezpieczeństwa sztucznej inteligencji w ogóle, jednak brakowało usystematyzowanych badań poświęconych konkretnie modelom LLM. Udało mi się jednak dotrzeć do kilku prac naukowych i wpisów na blogach, które poruszały wybrane zagadnienia z tego obszaru. Zacząłem gromadzić te materiały i podsumowywać je przy użyciu narzędzia ChatGPT. Następnie, posiłkując się przykładami z aktualnego zestawienia OWASP Top 10 dla aplikacji internetowych, poprosiłem ChatGPT o wygenerowanie propozycji listy „Top 10" dla modeli LLM w podobnym formacie. Uznałem, że rezultat jest interesujący, więc przesłałem go Jeffowi Williamsowi – jednemu z założycieli organizacji OWASP (Open Worldwide Application Security Project) – aby poznać jego opinię. Jeff, pełniący funkcję dyrektora ds. technologii (CTO) w firmie Contrast Security, był autorem pierwszego zestawienia OWASP Top 10 w 2001 roku. Jego celem było stworzenie przystępnego źródła wiedzy dla programistów, opisującego najpoważniejsze zagrożenia i najbardziej podatne na ataki obszary aplikacji internetowych. W tamtym okresie sieć WWW miała zaledwie kilka lat, a większość twórców oprogramowania miała niewielkie lub żadne pojęcie o tym, jak projektować bezpieczne aplikacje. Ów pierwotny dokument stał się przełomowym opracowaniem i fundamentem w dziedzinie bezpieczeństwa aplikacji. Nie wspomniałem Jeffowi, że moja lista została w dużej mierze wygenerowana przez sztuczną inteligencję. Liczyłem na to, że jako twórca oryginalnego zestawienia, będzie on w stanie ocenić, czy moja propozycja wnosi coś nowego i czy warto kontynuować prace w tym kierunku. Jeff zachęcił mnie, abym zwrócił się do zarządu OWASP z wnioskiem o uruchomienie tego przedsięwzięcia jako nowego projektu. Kilka tygodni później zarząd wyraził zgodę, a ja ogłosiłem powstanie projektu, udostępniając jednocześnie link do dopracowanej wersji listy, którą wcześniej przygotowałem przy wsparciu ChatGPT. To, co miało być jedynie niszowym projektem badawczym i ciekawostką, przerodziło się w coś znacznie większego. Gdy ogłosiłem powstanie projektu na swoim prywatnym profilu w serwisie LinkedIn, liczyłem na to, że uda mi się znaleźć kilkanaście osób o podobnych zainteresowaniach, które chciałyby zgłębić mało znany wówczas temat bezpieczeństwa modeli LLM. Jak się okazało, mój pierwszy wpis na blogu uzyskał prawie 10 000 wyświetleń, a w kolejnych tygodniach setki osób zgłosiły chęć dołączenia do zespołu ekspertów. Niniejsza książka nie jest produktem organizacji OWASP, a zawarte w niej luki i zagrożenia nie pokrywają się dokładnie z żadną publicznie dostępną wersją listy „OWASP Top 10" dla aplikacji wykorzystujących duże modele językowe (LLM). Przedstawiam tu własne spojrzenie na te zagrożenia. Niemniej jednak moje przemyślenia i wiedza w tym zakresie wynikają w dużej mierze z pracy przy kierowaniu tym projektem oraz z procesu tworzenia i pierwszej publikacji listy „OWASP Top 10" dla aplikacji LLM. Od tamtej pory wiele osób pytało mnie o szczegóły prowadzenia tego przedsięwzięcia oraz o to, w jaki sposób udało nam się stworzyć tak istotne i wpływowe zestawienie w tak krótkim czasie. Zanim zatem przejdziemy do analizy poszczególnych zagrożeń i luk, przybliżę nieco kulisy powstania projektu w ramach OWASP, poświęconego aplikacjom opartym na modelach LLM.

