Zapewnienie bezpieczeństwa aplikacji Android

mgr inż. Stanisław Lota

Wprowadzenie

Utrzymanie bezpieczeństwa aplikacji jest niezwykle ważne, ponieważ tworzymy aplikacje dostępne dla użytkowników.

System Android został zaprojektowany w taki sposób, aby zazwyczaj można było tworzyć aplikacje z domyślnymi uprawnieniami systemowymi i plikami oraz unikać trudnych decyzji dotyczących zabezpieczeń.

Wprowadzenie

Podczas tworzenia aplikacji Android trzeba wziąć pod uwagę wiele różnych aspektów dotyczących bezpieczeństwa.

Jednak funkcje zapewniania bezpieczeństwa na platformie Android są bogate i na ich podstawie można opracować naprawdę silne mechanizmy zabezpieczeń, które zmniejszą częstotliwość i wpływ problemów z bezpieczeństwem aplikacji. Również w Android Jetpack znaleźć możemy kilka bibliotek, aby zapewnić większe bezpieczeństwo danych aplikacji.

Główne zasady minimalizacji zagrożeń

Programista powinien przede wszystkim zachować ostrożność w zakresie wczytywania lub uruchamiania plików wykonywalnych pochodzących z niezaufanych źródeł. Aplikacja powinna do niezbędnego minimum ograniczyć liczbę eksportowanych komponentów. Im mniej wyeksportowanych komponentów, tym lepiej.

Sprawdzenie, czy wszystkie ścieżki kodu w komponentach aplikacji udostępniają jedynie zamierzoną funkcjonalność powinno ograniczyć możliwość zagrożeń.

Główne zasady minimalizacji zagrożeń

Programiści powinni ograniczać do absolutnego niezbędnego minimum ilości przechowywanych danych użytkownika. Jeżeli nie ma absolutnej konieczności przechowywania danych przez aplikację, po prostu nie przechowujmy ich.

To obejmuje przechowywanie danych zarówno w prywatnym katalogu danych aplikacji, jak i na karcie SD czy też w zewnętrznej bazie danych. W przypadku przechowywania danych ważne jest, aby przechowywać je w zaszyfrowanym formacie.

Jednym z najpopularniejszych algorytmów szyfrowania używanych obecnie przez programistów jest AES, skrót od Advanced Encryption Standard, o rozmiarze klucza 256 bitów.

Główne zasady minimalizacji zagrożeń

Aplikacja otrzymująca informacje z karty SD, internetu, sieci bezprzewodowej, Bluetooth lub z innego źródła niebędącego pod bezpośrednią kontrolą aplikacji powinna mieć zaimplementowany mechanizm uwierzytelniania.

To uwierzytelnianie powinno mieć postać sprawdzenia podpisu cyfrowego dla tych informacji, zastosowania pewnego rodzaju szyfrowania potwierdzającego tożsamość źródła przekazującego dane lub też jakikolwiek inny schemat weryfikacji.

Główne zasady minimalizacji zagrożeń

Aplikacje powinny wymagać minimalnych uprawnień niezbędnych do prawidłowego funkcjonowania aplikacji. Poza tym pomaga w zapewnianiu większego bezpieczeństwa użytkownikom i zmniejsza wielkość szkód, jakie mogą zostać wyrządzone po zakończonym sukcesem ataku na aplikację.

Wiele aplikacji żąda zbyt wiele uprawnień.

Główne zasady minimalizacji zagrożeń

Jeśli aplikacja prosi o pozwolenie na dostęp do lokalizacji, pomóż użytkownikom w podjęciu świadomej decyzji.

Jeśli aplikacja zbiera informacje o lokalizacji, wyjaśnij użytkownikom, w jaki sposób aplikacja wykorzystuje te informacje, aby zapewnić im określone korzyści.

Jeśli aplikacja może obsługiwać przypadki użycia bez konieczności podawania danych o lokalizacji, nie żądaj żadnych uprawnień do lokalizacji.

Główne zasady minimalizacji zagrożeń

Przed wypuszczeniem aplikacji należy poświęcić nieco czasu na rozpakowanie pakietu APK i sprawdzenie, co znajduje się wewnątrz.

W ten sposób unikniemy przypadkowego udostępnienia niepotrzebnych plików.

Nie chcemy przecież, aby ktokolwiek był w stanie otrzymać plik zawierający dane uwierzytelniające SSH do serwera testowego np. Firebase używanego podczas pracy nad aplikacją lub do innych poufnych plików.

Główne zasady minimalizacji zagrożeń

Upewnić należy się, czy nie będzie można przeprowadzać ataków typu tapjacking na wszelkie komponenty View aplikacji oraz sprawdzić czy aplikacje nie są podatne na ataki typu SQL injection.

Aby ataki phishingowe były bardziej widoczne i miały mniejsze szanse powodzenia, zminimalizujmy częstotliwość pytań o poświadczenia użytkownika. Jeśli to możliwe, nie przechowuj nazw użytkowników i haseł na urządzeniu. Można dodatkowo użyć tokena autoryzacyjnego i odświeżyć go.

Tapjacking

Tapjacking to mobilny odpowiednik znanej z Internetu luki w zabezpieczeniach o nazwie clickjacking, która wyświetla specjalnie spreparowany interfejs użytkownika nad czynnością innej aplikacji, aby skłonić użytkownika do kliknięcia tego, czego nie zamierzał kliknąć.

