KRYPTOGRAFIA

Ochrona danych podczas ich przesyłania przez Internet lub przechowywania na komputerze jest jednym z najważniejszych zadań specjalisty ds. bezpieczeństwa sieci. Firmy, jak i użytkownicy, nie chcą, aby inni mogli przeglądać poufne dokumenty i pliki. W tym module przyjrzysz się technologiom kryptograficznym, których specjaliści ds. bezpieczeństwa używają do ochrony danych firmy. Zobaczysz, jak informacje mogą zostać przekonwertowane na nieczytelny format i jak tylko osoby posiadające prawidłowy klucz lub „dekoder" mogą odczytać wiadomość. Przyjrzysz się również atakom kryptograficznym i niektórym narzędziom używanym do przeprowadzania tych ataków.

ZROZUMIENIE PODSTAW KRYPTOGRAFII

Kryptografia to proces konwersji tekstu jawnego, który jest tekstem czytelnym, na tekst zaszyfrowany, który jest tekstem nieczytelnym lub zaszyfrowanym. Kryptografia może być stosowana w przypadku danych, które ludzie lub organizacje chcą zachować w tajemnicy lub danych, które powinny być dostępne tylko dla niektórych użytkowników. Innymi słowy, kryptografia jest stosowana w celu ukrycia informacji przed nieautoryzowanymi użytkownikami. Deszyfrowanie to proces konwersji tekstu zaszyfrowanego z powrotem na tekst jawny (nazywany również tekstem jawnym). Jako dziecko mogłeś mieć pierścień dekodujący z pudełka płatków śniadaniowych, którego mogłeś użyć do napisania listu do przyjaciela w tajnym kodzie. Jeśli twój przyjaciel miał taki sam pierścień dekodujący, mógł zdekodować twój list i go przeczytać.

Historia kryptografii

Kryptografia istnieje od tysięcy lat. Na przykład niektóre egipskie hieroglify na starożytnych pomnikach były szyfrowane. Części Księgi Jeremiasza zostały napisane przy użyciu szyfru lub klucza znanego jako Atbash. Ten prosty szyfr odwracał alfabet — na przykład zastępując A przez Z — i tylko osoba znająca odwzorowanie mogła rozszyfrować (odszyfrować) wiadomość. Ten rodzaj kryptografii nazywa się szyfrem podstawieniowym. Juliusz Cezar opracował podobny szyfr podstawieniowy do szyfrowania wiadomości, przesuwając każdą literę alfabetu o trzy pozycje. Na przykład A było kodowane jako litera D. Wydaje się, że każda kultura stosowała jakąś formę ukrywania lub maskowania tekstu jawnego. Kamasutra, napisana przez indyjskiego uczonego Vatsyayanę prawie 2000 lat temu, zaleca mężczyznom i kobietom naukę i praktykowanie sztuki kryptografii, którą definiuje jako „sztukę rozumienia pisma w szyfrze i pisania dzieł w szczególny sposób". Dopóki ludzie będą próbowali tworzyć algorytmy szyfrowania w celu ochrony danych, inni będą starali się je złamać. Badanie łamania algorytmów szyfrowania nazywa się kryptoanalizą. Jest ona nauczana na uniwersytetach i przez agencje rządowe, ale hakerzy również uważają wyzwanie złamania algorytmu szyfrowania za intrygujące i nadal zmuszają twórców algorytmów szyfrowania do przekraczania granic w poszukiwaniu trudniejszych do złamania algorytmów. Kiedy opracowywany jest nowy algorytm szyfrowania, kryptoanaliza jest wykorzystywana w celu zapewnienia, że złamanie kodu jest niemożliwe lub zajęłoby tak dużo czasu i zasobów, że próba byłaby niepraktyczna. Innymi słowy, jeśli złamanie algorytmu szyfrowania wymaga mocy obliczeniowej superkomputera o wartości 500 milionów dolarów i 500 lat, algorytm można uznać za wystarczająco bezpieczny do celów praktycznych. Jednak kiedy kryptoanaliza jest możliwa przy rozsądnej mocy obliczeniowej, atak na algorytm jest uważany za „praktyczny", a algorytm za słaby.

Maszyny wojenne

Najsłynniejszym urządzeniem szyfrującym była maszyna Enigma, opracowana przez Arthura Scherbiusa i używana przez Niemców podczas II wojny światowej. Większość książek o kryptografii omawia to urządzenie. Jak ono działało? Operator wpisywał literę, która miała zostać zaszyfrowana, a

maszyna wyświetlała znak podstawienia dla litery. Następnie operator zapisywał ten znak podstawienia i obracał wirnikiem lub przełącznikiem. Następnie wprowadzał kolejną literę i ponownie zapisywał znak podstawienia wyświetlany przez Enigmę. Gdy wiadomość była całkowicie zaszyfrowana, była przesyłana przez fale radiowe. Oczywiście wiadomość mogła zostać odszyfrowana tylko przez maszynę Enigma po drugiej stronie, która wiedziała, w jakiej pozycji przesunąć wirniki. Kod został złamany najpierw przez grupę polskich kryptografów, a następnie przez Brytyjczyków i Amerykanów. Maszyna, której brytyjscy i amerykańscy kryptolodzy użyli do złamania kodu, opracowana przez brytyjskiego matematyka Alana Turinga, nazywała się Bombe. Podczas II wojny światowej Japończycy opracowali inną godną uwagi maszynę wojenną, zwaną Purple Machine, która wykorzystywała techniki odkryte przez Herberta O. Yardleya. Zespół kierowany przez Williama Fredericka Friedmana, kryptoanalityka armii USA, znanego jako Ojciec Kryptoanalizy USA, złamał kod. FBI zatrudniło pana Friedmana i jego żonę do pomocy w odszyfrowywaniu wiadomości radiowych wysyłanych przez przemytników i przemytników w latach 30. XX wieku. Te kody szyfrujące okazały się trudniejsze i bardziej złożone niż te używane w czasie wojny. Głównym celem kryptografii jest ukrywanie informacji przed innymi, a istnieją metody ukrywania danych, które nie wykorzystują szyfrowania. Jedną z nich jest steganografia, sposób ukrywania danych na widoku w obrazach, grafikach lub tekście. Na przykład zdjęcie mężczyzny stojącego przed Białym Domem może zawierać ukrytą wiadomość, która przekazuje szpiegowi informacje o ruchach wojsk. W 1623 roku sir Francis Bacon zastosował formę steganografii, ukrywając fragmenty informacji w odmianach czcionki używanej w książkach.

Tworzenie szyfru podstawieniowego

Czas trwania: 30 minut

Cel: Naucz się, jak utworzyć szyfr podstawieniowy i zaszyfrować wiadomość.

Opis: Aby lepiej zrozumieć kryptografię, podziel uczniów na grupy. Każda grupa powinna utworzyć krótką wiadomość nie dłuższą niż pięć słów w tekście jawnym. Twoja grupa szyfruje wiadomość szyfrem podstawieniowym, a następnie inne grupy (deszyfrujący) próbują odszyfrować wiadomość. Każda grupa powinna utworzyć jedną zaszyfrowaną wiadomość i odszyfrować każdą wiadomość utworzoną przez inne grupy.

1. Grupa szyfrująca pisze pięciowyrazową wiadomość na czystej kartce papieru.

2. Utwórz szyfr podstawieniowy, aby zaszyfrować wiadomość. Na przykład każdy znak można przesunąć o trzy znaki, tak aby na przykład litera A stała się literą D.

3. Zapisz wiadomość zaszyfrowaną, którą utworzyłeś za pomocą szyfru swojej grupy.

4. Gdy otrzymasz polecenie, aby to zrobić, przekaż swoje wiadomości zaszyfrowane innym grupom w celu odszyfrowania.

5. Gdy grupa odszyfruje wiadomość, lider grupy powinien powiedzieć „Gotowe!”, aby instruktor mógł zobaczyć, która grupa ukończyła zadanie najszybciej.

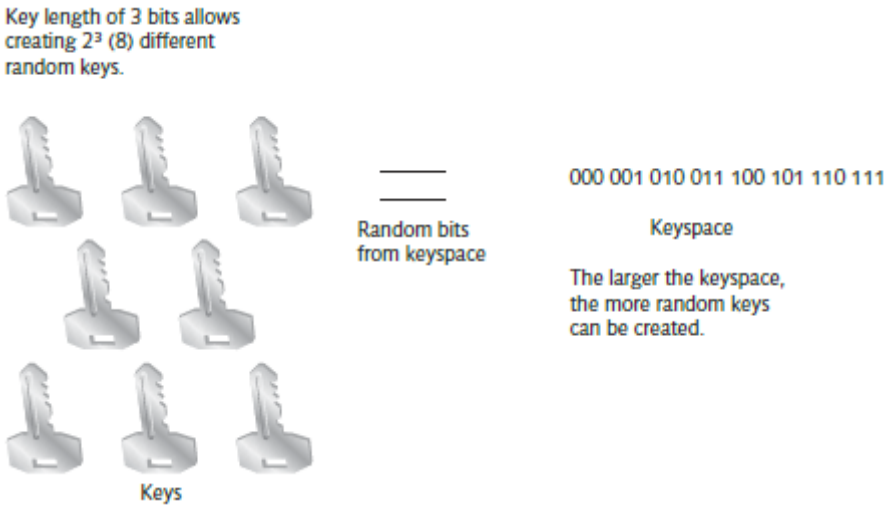
6. Gdy wszystkie grupy miały okazję spróbować odszyfrować wiadomości, omów szyfry utworzone przez każdą grupę.