O OWASP

Open Worldwide Application Security Project (OWASP) to organizacja non-profit, której celem jest poprawa bezpieczeństwa oprogramowania. Założona w 2001 roku, stanowi platformę dla ekspertów ds. bezpieczeństwa, umożliwiającą wymianę wiedzy oraz najlepszych praktyk w dziedzinie bezpieczeństwa aplikacji internetowych – od podatności na poziomie aplikacji po pojawiające się nowe zagrożenia. Obecnie organizacja zrzesza dziesiątki tysięcy aktywnych członków i posiada ponad 250 lokalnych oddziałów na całym świecie. OWASP opiera się na

społeczności i zachęca wolontariuszy do udziału w różnorodnych projektach, obejmujących tworzenie dokumentacji, narzędzi oraz prowadzenie forów dyskusyjnych. Organizacja działa w modelu open source, dzięki czemu jej zasoby są ogólnodostępne dla każdego. Na przestrzeni lat OWASP zyskało duże uznanie w środowisku specjalistów ds. bezpieczeństwa, a opracowane przez nie wytyczne i narzędzia są w wielu obszarach uznawane za standardy branżowe. Oprócz flagowego zestawienia „Top 10" dla aplikacji internetowych (regularnie aktualizowanego – ostatnia aktualizacja miała miejsce w 2021 roku), w ramach OWASP powstały również wyspecjalizowane listy „Top 10", w tym:

OWASP Mobile Top 10

Wskazuje kluczowe zagrożenia dla aplikacji mobilnych na systemy Android i iOS, takie jak niezabezpieczone przechowywanie danych, niewystarczająca kryptografia czy brak bezpieczeństwa komunikacji.

OWASP API Security Top 10

Koncentruje się na zagrożeniach specyficznych dla interfejsów API, takich jak nieprawidłowe zarządzanie zasobami czy naruszenia bezpieczeństwa na poziomie obiektów (Broken Object Level Security).

OWASP IoT Top 10

Identyfikuje główne problemy związane z bezpieczeństwem w obszarze Internetu Rzeczy (IoT), w tym niezabezpieczone usługi sieciowe, brak odpowiednich zabezpieczeń fizycznych oraz luki w oprogramowaniu lub oprogramowaniu układowym (firmware).

OWASP Cloud Native Top 10

Skupia się na zagrożeniach dla aplikacji typu „cloud native" (natywnych dla chmury), obejmujących m.in. ujawnienie danych, błędy w uwierzytelnianiu oraz niebezpieczne konfiguracje wdrożeniowe. OWASP Top 10 for Serverless odnosi się do kwestii bezpieczeństwa charakterystycznych dla architektury bezserwerowej (serverless) – modelu coraz popularniejszego, lecz wiążącego się z określonym ryzykiem.

OWASP Top 10 Privacy Risks

Promuje dobre praktyki w zakresie ochrony prywatności, poruszając kwestie takie jak brak szyfrowania danych czy niewystarczający poziom audytowania i rejestrowania zdarzeń (logowania).

Projekt „Top 10 dla aplikacji wykorzystujących LLM"

W ciągu tygodnia od opublikowania ogłoszenia o utworzeniu projektu „Top 10 dla aplikacji wykorzystujących LLM" dołączyło do niego ponad 200 osób, a my zorganizowaliśmy spotkanie inauguracyjne za pośrednictwem platformy Zoom. Podczas tego pierwszego spotkania przedstawiłem wizję tego, co – jak miałem nadzieję – uda nam się osiągnąć, oraz zaproponowałem ambitny harmonogram: stworzenie pierwszej wersji listy w ciągu ośmiu tygodni. Opracowanie typowej listy „OWASP Top 10" może zająć rok lub dłużej, jednak uznaliśmy, że w tej dynamicznie rozwijającej się dziedzinie istnieje tak pilne zapotrzebowanie na tego typu zasoby, iż musimy działać znacznie szybciej. Zdecydowaliśmy się prowadzić projekt w dwutygodniowych sprintach,

w duchu metodologii Agile. Ponieważ większość ekspertów w grupie była zaznajomiona z podejściem Agile, wszyscy szybko przystosowali się do narzuconego tempa.