To jest możliwe dzięki funkcji interfejsu użytkownika o nazwie toast, który jest najczęściej używany do wyświetlania użytkownikowi niewielkich fragmentów informacji, z którymi nie ma on możliwości prowadzenia interakcji.

Tapjacking

Najniebezpieczniejsze jest to, że kiedy użytkownik spróbuje coś nacisnąć na tej nowej „czynności”, wprowadzone przez niego dane wejściowe mogą zostać otrzymane przez czynność znajdującą się pod toastem, np. do otworzenia czynności sklepu Google Play i zainstalowania aplikacji.

Główne zasady minimalizacji zagrożeń

Należy upewnić się, że czynności logowania nie pozwalają na wyświetlenie uwierzytelnionych czynności przed pomyślnym zakończeniem procesu uwierzytelniania. Sprawdzić należy wszystkie pola, w których użytkownik wpisuje hasło, są prawidłowo maskowane.

Niewystarczająca walidacja danych wejściowych jest jednym z najczęstszych problemów związanych z bezpieczeństwem aplikacji, niezależnie od platformy, na której działają.

Należy upewnić się, że klucze szyfrowania zostały wygenerowane z zastosowaniem najlepszych praktyk w tym zakresie.

Główne zasady minimalizacji zagrożeń

Możliwe jest też dodanie dodatkowego mechanizmu zabezpieczenia warstwy transportowej, na przykład przypięcia certyfikatu SSL, w całej komunikacji z internetem. Należy tym samym uniemożliwić użycie nieprawidłowych certyfikatów SSL.

Pliki zawierające poufne dane powinny znajdować się w katalogu app-private w pamięci wewnętrznej. Ważne jest też upewnienie się o braku możliwości utworzenia kopii zapasowej aplikacji za pomocą funkcjonalności oferowanej przez ADB. Należy też upewnić się, że aplikacja nie pozwala na późniejsze debugowanie.

Główne zasady minimalizacji zagrożeń

Podczas procesu tworzenia aplikacji na Androida kod źródłowy musi być chroniony. Dlatego programiści powinni uczynić go niezrozumiałym zarówno dla dekompilatora, jak i ludzi. Zaciemnianie kodu to technika przekształcania kodu źródłowego w coś, co jest trudne do odczytania dla ludzi. Odbywa się to głównie za pomocą zautomatyzowanych narzędzi przed zbudowaniem aplikacji.

Nie zwiększa bezpieczeństwa samego kodu. Jedynym celem jest skomplikowanie procesu inżynierii wstecznej kodu źródłowego ze skompilowanej aplikacji.

ProGuard

ProGuard jest darmowym, najczęściej wykorzystywanym narzędziem przy zaciemnianiu i optymalizacji kodu w Androidzie. Ponadto Proguard jest łatwo dostępny, ponieważ jest instalowany wraz z zestawem narzędzi Android SDK.

ProGuard stosuje się do zabezpieczania aplikacji przed różnymi atakami, w tym modyfikacjami, inżynierią wsteczną, klonowaniem i nieautoryzowanym użyciem.

Inżynieria odwrotna(wsteczna) Aplikacji

Inżynieria odwrotna aplikacji umożliwia uzyskanie dostępu do pliku APK wybranej aplikacji i odczytania kodu programu. Przykładem programu może być narzędzie dex2jar, które jest przeznaczone do przeprowadzania konwersji plików Android DEX na postać zwykłych plików klas Javy.

Program poprawy bezpieczeństwa aplikacji

Program poprawy bezpieczeństwa aplikacji to usługa świadczona deweloperom aplikacji Google Play w celu poprawy bezpieczeństwa ich aplikacji.

Program zawiera wskazówki i zalecenia dotyczące tworzenia bezpieczniejszych aplikacji oraz identyfikuje potencjalne ulepszenia zabezpieczeń po przesłaniu aplikacji do Google Play. Do tej pory program ułatwił programistom naprawienie ponad 1 000 000 aplikacji w Google Play.

Program poprawy bezpieczeństwa aplikacji

Zanim jakakolwiek aplikacja zostanie zaakceptowana w Google Play, Google skanuje ją pod kątem bezpieczeństwa, w tym potencjalnych problemów z bezpieczeństwem. Stale skanowane jest również ponad milion aplikacji w Google Play w poszukiwaniu dodatkowych zagrożeń.

Jeśli aplikacja zostanie oznaczona jako potencjalny problem z zabezpieczeniami, Google natychmiast powiadamia programistę aby pomóc szybko rozwiązać problem i zapewnić bezpieczeństwo użytkownikom.

Zadanie

1.Podstawy pracy z ProGuardem :

https://www.raywenderlich.com/7449-getting-started-with-proguard

Implementacja ProGuarda : https://riptutorial.com/android/example/21062/enabling-proguard-with-a-custom-obfuscation-configuration-file

2.Przygotuj prezentację/dokument na temat ostatnich zagrożeń związanych z platformą Android w raz z opisem zagrożeń, sposobem ich neutralizacji oraz nazwą i oznaczeniem ataku. Wykorzystaj np. stronę sans.org bądź w biuletynie bezpieczeństwa na stronie Android.com