BAJTY BEZPIECZEŃSTWA

Czy wiesz, że Thomas Jefferson wynalazł szyfr kołowy w XVIII wieku, który Marynarka Wojenna przebudowała i wykorzystała podczas II wojny światowej, nazywając go M-138-A? Im więcej rzeczy się zmienia, tym bardziej pozostają takie same.

ZROZUMIENIE ALGORYTMÓW SYMETRYCZNYCH I ASYMETRYCZNYCH

Nowoczesna kryptografia wykorzystuje algorytmy szyfrowania do szyfrowania danych, transakcji bankowych, innych transakcji online wykorzystujących protokół HTTPS, komunikacji bezprzewodowej (szyfrowanie WEP i WPA) itd. Algorytm szyfrowania to funkcja matematyczna lub program, który działa z kluczem. Siła algorytmu i tajność klucza decydują o tym, jak bezpieczne są zaszyfrowane dane. W większości przypadków algorytm nie jest tajemnicą; jest znany opinii publicznej. Tajemnicą jest klucz. Klucz to sekwencja losowych bitów wygenerowana z zakresu dopuszczalnych wartości zwana przestrzenią kluczy, która jest zawarta w algorytmie. Im większa przestrzeń kluczy, tym więcej kluczy można utworzyć. Na przykład algorytm z 256-bitową przestrzenią kluczy ma 2256 możliwych kluczy. Im więcej losowych kluczy można utworzyć, tym trudniej jest hakerom odgadnąć, który klucz został użyty do zaszyfrowania danych. Oczywiście użycie tylko ośmiu losowych kluczy (jak pokazano na rysunku) sprawia, że algorytm jest zbyt łatwy do złamania i jest pokazany jedynie jako przykład.



Mówiąc prościej, kryptosystem konwertuje tekst jawny na tekst zaszyfrowany. Większość prób złamania kryptosystemu wiąże się ze zgadywaniem klucza. Niezależnie od tego, jak silny jest algorytm lub jak duża jest przestrzeń kluczy, jeśli klucz nie jest chroniony, atakujący może odszyfrować wiadomość. Jeśli użytkownicy udostępnią komuś swoje klucze, wszystkie zakłady są nieważne. Tabela podsumowuje trzy typy algorytmów.

Typ algorytmu: Opis

Symetryczny: Używa jednego klucza do szyfrowania i odszyfrowywania danych. Zarówno nadawca, jak i odbiorca muszą uzgodnić klucz przed przesłaniem danych. Algorytmy symetryczne obsługują poufność, ale nie uwierzytelnianie i niezaprzeczalność (omówione później w „Algorytmach asymetrycznych"). Są jednak co najmniej 1000 razy szybsze niż algorytmy asymetryczne.

Asymetryczny: Używa dwóch kluczy: jednego do szyfrowania danych i jednego do odszyfrowywania danych. Algorytmy asymetryczne obsługują uwierzytelnianie i niezaprzeczalność, ale są wolniejsze niż algorytmy symetryczne. Algorytmy asymetryczne są również znane jako kryptografia klucza publicznego.

Hashowanie: Używane do weryfikacji. Hashowanie przyjmuje dane wejściowe o zmiennej długości i konwertuje je na ciąg wyjściowy o stałej długości, nazywany wartością skrótu lub streszczeniem wiadomości.

Posiadanie umiejętności kryptologa nie jest konieczne dla testerów bezpieczeństwa, ale zrozumienie podstawowych terminów kryptologicznych jest pomocne. Na przykład, jeśli widzisz opis „Blowfish to szyfr blokowy o rozmiarze klucza do 448 bitów", chcesz wiedzieć wystarczająco dużo, aby zrozumieć, co to oznacza. Poniższe sekcje dokładniej omawiają te typy algorytmów i wyjaśniają niektóre podstawowe terminy.

Algorytmy symetryczne

Kryptosystemy wykorzystujące algorytmy symetryczne mają jeden klucz, który szyfruje i odszyfrowuje dane. Jeśli użytkownik chce wysłać wiadomość do współpracownika, szyfruje ją tajnym kluczem, a współpracownik, który musi mieć kopię tego samego klucza, odszyfrowuje wiadomość. Jeśli użytkownik chce zaszyfrować inną wiadomość i wysłać ją innemu współpracownikowi, musi użyć innego tajnego klucza. Jeśli w równaniu uwzględniono setki współpracowników, śledzenie którego tajnego klucza użycie staje się dużym problemem. Aby obliczyć liczbę kluczy potrzebnych do obsługi systemu symetrycznego, należy użyć wzoru n(n − 1)/2. Na przykład, jeśli pięciu użytkowników musi użyć tajnych kluczy do przesłania danych, potrzebujesz 5(5 − 1)/2 kluczy, czyli 10 kluczy. Innym problemem z tajnymi kluczami jest to, jak wysłać jeden do współpracownika odszyfrowującego Twoją wiadomość. Wysyłanie jej e-mailem może być niebezpieczne, ponieważ wiadomość może zostać przechwycona. Możesz spróbować umieścić tajny klucz na płycie CD-R lub dysku USB, ale oba nośniki mogą zostać zgubione lub skradzione. Ponieważ dwóch użytkowników współdzieli ten sam klucz w algorytmach symetrycznych, nie ma sposobu, aby dowiedzieć się, który użytkownik wysłał wiadomość. Innymi słowy, algorytmy symetryczne nie obsługują uwierzytelniania i niezaprzeczalności (opisane bardziej szczegółowo w „Algorytmach asymetrycznych"). Jak widać, algorytmy symetryczne mają pewne problemy. Są idealnymi mechanizmami do szybkiego szyfrowania dużych bloków danych i trudno je złamać, jeśli używany jest duży rozmiar klucza. Zalety algorytmów symetrycznych są następujące:

• Znacznie szybsze niż algorytmy asymetryczne

• Trudne do złamania, jeśli używany jest duży rozmiar klucza

• Tylko jeden klucz potrzebny do szyfrowania i odszyfrowywania danych

Algorytmy symetryczne mają następujące wady:

• Wymagają, aby każda para użytkowników miała unikalny klucz tajny, co utrudnia zarządzanie kluczami

• Trudne do dostarczenia kluczy bez ryzyka kradzieży

• Nie zapewniają uwierzytelniania ani niezaprzeczalności dla użytkowników

Obecnie używane są dwa typy algorytmów symetrycznych: szyfry strumieniowe i szyfry blokowe. Szyfry strumieniowe, takie jak A5/1, A5/2 i RC4, działają na tekście jawnym, jeden bit na raz. Wiadomości są traktowane jako strumień bitów, a szyfr strumieniowy wykonuje funkcje matematyczne na każdym bicie, co sprawia, że algorytmy te są świetnymi kandydatami do urządzeń szyfrujących na poziomie sprzętowym lub chipowym. Szyfry blokowe, takie jak DES, 3DES, AES i RC5, działają na blokach bitów. Te bloki są używane jako dane wejściowe do funkcji matematycznych, które wykonują podstawienie i transpozycję bitów. Czasami, gdy szyfr blokowy rozdziela dane wejściowe na bloki, musi dodać

wypełnienie, aby wypełnić dany blok. To wypełnienie sprawia, że szyfr jest podatny na ataki. Nagłośnione ataki obejmują CRIME, BEAST i Lucky 7, które są typami ataków Padding Oracle. Możesz przeczytać więcej szczegółów na temat ataków Padding Oracle na blogu Grymoire (https://grymoire.wordpress.com/2014/12/05/cbc-padding-oracle-attacks-simplified-key-concepts-and-pitfalls/). W poniższych sekcjach przyjrzysz się niektórym algorytmom symetrycznym, które stały się standardami w branży. Niezależnie od standardu, algorytmy symetryczne opierają się na jednym i tym samym kluczu do szyfrowania i odszyfrowywania danych.

Zabezpieczanie komunikacji osobistej za pomocą szyfrowania

Wymagany czas: 20 minut

Cel: Użyj Internetu, aby zbadać aplikacje do szyfrowania wiadomości osobistych.

Opis: W tej aktywności użyjesz Internetu, aby zbadać aplikacje do szyfrowania wiadomości osobistych, które mogą być używane do szyfrowania wiadomości osobistych w celu zapewnienia większej prywatności.

1. W systemie Windows uruchom przeglądarkę internetową i wprowadź adres www.tomsguide.com/reference/ best-encrypted-messaging-apps/.

2. Przeczytaj artykuł, aby poznać opcje szyfrowania komunikacji osobistej. Czy obecnie korzystasz z którejś z tych aplikacji? Czy wiedziałeś, że niektóre aplikacje (takie jak Facebook Messenger) mają opcje szyfrowania komunikacji?

3. Użyj przeglądarki internetowej, aby przeczytać artykuł pod następującym adresem: www.wired.com/story/nahoft-iran-messaging-encryption-app/.

4. Jakiego podstawowego typu kryptografii używa aplikacja Nahoft? 5. Użyj swojej ulubionej wyszukiwarki i wyszukaj aplikacje do steganografii na Androida.

6. Wybierz kilka linków, aby odpowiedzieć na poniższe pytania. Czym jest steganografia? Jakie aplikacje oferują tę możliwość? Znajdź znajomego i wypróbuj jedną z tych aplikacji.

7. Wyjdź z przeglądarki internetowej i wyloguj się z systemu Windows, aby wykonać następną czynność.

UWAGA : Mimo że DEA używa szyfrowania 64-bitowego, efektywnie używanych jest tylko 56 bitów. Osiem z 64 bitów jest używanych do parzystości (korekcji błędów).