Realizacja projektu

Pierwszy sprint poświęciliśmy na burzę mózgów i wymianę uwag. Wszyscy przeanalizowali pierwotną wersję listy, którą nazwałem wersją 0.1. Zawierała ona wiele niedociągnięć, na które członkowie zespołu bardzo zdecydowanie zwracali uwagę. Równolegle rozpoczęliśmy tworzenie strony typu wiki, gromadzącej odnośniki do wszelkich materiałów dotyczących bezpieczeństwa modeli LLM, na jakie natrafili uczestnicy. Okazało się, że na ten temat napisano już całkiem sporo, jednak po raz pierwszy ktoś zebrał te informacje w jednym, łatwo dostępnym miejscu. Ten nowy, starannie wyselekcjonowany zbiór zasobów stanowił pierwszy sukces grupy. Drugi sprint miał na celu opracowanie kolejnej wersji listy. Tym razem nie była ona już dziełem jednej osoby wspieranej przez sztuczną inteligencję, lecz efektem zbiorowej wiedzy naszego zespołu ekspertów, który z tygodnia na tydzień stawał się coraz liczniejszy. W pierwszym tygodniu grupa skupiła się na generowaniu pomysłów do zestawienia „Top 10". Udostępniliśmy szablon i poprosiliśmy uczestników o zgłaszanie potencjalnych podatności. W ciągu tego tygodnia przygotowaliśmy 43 szczegółowe opisy możliwych obszarów zagrożeń. Następnie przeprowadziliśmy dwie rundy głosowania za pomocą formularzy Google, wykorzystując zbiorową wiedzę zespołu do wyłonienia 10 najważniejszych pozycji, które opublikowaliśmy jako wersję 0.5. Ta wersja była znacznie bardziej szczegółowa i wyczerpująca niż wersja 0.1. Pozytywny odbiór ze strony szerszej społeczności dodał grupie energii do dalszej pracy. Kolejny sprint poświęcono na dopracowanie poszczególnych punktów. Utworzyliśmy kanały na platformie Slack dla każdego rodzaju podatności i wyznaczyliśmy osoby odpowiedzialne za poszczególne zagadnienia. Następnie podzespoły liczące od 10 do 30 osób rozbudowały i udoskonaliły każdy z punktów. Przeprowadziliśmy również głosowanie, w którym mógł wziąć udział cały zespół, aby wskazać słabsze elementy wymagające większej uwagi. W trakcie prac zauważyliśmy, że niektóre punkty się pokrywają, więc połączyliśmy je. Dzięki temu zwolniło się miejsce na włączenie zagadnień, które wcześniej nie załapały się na listę 10 najważniejszych pozycji. Efektem tego sprintu była wersja 0.9 zestawienia. Co ciekawe, wersja 0.9 była o około 33% krótsza pod względem liczby słów niż wersja 0.5; dodatkowy czas i dopracowywanie treści pozwoliły zespołom skoncentrować się na istocie zagadnień, dzięki czemu opisy stały się bardziej konkretne i zwięzłe. Na koniec przeprowadziliśmy jeszcze jeden sprint, aby przejrzeć, skorygować i uporządkować każdy punkt. Zebraliśmy kolejną porcję uwag za pośrednictwem formularzy Google, upewniając się, że wszystko jest gotowe. Na tym etapie mieliśmy już wyznaczoną osobę odpowiedzialną za projekt graficzny, która przygotowała całość dokumentu w formie atrakcyjnego pliku PDF, gotowego do publikacji.