Standard szyfrowania danych

Dyskusja na temat algorytmów symetrycznych musi obejmować Standard szyfrowania danych (DES). Narodowy Instytut Standardów i Technologii (NIST) chciał znaleźć sposób na ochronę poufnych, ale niejawnych danych, więc na początku lat 70. zaprosił dostawców do przesyłania algorytmów szyfrowania danych. Najlepszy algorytm stałby się standardową metodą szyfrowania dla agencji rządowych i firm z sektora prywatnego. IBM stworzył już 128-bitowy algorytm o nazwie Lucifer. NIST zaakceptował go jako standardowy algorytm szyfrowania; jednak Narodowa Agencja Bezpieczeństwa (NSA) chciała wprowadzić pewne modyfikacje, zanim zezwoliła na jego użycie. NSA zdecydowała się zmniejszyć rozmiar klucza ze 128 bitów do 64 bitów i nadała mu nazwę Data Encryption Algorithm (DEA). Aby było jasne, DES jest standardem, a DEA jest algorytmem szyfrowania używanym w tym standardzie. DEA nie jest najbardziej kreatywną nazwą, ale NSA prawdopodobnie uważała, że nazwa Lucifer nie ma oficjalnego rządowego brzmienia. Nie wiadomo, dlaczego NSA zmniejszyła przestrzeń kluczy algorytmu. Wiadomo, że szyfrowanie 128-bitowe jest znacznie trudniejsze do złamania niż

szyfrowanie 64-bitowe. Jak w przypadku większości rzeczy, czas odcisnął swoje piętno na DES. W 1988 r. NSA uważała, że standard jest zagrożony złamaniem ze względu na swoją długowieczność i rosnącą moc komputerów. Każdy system, bez względu na to, jak bezpieczny, jest podatny na ataki, gdy hakerzy mają lata na szukanie luk. NSA miała rację w swoim założeniu. Zwiększona moc obliczeniowa komputerów wkrótce umożliwiła złamanie szyfrowania DES. W rzeczywistości w 1998 r. zaprojektowano system komputerowy, który był w stanie złamać klucz szyfrujący w ciągu zaledwie trzech dni. Istnieją również przykłady hakerów łączących moc obliczeniową tysięcy komputerów (bez wiedzy właścicieli systemów) przez Internet w celu złamania złożonych algorytmów szyfrowania. Wielu kryptologów zbyt szybko twierdzi, że kilku superkomputerom Cray zajęłoby 200 lat, aby odkryć tajny klucz w ich algorytmach szyfrowania, podczas gdy zaledwie kilka lat ulepszeń szybkości procesora dowodzi, że można to zrobić przy użyciu tylko wydajnego laptopa i dostępu do Internetu.

Triple DES

Potrzebny był nowy standard, ponieważ DES nie był już rozwiązaniem. Triple Data Encryption Standard (3DES) służył jako szybka poprawka luk w zabezpieczeniach DES. Aby utrudnić atakującym złamanie kodu szyfrującego, 3DES wykonuje oryginalne obliczenia DES trzy razy. Opcje kluczy mogą się różnić dla każdej z trzech rund szyfrowania DES; jednak 3DES jest najsilniejszy, gdy dla każdej z nich używany jest unikalny klucz. To bardziej złożone obliczenie danych sprawia, że 3DES jest znacznie silniejszy niż DES. Jednak ta poprawa miała kompromis w zakresie wydajności. 3DES potrzebuje więcej czasu na szyfrowanie i odszyfrowywanie danych niż jego poprzednik, ale jest to niewielka cena za znacznie lepsze bezpieczeństwo.

Advanced Encryption Standard

Ostatecznie NIST zdecydował, że 3DES jest środkiem tymczasowym dla słabego algorytmu i konieczne jest opracowanie nowego standardu: Advanced Encryption Standard (AES). W 1997 r. NIST wystosował do opinii publicznej kolejną prośbę o nowy standard szyfrowania, prosząc o symetryczny szyfr blokowy obsługujący klucze 128-, 192- i 256-bitowe. Spośród pięciu finalistów NIST wybrał Rijndael, opracowany przez Joan Daemen i Vincenta Rijmena, ze względu na jego udoskonalenia w zakresie bezpieczeństwa, wydajności, efektywności i elastyczności. Pozostali czterej finaliści to MARS, RC6, Serpent i Twofish. (Więcej szczegółów można znaleźć na stronie https://csrc.nist.gov.) AES-256, część zestawu algorytmów kryptograficznych Suite B NSA, jest jednym z nielicznych komercyjnych algorytmów, które zostały uznane za wystarczająco silne, aby chronić informacje tajne.

Międzynarodowy algorytm szyfrowania danych

Międzynarodowy algorytm szyfrowania danych (IDEA) to szyfr blokowy, który działa na 64-bitowych blokach tekstu jawnego. Używa 128-bitowego klucza i jest używany w oprogramowaniu szyfrującym PGP (opisanym później w „Algorytmach asymetrycznych”). IDEA został opracowany przez Xuejię Lai i Jamesa Masseya, aby działał wydajniej na komputerach używanych w domu i w firmach. Jest bezpłatny do użytku niekomercyjnego, ale do użytku komercyjnego należy zakupić licencję. Ostateczny patent na szyfr blokowy IDEA wygasł w 2012 roku.

Blowfish

Blowfish to kolejny szyfr blokowy, który działa na 64-bitowych blokach tekstu jawnego. Jednak długość klucza jest zmienna, od 32 bitów do 448 bitów. Został on opracowany jako algorytm domeny publicznej przez Bruce'a Schneiera, czołowego kryptologa i autora książki Applied Cryptography: Protocols, Algorithms, and Source Code in C, Second Edition (Wiley, 1996, ISBN 0471117099), którą gorąco

polecamy tym, którzy chcą dowiedzieć się więcej o tym algorytmie i obejrzeć jego kod źródłowy w języku C.

RC4

RC4, najpowszechniej używany szyfr strumieniowy, jest używany w bezprzewodowym szyfrowaniu WEP. To ze względu na sposób implementacji RC4 w WEP znalezienie klucza za pomocą programów do łamania szyfrów jest tak łatwe. RC4 jest uważany za trudny do złamania, ale należy go unikać w większości zastosowań. Algorytm został stworzony przez Ronalda L. Rivesta w 1987 r. dla RSA Security (www.rsa.com).

RC5

RC5 to szyfr blokowy, który może działać na różnych rozmiarach bloków: 32, 64 lub 128 bitów. Rozmiar klucza może osiągnąć 2048 bitów. Algorytm został stworzony przez Ronalda L. Rivesta w 1994 r. dla RSA Security.

Algorytmy asymetryczne

Zamiast pojedynczego klucza używanego w algorytmach symetrycznych, algorytmy asymetryczne używają dwóch matematycznie powiązanych kluczy, więc dane zaszyfrowane jednym kluczem można odszyfrować tylko drugim kluczem. Inną nazwą algorytmów asymetrycznych jest kryptografia klucza publicznego; te terminy są często używane zamiennie. Klucz publiczny jest dostępny publicznie; w wielu przypadkach klucze publiczne można pobrać ze stron internetowych, aby mogła z nich korzystać publiczność. Klucz prywatny to klucz tajny znany tylko właścicielowi klucza i nigdy nie powinien być udostępniany. Nawet jeśli ludzie znają klucz publiczny używany do szyfrowania wiadomości, nie mogą ustalić klucza prywatnego właściciela klucza. W przypadku kryptosystemów asymetrycznych przechwycenie klucza publicznego podczas transmisji nie stanowi problemu. Ponadto algorytmy asymetryczne są bardziej skalowalne niż algorytmy symetryczne, ponieważ jeden klucz publiczny może być używany przez tysiące użytkowników; jednak algorytmy te wymagają większych zasobów procesora, więc są wolniejsze. Zanim przyjrzymy się niektórym powszechnie stosowanym algorytmom asymetrycznym, przyjrzyjmy się prostemu przykładowi kryptografii klucza publicznego. Istnieją różne sposoby szyfrowania wiadomości za pomocą algorytmów asymetrycznych, w zależności od tego, czy celem jest zapewnienie uwierzytelniania i niezaprzeczalności. Uwierzytelnianie weryfikuje, czy nadawca lub odbiorca (lub obaj) są tymi, za których się podają. Zaprzeczalność zapewnia, że nadawca i odbiorca nie mogą zaprzeczyć wysłaniu lub odebraniu wiadomości. Te funkcje nie są obsługiwane w algorytmach symetrycznych. Jeśli Użytkownik A zaszyfruje wiadomość swoim kluczem prywatnym i wyśle wiadomość do Użytkownika B, Użytkownik B może odszyfrować wiadomość za pomocą klucza publicznego Użytkownika A. Klucze prywatny i publiczny Użytkownika są powiązane matematycznie, co oznacza, że klucz publiczny może odszyfrować tylko wiadomość, która została zaszyfrowana za pomocą odpowiadającego mu klucza prywatnego. Jeśli poufność jest głównym zmartwieniem Użytkownika A, szyfruje on wiadomość za pomocą klucza publicznego odbiorcy. W ten sposób tylko odbiorca może odszyfrować wiadomość za pomocą swojego klucza prywatnego. Jeśli Użytkownik A chce zapewnić Użytkownika B, że to on jest osobą wysyłającą wiadomość (uwierzytelnianie), może zaszyfrować wiadomość swoim kluczem prywatnym. W końcu jest jedyną osobą posiadającą swój klucz prywatny.

RSA

RSA został opublikowany w 1978 roku przez trzech profesorów MIT: Ronalda L. Rivesta, Adiego Shamira i Leonarda M. Adlemana. Jest to pierwszy algorytm używany zarówno do szyfrowania, jak i

podpisywania cyfrowego i jest nadal szeroko stosowany, szczególnie w handlu elektronicznym. Autorzy udostępnili swoje odkrycia każdemu, kto wysłał im zaadresowaną kopertę. NSA podeszła do tego podejścia z rezerwą i zasugerowała profesorom zaprzestanie i zaniechanie. Jednak zapytana o legalność swojego żądania NSA nie odpowiedziała, a algorytm został opublikowany. Więcej informacji na temat RSA i jego relacji z NSA można znaleźć na stronie https://en.wikipedia.org/wiki/RSA_Security. Wiele przeglądarek internetowych korzystających z protokołu Transport Layer Security (TLS) używa algorytmu RSA, który opiera się na trudności rozkładania dużych liczb na czynniki. Aby wygenerować klucz, RSA używa funkcji jednokierunkowej — wzoru matematycznego, który jest łatwy do obliczenia w jednym kierunku, ale trudny lub prawie niemożliwy do obliczenia w przeciwnym kierunku. Na przykład, mnożenie dwóch dużych liczb pierwszych w celu określenia ich iloczynu jest łatwe, ale gdy masz podany tylko iloczyn, określenie, jakie liczby zostały użyte w obliczeniach, jest trudne. Prostą analogią jest robienie koktajlu. Łatwo jest zmiksować banana, truskawki i kostki lodu w blenderze, ale jeśli musisz odtworzyć banana, truskawki i kostki lodu do ich pierwotnego stanu po zmiksowaniu, zadanie to może okazać się niemożliwe.

Diffie-Hellman

Algorytm ten został opracowany w 1976 roku przez Whitfielda Diffiego i Martina Hellmana, twórców koncepcji klucza publicznego i prywatnego. Nie zapewnia szyfrowania, ale służy do ustanowienia jednego tajnego klucza współdzielonego przez dwie strony. Chociaż często uważa się to za wymianę kluczy, każda ze stron w rzeczywistości generuje wspólny klucz na podstawie matematycznej relacji klucz-zgoda. Jeśli klucz zostanie przechwycony podczas transmisji, sieć jest podatna na ataki, więc zarządzanie kluczami jest ważnym elementem zabezpieczania danych. Dzięki metodzie udostępniania tajnego klucza użytkownicy mogą zabezpieczyć swoją komunikację elektroniczną bez obawy przed przechwyceniem.

Kryptografia krzywych eliptycznych

Kryptografia krzywych eliptycznych (ECC), opracowana w 1985 r., jest używana do szyfrowania, a także podpisów cyfrowych i wymiany kluczy. ECC opiera się na złożonej algebrze i obliczeniach na krzywych. Wystarczy powiedzieć, że jest to wydajny algorytm wymagający niewielu zasobów (takich jak pamięć, miejsce na dysku i przepustowość), więc jest dobrym kandydatem do urządzeń bezprzewodowych i telefonów komórkowych. NSA uwzględniła ECC w swoich algorytmach kryptograficznych Suite B.

ElGamal

ElGamal to asymetryczny algorytm używany do generowania kluczy i podpisów cyfrowych oraz szyfrowania danych. Opracowany przez Tahera Elgamala w 1985 r. algorytm wykorzystuje dyskretne logarytmy, których rozwiązanie jest trudne. Rozwiązanie dyskretnego logarytmu może zająć wiele lat i wymagać operacji intensywnie wykorzystujących procesor.

Podpisy cyfrowe

Algorytmy asymetryczne mają przydatną cechę, która umożliwia kluczowi publicznemu odszyfrowanie wiadomości zaszyfrowanej kluczem prywatnym lub odwrotnie. Klucz publiczny może odszyfrować tylko wiadomość, która została zaszyfrowana odpowiednim kluczem prywatnym. Skrót obliczony z zawartości wiadomości jest szyfrowany kluczem prywatnym w celu zapewnienia uwierzytelnienia i niezaprzeczalności.

Standard podpisu cyfrowego

W 1991 r. NIST ustanowił Standard podpisu cyfrowego (DSS), aby zapewnić możliwość weryfikacji podpisów cyfrowych. Rząd federalny określił stosowanie RSA i algorytmu podpisu cyfrowego (DSA) dla wszystkich podpisów cyfrowych oraz stosowanie algorytmu haszującego w celu zapewnienia integralności wiadomości (weryfikacji, czy wiadomość nie została naruszona). NIST wymaga stosowania algorytmu Secure Hash Algorithm (SHA), omówionego później w części „Algorytmy haszujące". Zasadniczo podpis cyfrowy można utworzyć tylko przy użyciu klucza prywatnego użytkownika, a podpis użytkownika może zostać zweryfikowany przez dowolną osobę przy użyciu klucza publicznego tego użytkownika.

Pretty Good Privacy

Pretty Good Privacy (PgP) został opracowany przez Phila Zimmermana jako darmowy program do szyfrowania wiadomości e-mail, który pozwalał typowym użytkownikom na szyfrowanie wiadomości e-mail. Brzmi niegroźnie, ale Zimmerman został prawie aresztowany za swoją innowację. Departament Sprawiedliwości wszczął dochodzenie w sprawie tego, czy oferowanie programu PGP publicznie było przestępstwem. W połowie lat 90. każdy rodzaj „niezniszczalnego" szyfrowania był uważany za broń, a dzielenie się nim porównywano do sprzedaży broni wrogowi. PGP znacznie ewoluował od czasu jego powstania. Internetowy standard wiadomości PGP nazywa się teraz openPgP. OpenPGP używa certyfikatów podobnych do tych w PKI, ale ponieważ nie jest używany scentralizowany urząd certyfikacji (CA), weryfikacja CA nie jest tak wydajna jak w PKI. OpenPGP może używać algorytmów AES, IDEA, RSA, DSA i SHA do zarządzania kluczami oraz szyfrowania, uwierzytelniania i weryfikacji integralności wiadomości. Najpopularniejszą darmową wersją OpenPGP jest GNU Privacy Guard (GnuPG lub GPG; www.gnupg.org). Czasami OpenPGP może być używane do sprawdzania integralności dystrybucji i aktualizacji Linuksa typu open source, gdy programiści udostępniają metodę weryfikacji. Jeśli używasz dowolnej wersji Linuksa, GPG zapewnia, że zainstalowane pakiety oprogramowania i aktualizacje nie zostały naruszone przez intruza lub hakera. GPG jest przydatne do nauki, jak używany jest algorytm szyfrowania kluczem publicznym, a co najważniejsze, jest bezpłatne. Chociaż komercyjna wersja PGP (dostępna na stronie www.broadcom.com/products/cybersecurity/information-protection/encryption) jest zgodna ze standardem OpenPGP, nie jest już bezpłatna. Jednak, podobnie jak wiele komercyjnych produktów, PGP zapewnia wsparcie techniczne i ma więcej funkcji, dzięki czemu nadaje się do dużych sieci korporacyjnych. OpenPGP to ekonomiczny sposób wysyłania wiadomości e-mail z dodatkową warstwą zabezpieczeń. Nie jest to jednak jedyna technologia wysyłania bezpiecznych wiadomości e-mail. Poniższa sekcja krótko omawia inny powszechnie używany standard bezpiecznej poczty e-mail: S/MIME.

Secure Multipurpose Internet Mail Extension

Secure Multipurpose Internet Mail Extension (S/MIME) to kolejny standard szyfrowania kluczem publicznym do szyfrowania i cyfrowego podpisywania wiadomości e-mail. Może również szyfrować wiadomości e-mail zawierające załączniki i używać certyfikatów PKI do uwierzytelniania. (Zobacz RFC-2311, aby uzyskać informacje o S/MIME w wersji 2, RFC-2633, aby uzyskać informacje o S/MIME w wersji 3 i RFC-8551, aby uzyskać informacje o S/MIME w wersji 4.) Jednym z powodów, dla których S/MIME jest szeroko stosowany do szyfrowania wiadomości e-mail, jest to, że jest wbudowany w program Microsoft Outlook. Ponieważ program Outlook jest zawarty w pakiecie Microsoft Office, organizacje, które już korzystają z pakietu Microsoft Office, nie muszą instalować dodatkowego oprogramowania do szyfrowania wiadomości e-mail.

UWAGA : Privacy Enhanced Mail (PEM) i MIME Object Security Services (MOSS) to starsze standardy szyfrowania wiadomości e-mail, które zostały porzucone z powodu braku zgodności ze standardami OpenPGP i S/MIME.