Odbiór

Mój wpis na LinkedIn, zapowiadający wersję 1.0 zestawienia, zanotował ponad 40 000 wyświetleń. A to nie uwzględnia licznych publikacji przygotowanych przez członków grupy na ich własnych stronach i blogach. W dniach następujących po ogłoszeniu informację podchwycili dziennikarze; pojawiła się ona w mediach takich jak Wired, SD Times, The Register, Infosecurity Magazine oraz Diginomica. Można śmiało powiedzieć, że w ciągu zaledwie kilku pierwszych tygodni o naszej pracy dowiedziały się setki tysięcy osób. Co więcej – poza samym zasięgiem – ogromne wrażenie zrobił na mnie niezwykle pozytywny odzew. Co więcej, pierwsze agencje rządowe w USA i Europie zaczęły powoływać się na nasz dokument, traktując go jako materiał źródłowy. Choć wszyscy eksperci w naszym zespole byli zgodni, że pozostaje jeszcze wiele do zrobienia, okazało się, że świat tak bardzo potrzebował wskazówek w tym obszarze, iż nasz

dokument trafił w punkt. Mimo wielu pytań i uwag, wszyscy zaangażowani byli zadowoleni z efektów i dumni z wykonanej pracy.

Klucze do sukcesu

Wiele osób pytało mnie, w jaki sposób udało nam się tak szybko doprowadzić projekt do pomyślnego finału. Z perspektywy czasu uważam, że złożyło się na to kilka czynników. Dzielę się nimi tutaj, mając nadzieję, że przydadzą się innym osobom prowadzącym podobne przedsięwzięcia. Bez wątpienia ogromną rolę odegrał odpowiedni moment. Fala zainteresowania wielkimi modelami językowymi (LLM), która pojawiła się po premierze ChatGPT, była ogromna. Przykuła moją uwagę i wzbudziła entuzjazm u niezliczonej rzeszy innych osób. Dzięki temu udało się zgromadzić liczne i różnorodne grono ekspertów, a część z nich poświęciła wiele godzin na pracę nad projektem w warunkach napiętych terminów. Kluczowe było posiadanie od samego początku jasnego planu i harmonogramu. Choć na starcie moja wiedza na temat bezpieczeństwa modeli LLM była ograniczona, mam duże doświadczenie w prowadzeniu złożonych projektów z udziałem wielu osób. Stworzenie przejrzystej mapy drogowej, z wyznaczonymi etapami i terminarzem, pozwoliło wszystkim zrozumieć, co robimy i w jakim czasie. Świadomość, że cel jest na wyciągnięcie ręki, stanowiła dla zespołu dodatkową motywację. Co dwa tygodnie organizowaliśmy spotkania online przez platformę Zoom, a nagrania z nich udostępnialiśmy w serwisie YouTube dla osób, które nie mogły wziąć w nich udziału na żywo. Spotkania i nagrania miały kluczowe znaczenie dla koordynacji zespołu rozproszonego po całym świecie. Niezbędny okazał się początkowy etap burzy mózgów – swobodny, lecz krótki. Bezpieczeństwo modeli językowych (LLM) było tak nową dziedziną, że poświęcenie pierwszych dwóch tygodni na swobodne zgłaszanie pomysłów i dyskusje na platformie Slack było strzałem w dziesiątkę. Pozwoliło nam to również zebrać i udostępnić wszystkim uczestnikom bazę dotychczasowych badań w tym obszarze. Dzięki temu każdy członek projektu mógł od samego początku korzystać z najlepszych dostępnych opracowań. Równie ważne było jednak to, by etap ten nie trwał zbyt długo. Ograniczenie burzy mózgów do dwóch tygodni pozwoliło utrzymać tempo prac i szybko przejść do fazy tworzenia. Widziałem już projekty, które utknęły na etapie burzy mózgów, tracąc w końcu zainteresowanie uczestników. Utworzenie zespołu rdzeniowego (core team) nie było częścią mojego pierwotnego planu, ale okazało się kluczowe. Dysponowanie szerokim gronem ekspertów było ogromnym atutem – do momentu publikacji wersji 1.0 grupa rozrosła się do blisko 500 osób. Zarządzanie tak licznym zespołem byłoby niemożliwe. W pierwszych tygodniach projektu szukałem osób aktywnych i posiadających odpowiednią wiedzę. Zwróciłem się do kilkunastu z nich z propozycją dołączenia do głównego zespołu kierowniczego. Wyjaśniłem, że wiąże się to z dodatkową pracą, ale pozwoli im znaleźć się w samym centrum projektu. Nie przewidziano za to żadnego konkretnego wynagrodzenia. Większość osób zgodziła się bez wahania. Uważam, że docenienie tych ludzi i formalne zwrócenie się do nich o wsparcie zmotywowało ich do poświęcenia projektowi więcej czasu i energii. Byli w pełni zaangażowani! Krótkie sprinty z konkretnymi, widocznymi efektami pracy to fundament metodologii Agile – i w tym przypadku sprawdziły się znakomicie. Dzięki modelowi „Agile Release Train" mogłem nieustannie posuwać prace do przodu, nawet gdy pojawiały się rozbieżne opinie. Jeśli ktoś miał wątpliwości dotyczące jakiegoś obszaru, nie pozwalaliśmy, by nas to zatrzymało; przyjmowaliśmy uwagę do wiadomości i umawialiśmy się na jej rozwiązanie w kolejnym sprincie. Kiedy dotarliśmy do wersji 1.0, wciąż istniały kwestie, które niektórzy chcieli rozwinąć bardziej – ustaliliśmy więc po prostu, że powstaną kolejne wersje zestawienia. Miał to być „żywy" dokument, a najważniejsze było dostarczenie jego wersji twórcom, którzy go potrzebowali.