Szyfrowanie poufnych danych

Jako tester bezpieczeństwa często komunikujesz się z klientami za pośrednictwem poczty e-mail. Jednak wysyłanie wyników testów, które ujawniają luki w zabezpieczeniach sieci klienta za pośrednictwem niezaszyfrowanej poczty e-mail, która podlega przechwyceniu, może skutkować poważną luką w zabezpieczeniach. Nie przyczyniaj się do problemów z bezpieczeństwem sieci klienta, wprowadzając je samodzielnie. Przestrzeganie dobrych zasad bezpieczeństwa zwiększa zaufanie klienta do Twojej pracy. Dlatego też, ustal zasadę wymiany wyników testów lub innych poufnych dokumentów w formie zaszyfrowanej. Jeśli Twój klient nie korzysta z zaszyfrowanej poczty e-mail, powinieneś to uczynić jednym z pierwszych zaleceń. Organizacje mogą również potrzebować szyfrowania danych w spoczynku, co oznacza wszelkie dane, które nie są przesyłane przez sieć lub nie są używane przez system operacyjny; termin ten zwykle odnosi się do danych przechowywanych na stacjach roboczych, serwerach, smartfonach, dyskach wymiennych, nośnikach kopii zapasowych i laptopach. Wiele organizacji jest zobowiązanych przez prawo do szyfrowania poufnych i finansowych informacji oraz zgłaszania władzom, jeśli te informacje są niezaszyfrowane i zostały utracone lub skradzione. Utrata tych informacji zwykle okazuje się kosztowna, nie tylko pod względem kosztów ich zastąpienia, ale także złej reklamy. Wiele komercyjnych programów może szyfrować dane w stanie spoczynku, a bezpłatne programy, takie jak VeraCrypt (https://veracrypt.fr/en/Home.html), są również dostępne. VeraCrypt wykorzystuje silne algorytmy szyfrowania, takie jak AES-256, który jest autoryzowany do ochrony tajnych informacji rządu USA.

BAJTY BEZPIECZEŃSTWA

Szyfrowanie zawsze było czymś, co ktoś chciał złamać. Dla niektórych jest to wyzwanie rozwiązania zagadki. Dla innych złamanie szyfrowania umożliwia im podsłuchiwanie dyskusji dwóch przeciwników za pośrednictwem wiadomości e-mail, wiadomości tekstowych lub innych mediów. Cyberprzestępcy biorą teraz firmy jako zakładników, szyfrując wszystkie dane firmy i żądając znacznych płatności za ich odszyfrowanie. Niedawno FBI znalazło iPhone'a 5c należącego do zmarłej pary podejrzewanej o terroryzm. FBI uważało, że telefon może zawierać dowody lub możliwe wskazówki dotyczące innych ataków, ale nie mogło złamać szyfrowania telefonu. Apple, Inc. odmówiło pomocy FBI, powołując się na wolności obywatelskie i wolność słowa. W rezultacie FBI zatrudniło firmę hakerów, aby stworzyli narzędzie do złamania szyfrowania iPhone'a. Nie trzeba dodawać, że FBI dostało się do iPhone'a. Ile FBI zapłaciło hakerom? Trochę ponad 1 000 000 dolarów!

Algorytmy haszujące

Obecnie w użyciu jest kilka algorytmów haszujących; Tabela 12-2 podsumowuje niektóre z najpopularniejszych.

Algorytm: Opis

MD2: Opracowany przez Ronalda L. Rivesta w 1989 roku, ten algorytm został zoptymalizowany dla maszyn 8-bitowych.

MD4: Opracowany przez Rivesta w 1990 roku. Używając komputera PC, kolizje w tej wersji można teraz znaleźć w mniej niż 1 minutę. Microsoft Windows nadal używa RC4 do przechowywania skrótów haseł.

MD5: Opracowany przez Rivesta w 1991 roku. W 1994 roku oszacowano, że stworzenie komputera, który mógłby znajdować kolizje za pomocą ataków siłowych, kosztowałoby 10 milionów dolarów.

Jednak kolizję dla skrótu MD5 można teraz znaleźć za pomocą zaledwie kilku maszyn w ciągu kilku godzin.

MD6: Opracowany przez Rivesta i jego zespół w MIT w 2008 roku. Wykorzystuje strukturę podobną do drzewa Merkle'a, aby umożliwić ogromne równoległe obliczenia skrótów dla bardzo długich danych wejściowych. Zgłaszano, że prędkości przekraczające 1 GB/s są możliwe dla długich wiadomości na 16-rdzeniowej architekturze procesora. MD6 nie jest powszechnie używany.

SHA-1: SHA-160, powszechnie znany jako SHA-1, jest uważany za zepsuty od 2005 r., ale zbliża się data, w której ataki kolizyjne staną się dostępne. Używa 160-bitowego skrótu i w momencie pisania tego tekstu był stosowany w wielu aplikacjach w sektorze rządowym i prywatnym.

SHA-2: Zbiorcze określenie dłuższych wersji skrótów algorytmów SHA: SHA-224, SHA-256, SHA-384 i SHA-512. Wersje SHA-2 wykorzystują zasadniczo ten sam algorytm co SHA-160, ale dłuższe skróty utrudniają znalezienie kolizji.

SHA-3: Keccak został wybrany jako standard SHA-3 przez NIST w 2015 r. Keccak, lub SHA-3, może przyjąć dane wejściowe dowolnej wielkości i utworzyć dane wyjściowe dowolnej wielkości. SHA-3 nie ma na celu natychmiastowego zastąpienia SHA-2 i znacznie różni się pod względem konstrukcji od swoich poprzedników SHA. Możesz przeczytać więcej o algorytmie Keccak na jego stronie internetowej (keccak.noekeon.org).

Algorytm haszujący to funkcja, która przyjmuje ciąg znaków o zmiennej długości lub wiadomość i generuje wartość skrótu o stałej długości, zwaną również skrótem wiadomości, używaną do weryfikacji integralności danych lub wiadomości. W pewnym sensie jest to jak odcisk palca wiadomości. Na przykład, jeśli wiadomość „Jak się masz?” zostanie później zmieniona na „Kim jesteś?”, wartość skrótu również się zmienia, tak aby odbiorca wiedział, że oryginalna wiadomość została zmieniona podczas transmisji. Dwie różne wiadomości generujące tę samą wartość skrótu powodują kolizję. Dlatego dobry algorytm haszujący to taki, który jest odporny na kolizje. Wiele starszych systemów opiera się na Message Digest 5 (MD5) i Secure Hash Algorithm 1 (SHA-1); jednak nowoczesne systemy szybko aktualizują się do SHA-2 i SHA-3. Pod koniec 2016 roku główne przeglądarki internetowe nie obsługiwały już SHA-1. Przy rozsądnej mocy obliczeniowej kolizje skrótów MD5 można znaleźć w ciągu kilku dni. Ataki na SHA-1, 160-bitową wersję SHA, są obecnie uważane za bardziej praktyczne, a badacze publikują metody ataków. Na przykład badacze z Uniwersytetu Shandong we wschodnich Chinach wykazali, że kluczowa funkcja skrótu w najnowocześniejszym szyfrowaniu może być mniej odporna na ataki, niż sądzono. Od 2015 roku eksperci zalecają nieużywanie SHA-1. NIST poinstruował agencje federalne, aby usunęły SHA-1 z przyszłych aplikacji i zastąpiły go SHA-2 lub SHA-3. Jak widać, specjaliści ds. bezpieczeństwa muszą być czujni i na bieżąco śledzić zmiany. Banki, witryny e-commerce, firmy obsługujące karty kredytowe i wojsko używają SHA od wielu lat. Z tego powodu NIST ogłosił konkurs, podobny do konkursu AES omawianego wcześniej, aby zastąpić SHA, a nie tylko zwiększyć długość jego skrótu.

ZROZUMIENIE INFRASTRUKTURY KLUCZA PUBLICZNEGO

Dyskusja na temat szyfrowania kluczem publicznym nie może odbyć się bez wspomnienia o infrastrukturze klucza publicznego (PKI). PKI nie jest algorytmem; jest to struktura składająca się z programów, protokołów i zasad bezpieczeństwa służących do szyfrowania danych i wykorzystuje kryptografię klucza publicznego do ochrony danych przesyłanych przez Internet. Temat PKI może zająć całą książkę, więc ta sekcja daje jedynie przegląd jej głównych komponentów i sposobu, w jaki PKI jest wykorzystywane do tworzenia certyfikatów.

Składniki PKI

Innym sposobem uwierzytelniania za pośrednictwem kanału komunikacyjnego są certyfikaty. Certyfikat to cyfrowy dokument potwierdzający, że dwie strony wymieniające dane przez Internet są rzeczywiście tymi, za których się podają. Każdy certyfikat zawiera unikalny numer seryjny i musi być zgodny ze standardem X.509, który opisuje tworzenie certyfikatu. Na przykład SSL i S/MIME to standardy internetowe wykorzystujące certyfikaty X.509. Klucze publiczne są wydawane przez urząd certyfikacji (CA). Urząd certyfikacji ręczy za firmę, do której wysyłasz numer swojej karty kredytowej, zamawiając motocykl Harley-Davidson online. Prawdopodobnie chcesz wiedzieć, że firma, od której zamawiasz motocykl, jest wiarygodna, a nie ktoś, kto założył fałszywą stronę internetową, aby zbierać numery kart kredytowych od niczego niepodejrzewających ofiar. Pomyśl o CA jako o agencji paszportowej. Kiedy obywatele USA pokazują swoje paszporty, aby wjechać do obcego kraju, agenci celni przeglądający paszport niekoniecznie ufają posiadaczom paszportów. Ufają jednak agencji paszportowej, która wydała paszporty, więc obywatele USA mogą wjechać do kraju. Certyfikat wystawiony przez zaufany CA wiąże klucz publiczny z tożsamością organizacji lub osoby, która go kupiła. W ten sposób, jeśli zaszyfrujesz wiadomość e-mail kluczem publicznym swojej przyjaciółki Ye-Jun, wiesz, że tylko ona może odszyfrować wiadomość za pomocą swojego klucza prywatnego, który jest matematycznie powiązany z jej kluczem publicznym. Wiesz również, że użyty przez Ciebie klucz publiczny jest rzeczywiście kluczem publicznym Ye-Jun, ponieważ ufasz CA, które go wydało.

Wygasanie, unieważnianie i zawieszanie certyfikatów

Certyfikatowi wydanemu przez CA przypisuje się okres ważności, a po upływie tej daty certyfikat wygasa. Jeśli klucze są nadal ważne i nie zostały naruszone, certyfikat można odnowić z nową datą wygaśnięcia. Czasami certyfikat może wymagać zawieszenia lub unieważnienia przed upływem daty wygaśnięcia, jak w następujących okolicznościach:

• Użytkownik opuszcza firmę.

• Awaria sprzętu powoduje utratę klucza.

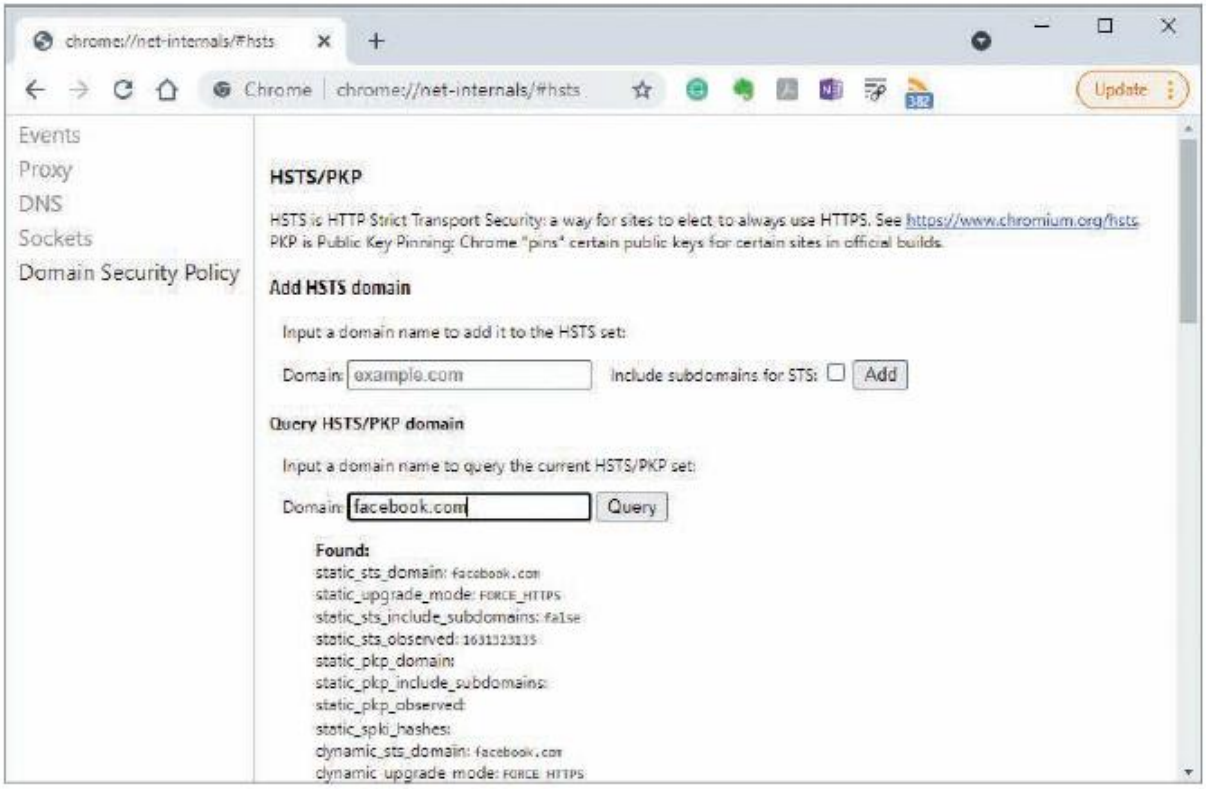
• Klucz prywatny został naruszony.

• Firma, której wydano certyfikat, już nie istnieje.

• Firma podała fałszywe informacje podczas żądania certyfikatu.

Urząd certyfikacji sporządza listę odwołanych certyfikatów (CRL) zawierającą wszystkie odwołane i zawieszone certyfikaty. Certyfikat może zostać zawieszony, gdy strony nie dotrzymają umów zawartych podczas wydawania certyfikatu. Zamiast unieważniać certyfikat, można go zawiesić, aby łatwiej go było przywrócić, jeśli strony dojdą do porozumienia później. Jeśli chcesz sprawdzić, czy certyfikat jest nadal ważny, możesz pobrać listę CRL z adresu URL określonego w certyfikacie. W przypadku urzędów certyfikacji, które mogą mieć wiele odwołanych certyfikatów, możesz użyć protokołu Online Certificate Status Protocol (OCSP) zamiast pobierania listy CRL. Za pomocą protokołu OCSP możesz (lub urządzenie) sprawdzić status certyfikatu bez konieczności pobierania i sprawdzania całej listy CRL. Ścisłe zabezpieczenia transportu HTTP Ścisłe zabezpieczenia transportu HTTP (Hsts) zostały stworzone w 2012 r. jako mechanizm dla serwerów internetowych, aby poinformować klientów, że wymagają bezpiecznej komunikacji. HSTS robi dwie rzeczy, aby promować bezpieczną komunikację klient-serwer. Po pierwsze, wymusza, aby cały ruch między przeglądarką internetową a serwerem internetowym był wysyłany przez bezpieczny kanał. Jest to realizowane za pomocą pola wysyłanego w nagłówku każdej odpowiedzi serwera internetowego, które wymaga od przeglądarki

wymuszenia protokołu HTTPS dla całego ruchu wysyłanego do tego serwera internetowego. Rysunek przedstawia informacje HSTS przechowywane w przeglądarce Chrome dla facebook.com.



Po drugie, jeśli przeglądarka nie może zweryfikować certyfikatu serwera internetowego, przeglądarka nie zezwala na dostęp do witryny. Chociaż HSTS nie jest niezawodny i atakujący mogliby usunąć pole HSTS z początkowej odpowiedzi serwera internetowego, jest to ważna część podejścia do obrony w głąb w zakresie bezpieczeństwa sieci.

Tworzenie kopii zapasowych kluczy

Tworzenie kopii zapasowych kluczy jest równie ważne, jak tworzenie kopii zapasowych danych. Jeśli klucze zostaną zniszczone i nie zostaną poprawnie utworzone, zaszyfrowane informacje o znaczeniu krytycznym dla firmy mogą być nie do odzyskania. Firmy zazwyczaj tworzą kopie zapasowe swoich kluczy i przechowują je w trybie offline w sejfie lub skarbcu. Rejestr łańcucha dostaw kluczy jest często wymagany w przypadku firm przetwarzających poufne informacje lub transakcje finansowe.

Główny urząd certyfikacji firmy Microsoft

Firma Microsoft uwzględnia w swoich systemach operacyjnych serwerów funkcje umożliwiające skonfigurowanie serwera jako urzędu certyfikacji zamiast korzystania z urzędu certyfikacji innej firmy. Na przykład za pomocą Kreatora dodawania ról i funkcji w systemie Windows Server 2019 jako administrator możesz wybrać Usługi certyfikatów usługi Active Directory. Po kliknięciu przycisku Dalej możesz wybrać dowolne funkcje, które chcesz zainstalować, w tym narzędzia do zarządzania urzędem certyfikacji. Po zainstalowaniu roli serwera należy skonfigurować usługi CA, wybierając usługi ról. Rola urzędu certyfikacji służy do wydawania i zarządzania certyfikatami cyfrowymi. Następnie należy

określić typ urzędu certyfikacji. Można wybrać urząd certyfikacji przedsiębiorstwa lub samodzielny jako typ ogólny, a następnie wybrać urząd główny lub podrzędny jako typ szczegółowy. Urząd certyfikacji głównej wydaje własny certyfikat, a urząd podrzędny otrzymuje certyfikat od innego urzędu certyfikacji znajdującego się wyżej w strukturze PKI. Jeśli wybierzesz typ urzędu certyfikacji głównej, musisz wygenerować nowy certyfikat. Można wybrać trzy ustawienia generowania certyfikatów: dostawcę usług kryptograficznych (CSP), algorytm skrótu i długość klucza. Spośród dostawców CSP dostępnych w systemie Windows Server 2019 wiele z nich dopuszcza słabe algorytmy skrótu, takie jak SHA-1, MD2, MD4 i MD5. RSA#Microsoft CSP zezwala jednak na długość skrótu do SHA-512. Niezależnie od tego, który CA wybierze firma, powinieneś być świadomy rodzaju używanego algorytmu, aby wiedzieć, czy firma jest podatna na atak, jeśli informacje o algorytmie zostaną naruszone. Certyfikaty zagrożone naruszeniem mogą stworzyć poważną lukę w zabezpieczeniach firmy i nie należy ich pomijać podczas przeprowadzania testu bezpieczeństwa.

Tworzenie certyfikatu Rogue Server przez złamanie algorytmu haszującego

Czas trwania: 30 minut

Cel: Zbadanie, co atakujący mogą zrobić z wynikami kolizji MD5.