Ta książka a zestawienie „Top 10"

Jak już wspomniałem, niniejsza publikacja nie jest oficjalnym produktem fundacji OWASP. Niemniej jednak współpraca z tym zespołem wywarła ogromny wpływ na moje rozumienie bezpieczeństwa modeli językowych (LLM) oraz moje podejście do tego zagadnienia. Dzięki temu wiele wskazówek zawartych w kolejnych rozdziałach odzwierciedla wiedzę i doświadczenie fantastycznego zespołu, który tworzy i rozwija projekt „Top 10". Czytelnicy mogą mieć zatem pewność, że otrzymują porady będące efektem pracy nie jednego autora, lecz szerszej społeczności ekspertów. W następnych rozdziałach przyjrzymy się najważniejszym zagrożeniom i podatnościom dotyczącym modeli LLM. Choć omawiane tu ryzyka w dużej mierze pokrywają się z tymi ujętymi w zestawieniu OWASP Top 10, nie stanowią one jego dokładnego odzwierciedlenia. Podczas gdy lista OWASP Top 10 stanowi zwięzły przegląd kluczowych obszarów, w tej książce zagłębiamy się w poszczególne zagrożenia, sposoby przeciwdziałania im oraz szczegółowe studia przypadków z rzeczywistych scenariuszy. Do zestawienia OWASP Top 10 powrócimy w rozdziale 10., gdzie krótko omówimy jego wersję z 2023 roku i wskażemy, jak odnosi się ona do treści poszczególnych rozdziałów tej książki. Pokażemy również, w jaki sposób wykorzystać ramy OWASP Top 10 do dokumentowania i udostępniania analiz podatności oraz skutecznych ataków. W rozdziale 3. przyjrzymy się ogólnej strukturze typowych aplikacji opartych na modelach LLM, omawiając granice zaufania oraz związane z nimi niebezpieczeństwa. Kolejne rozdziały poświęcimy analizie konkretnych obszarów ryzyka, podatności, ataków i praktycznych przykładów, co pozwoli Ci opracować własną strategię zabezpieczania wykorzystywanych przez Ciebie rozwiązań. Zaczynajmy!