Opis: Kolizje algorytmów haszujących były historycznie teoretycznym zagrożeniem, chociaż ostatnie wzrosty mocy obliczeniowej uczyniły kolizje rzeczywistością. W 2017 r. naukowcy z Francji i Singapuru zademonstrowali udany atak kolizyjny SHA-1. Kolizje w MD5 były demonstrowane od ponad dekady. Do niedawna nawet niektóre znane urzędy certyfikacji używały MD5 do generowania certyfikatów SSL serwera internetowego. W tej aktywności badasz, co jest możliwe, gdy inteligentni badacze decydują się zwrócić uwagę na poważny problem bezpieczeństwa w Internecie.

1. Uruchom przeglądarkę internetową w systemie Windows i przejdź do witryny www.google.com.

2. Wpisz tworzenie certyfikatu Rogue CA i naciśnij Enter. Kliknij pierwszy link w wynikach wyszukiwania, który powinien przekierować Cię do strony badań Rogue CA w witrynie Phreedom.org. (Jeśli nie, przejdź do ww.phreedom.org i wyszukaj hasło rogue CA.)

3. Przeczytaj akapity podsumowujące ustalenia badaczy, a następnie kliknij łącze Slides from the 25c3 presentation, aby pobrać prezentację PowerPoint, której użyjesz do odpowiedzi na poniższe pytania.

Uwaga: w zależności od używanej przeglądarki może być konieczne kliknięcie ostrzeżenia o zabezpieczeniach.

● Badacze zebrali 30 000 certyfikatów witryn w 2008 roku. Ile z nich zostało podpisanych za pomocą MD5?

● Jakiego rodzaju sprzętu użyto do wygenerowania kolizji wybranego prefiksu? Ile pieniędzy badacze wydali na certyfikaty?

● Jaki był wpływ wygenerowania fałszywego certyfikatu CA? Co ten certyfikat umożliwiłby osobie o złośliwych zamiarach?

● Jakiego algorytmu haszującego musiały używać CA po wykazaniu, że ich metoda podpisywania nie jest bezpieczna?

● Jaki jest, według badaczy, jedyny sposób na wprowadzenie zmian i zabezpieczenie Internetu?

4. Zamknij wszystkie otwarte okna.

ZROZUMIENIE ATAKÓW KRYPTOGRAFICZNYCH

Używanie narzędzia do podsłuchu (takich jak tcpdump i Wireshark) jest szkodliwe dla pasywnych, zakłócających zbieranie tylko dane wysyłane do i z kryptosystemu. Aktywne użycie klucza używanego do szyfrowania tekstu jawnego poprzez aktywne wysyłanie danych do kryptosystemu. Pamiętaj, że jeśli sprawca lub społeczeństwo zna algorytm, zwykle dzieje się tak, firmy opracowujące algorytmy szyfrowania sobie sprawę, że może odkryć luki w zabezpieczeniach, których programiści nie zauważyli. Inżynierowie oprogramowania, którzy opracowują produkty z kodem open source, kierują się tą filozofią. Ponieważ udostępniają swój kod źródłowy, użytkownicy mogą uzyskać wnioski i rozszerzyć lub uzyskać kod źródłowy. Pozornie udostępnienie kodu źródłowego może być lepszym produktem. Agencje takie jak NSA i CIA nie udostępniają jednak informacji o wszystkich połączeniach przez siebie algorytmach szyfrowania. Następujące sekcje o określonych typach użytkownika.

Atak urodzinowy

Prawdopodobnie słyszałeś powiedzenie, że jeśli w pokoju znajduje się 23 osoby, prawdopodobieństwo, że dwie z nich będą miały te same urodziny, wynosi około 50 procent. Ataki urodzinowe służą do znajdowania tej samej wartości skrótu dla dwóch różnych danych wejściowych i ujawniania wszelkich matematycznych słabości algorytmu haszującego. Na przykład, jeśli atakujący ma jedną wartość skrótu i chce znaleźć inną wiadomość, która tworzy tę samą wartość skrótu, może to zrobić w ciągu kilku godzin, jeśli algorytm haszujący jest słaby. SHA-1, omówiony wcześniej, używa 160-bitowego skrótu. Teoretycznie znalezienie kolizji dla innej wiadomości (tej samej daty urodzin dla innej osoby, w tej analogii) wymagałoby 260 obliczeń, co może być możliwe w niedalekiej przyszłości.

Atak matematyczny

W ataku matematycznym właściwości algorytmu są atakowane za pomocą obliczeń matematycznych. Atakujący wykonują ten typ ataku na różne sposoby, w zależności od informacji, do których mają dostęp. Istnieje pięć głównych kategorii tego ataku:

• Atak wyłącznie tekstem zaszyfrowanym — Atakujący mają tekst zaszyfrowany kilku wiadomości za pomocą tego samego algorytmu szyfrowania, ale nie mają dostępu do tekstu jawnego, więc muszą spróbować ustalić klucz użyty do zaszyfrowania danych. Uzyskanie kopii tekstu zaszyfrowanego jest zwykle łatwe za pomocą sniffera, takiego jak tcpdump lub Wireshark, ale ten typ ataku jest zdecydowanie najtrudniejszy, ponieważ niewiele lub wcale nie wiadomo o użytym algorytmie szyfrowania.

• Atak znanym tekstem jawnym — Atakujący mają wiadomości zarówno w formie zaszyfrowanej, jak i odszyfrowanej. Ten atak jest łatwiejszy niż atak wyłącznie tekstem zaszyfrowanym, ponieważ można zbadać wzorce w tekście jawnym. Na przykład, jeśli komunikacja banku z klientami zawsze zaczyna się od określonego powitania i kończy znanym „Dziękujemy za współpracę”, atakujący mogą użyć technik inżynierii wstecznej, aby ustalić klucz użyty do zaszyfrowania danych.

• Atak z wybranym tekstem jawnym — atakujący mają dostęp do tekstu jawnego i zaszyfrowanego i mogą wybrać, które wiadomości zaszyfrować. Ponieważ cała wiadomość jawna i zaszyfrowana są dostępne, ustalenie klucza jest łatwiejsze. Atakujący mogą uzyskać te informacje, wysyłając wiadomość e-mail do kogoś, oświadczając, że jej zawartość nie ma być ujawniana nikomu poza, powiedzmy, Bobem Smithem. Najprawdopodobniej podrobią wiadomość e-mail, aby odbiorca uwierzył, że wiadomość pochodzi od kogoś znanego i zaufanego. Kiedy odbiorca przekaże wiadomość jako zaszyfrowany tekst, atakujący mogą następnie podsłuchiwać zawartość, aby uzyskać zarówno napisany przez siebie tekst jawny, jak i zaszyfrowany dokument wysłany przez użytkownika.

• Atak z wybranym tekstem jawnym — atakujący mają dostęp do zaszyfrowanego tekstu, który ma zostać odszyfrowany, oraz do powstałego tekstu jawnego. Potrzebują również dostępu do kryptosystemu, aby przeprowadzić ten typ ataku.

• Atak kanału bocznego — ten atak, który jest zupełnie inny od innych kategorii, polega na tym, że atakujący analizuje sprzęt używany do operacji kryptograficznych. Atakujący zbierają dane, takie jak temperatury robocze, czasy obliczeń, emisje elektromagnetyczne, hałas, wibracje, a nawet odbicia od oczu użytkownika kryptosystemu, aby zebrać informacje, których mogą użyć do uruchomienia exploita. Zazwyczaj atakujący potrzebują bliskiej odległości od kryptosystemu, aby zebrać te informacje.

Niezależnie od rodzaju ataku, atakujący opiera się na uzyskanych informacjach, a następnie przeprowadza inny rodzaj ataku. Cierpliwość i ciekawość są zwykle częścią osobowości kryptologów, niezależnie od tego, czy pracują w dobrych, czy złych celach.

Atak siłowy

Pomimo nazwy, ten typ ataku nie wymaga młota ani umiejętności sztuk walki; wymaga jedynie czasu i cierpliwości. Atak siłowy próbuje wszystkich możliwych kluczy w przestrzeni kluczy. Jednym z przykładów jest użycie programu do łamania haseł, aby wypróbować każdą możliwą kombinację znaków w celu złamania skrótu hasła. Ataki siłowe można przeprowadzić na dowolnym rodzaju skrótu wiadomości, takim jak żądanie certyfikatu. Jeśli chcesz dowiedzieć się, ile czasu może zająć atak siłowy na złamanie hasła, możesz pobrać kalkulator czasu ataku siłowego w Mandylion Labs (www.mandylionlabs.com/documents/BFTCalc.xls).

Atak typu Man-in-the-Middle

W ataku typu Man-in-the-Middle atakujący ustawiają się między komputerem ofiary a innym komputerem hosta. Mogą następnie przechwytywać wiadomości wysyłane przez ofiarę do hosta i udawać komputer hosta. Ten typ ataku przebiega następująco:

1. Gloria wysyła swój klucz publiczny do Bruce'a, a ty, atakujący, przechwytujesz klucz i wysyłasz Bruce'owi swój klucz publiczny. Bruce myśli, że właśnie otrzymał klucz publiczny Glorii, ale otrzymał twój.

2. Bruce wysyła Glorii swój klucz publiczny. Ty również przechwytujesz ten klucz i wysyłasz Glorii swój klucz publiczny.

3. Gloria wysyła wiadomość do Bruce'a zaszyfrowaną tym, co uważa za klucz publiczny Bruce'a, ale ponieważ używa twojego, możesz odszyfrować wiadomość swoim kluczem prywatnym.

4. Następnie możesz ponownie zaszyfrować wiadomość kluczem publicznym Bruce'a i wysłać ją Bruce'owi.

5. Bruce odpowiada Glorii, szyfrując swoją wiadomość tym, co uważa za klucz publiczny Glorii. Ty przechwytujesz wiadomość, odszyfrowujesz ją swoim kluczem prywatnym, szyfrujesz ją prawdziwym kluczem publicznym Glorii, a następnie wysyłasz ją do Glorii. Być może będziesz musiał przeczytać te kroki kilka razy, aby zrozumieć, jak działa ten typ ataku. Korzystanie z kartek indeksowych z nazwiskami uczestników może pomóc Ci uzyskać jaśniejszy obraz tego, co się dzieje; ta technika jest stosowana w Aktywności poniżej.

Przeprowadzenie ataku typu Man-in-the-middle

Czas trwania: 20 minut

Cel: Zrozumienie, jak działa atak typu Man-in-the-middle.

Opis: Używając kart indeksowych i dzieląc się na zespoły trzyosobowe, przeprowadzasz ręczny atak typu Man-in-the-middle.

1. Dwóch uczniów powinno utworzyć dwie karty indeksowe. Oznacz jedną kartę Imię Klucz publiczny, a drugą Imię Klucz prywatny. (Zamień swoje imię na Imię.)

2. Atakujący przeprowadzający atak typu Man-in-the-middle powinien nazwać swoje karty atakującym Klucz publiczny i atakującym Klucz prywatny.

3. Gdy pierwszy uczeń podaje swoje Imię Klucz publiczny drugiemu uczniowi, atakujący powinien przechwycić transfer i podmienić swój Atakujący Klucz publiczny.

4. Uczeń otrzymujący tę kartę atakującego jest pod wrażeniem, że otrzymał prawdziwy klucz publiczny, a następnie zaszyfruje wiadomość tym kluczem publicznym i odeśle ją do nadawcy. 5. Atakujący powinien przechwycić tę kartę i użyć swojej karty klucza prywatnego, aby symulować odszyfrowanie wiadomości

Atak obniżający SSL/TLS

W przypadku ataku obniżającego SSL/TLS atakujący, który przechwytuje początkową komunikację między serwerem WWW a przeglądarką internetową, może zmusić podatny serwer do niebezpiecznej renegocjacji używanego szyfrowania do słabszego szyfru. Dzieje się tak, ponieważ serwer WWW i przeglądarka internetowa muszą wynegocjować, który szyfr zostanie użyty do komunikacji, zanim zaczną. Jeśli klient poinformuje serwer, że może komunikować się tylko za pomocą słabych protokołów, a serwer zgodzi się na użycie tego słabego protokołu, późniejsza komunikacja może być zagrożona. Na przykład w 2014 r. badacze bezpieczeństwa opublikowali niebezpieczny atak obniżający, w którym atakujący mógł zmusić podatne serwery do komunikowania się za pomocą niezwykle słabego szyfru o nazwie „export-grade". Rozwiązaniem tego problemu było upewnienie się, że wszystkie szyfry dozwolone przez serwer są bezpieczne.

Atak słownikowy

W ataku słownikowym po uzyskaniu dostępu do pliku haseł atakujący mogą uruchomić program do łamania haseł, który używa słownika znanych słów lub haseł jako pliku wejściowego. Większość tych plików wejściowych jest dostępna w Internecie i można je pobrać bezpłatnie. Pamiętaj, że nieautoryzowane łamanie haseł jest nielegalne w większości części świata, w tym w Stanach Zjednoczonych.

Atak typu replay

W ataku typu replay atakujący przechwytuje dane i próbuje ponownie przesłać przechwycone dane, aby urządzenie, którym może być komputer lub router, myślało, że istnieje prawidłowe połączenie. Jeśli przechwycone dane są informacjami logowania, atakujący może uzyskać dostęp do systemu i zostać uwierzytelniony. Wiele systemów ma środki zaradcze, aby zapobiec wystąpieniu tych ataków, takie jak pakiety wykorzystujące numery sekwencyjne, które wykrywają, kiedy pakiet jest w niewłaściwej kolejności lub nie znajduje się w prawidłowej kolejności.

ZROZUMIENIE ŁAMANIA HASŁA

Jako specjalista ds. bezpieczeństwa możesz natknąć się na zaszyfrowane lub chronione hasłem pliki. Hasła często można łatwo odgadnąć, zwłaszcza gdy są to imiona zwierząt domowych, krewnych lub małżonków, rocznice i daty urodzenia. Badanie przeprowadzone przez NSA około 30 lat temu

wykazało, że 70 procent wszystkich haseł jest zapisywanych w odległości do 4 stóp od komputera użytkownika. Ponadto, parafrazując inżyniera społecznego, gdy zapytano go o łamanie haseł: „Po co tracić czas na próby odszyfrowania hasła, skoro można po prostu o nie poprosić?" W większości krajów, w tym w Stanach Zjednoczonych, łamanie haseł innych osób jest nielegalne. (Możesz złamać własne hasło, jeśli je zapomnisz). Nawet próba odkrycia metody szyfrowania może być nielegalna w wielu krajach. Jeśli hasło zawiera powszechne słowa znalezione w słowniku, większość programów do łamania haseł może użyć pliku słownika, aby przyspieszyć proces. Siła jest powszechną metodą łamania hasła. Jednym ze sposobów przyspieszenia próby łamania metodą brute-force jest użycie tablicy tęczowej. Program do łamania haseł może użyć tej tablicy wyszukiwania wartości skrótu hasła zamiast próbować losowych obliczeń na przestrzeni kluczy skrótu hasła. Jednak aby tablice tęczowe były skuteczne, muszą przechowywać wiele wartości skrótu. Ponadto tablice tęczowe dla haseł zawierających więcej niż 10 znaków mogą szybko zapełnić setki terabajtów pamięci ze względu na wykładniczy charakter dostępnych permutacji w stosunku do długości hasła. Sól, w terminologii kryptograficznej, to użycie losowych danych obok tekstu jawnego jako danych wejściowych do funkcji haszującej, tak aby dane wyjściowe były unikalne. Sole sprawiają, że wstępnie obliczone tablice tęczowe są bezużyteczne, ponieważ wynikowe skróty są całkowicie różne ze względu na losowe dane zawarte w procedurze haszującej. Sole należy zawsze stosować podczas przechowywania haseł. W niektórych dużych naruszeniach bezpieczeństwa stron internetowych w niedawnej historii, niezasolone hasła ujawniły hasła użytkowników i doprowadziły do naruszenia konta i dodatkowego naruszenia dla użytkowników, którzy mieli to samo hasło w wielu witrynach. Jednostka przetwarzania grafiki (GPU) znacznie przewyższa możliwości procesora CPU w zakresie przetwarzania obliczeń matematycznych. Typowy procesor GPU ma setki rdzeni, podczas gdy typowy procesor CPU ma cztery lub osiem rdzeni. Każdy z tych rdzeni może przetwarzać jedno obliczenie matematyczne na cykl. W momencie pisania tego tekstu równoległe przetwarzanie setek rdzeni zapewnia procesorom GPU od trzech do pięciu razy większą przepustowość w przypadku łamania haseł w porównaniu z procesorami CPU. Programy takie jak Hashcat można ustawić tak, aby wykorzystywały procesor GPU, a nie procesor CPU.

BAJTY BEZPIECZEŃSTWA

W 2021 r. 3,28 miliarda haseł powiązanych z 2,18 miliarda unikalnych adresów e-mail zostało ujawnionych w jednym z największych wycieków danych dotyczących naruszonych nazw użytkowników i haseł. Wycieki obejmowały 1 502 909 haseł powiązanych z adresami e-mail z domen rządowych na całym świecie. Hasła miały zostać uzyskane przy użyciu technik takich jak łamanie haseł. Hasła zostały zdobyte na wiele sposobów, w tym poprzez pliki skradzione z serwerów, ataki phishingowe e-mailem i przechwyconą niezabezpieczoną komunikację w postaci zwykłego tekstu.

Aby przeprowadzić łamanie haseł, najpierw musisz uzyskać skrót hasła z systemu, który przechowuje nazwy użytkowników i hasła, który różni się w zależności od testowanego systemu operacyjnego. W systemach *nix skrót hasła jest przechowywany w pliku /etc/shadow. Program Fgdump wyodrębnia pliki z pliku Security Accounts Manager (SAM), w którym przechowywane są skróty haseł systemu Windows. Ataki łamania haseł można przeprowadzić za pomocą następujących programów:

• Hashcat — szybkie narzędzie do łamania haseł wbudowane w system Kali Linux, które może wykorzystywać procesory graficzne w celu zwiększenia wydajności w porównaniu z narzędziami do łamania opartymi na procesorach

• John the Ripper — lekkie narzędzie do łamania haseł dostępne do łamania plików haseł; może używać słownika lub prostych metod siłowych

• 0phcrack — pierwszy program do łamania haseł, który używa tablic tęczowych

• EXPECT — język skryptowy dla systemów Windows i Linux, który wykonuje powtarzalne zadania, takie jak łamanie haseł

• L0phtcrack — oryginalny program do łamania haseł, obecnie używany przez wiele agencji rządowych do testowania siły haseł; potrafi używać tablic tęczowych

• Pwdump8 — najnowsza wersja programu pwdump do wyodrębniania wartości haszujących haseł kont użytkowników na komputerze z systemem Windows

Tester bezpieczeństwa może użyć następujących kroków, aby zebrać hasła na komputerze z systemem Windows.

UWAGA: Wykonywanie tych kroków na komputerze innym niż Twój jest nielegalne w większości części świata. W rzeczywistości korzystanie z oprogramowania do łamania haseł na komputerze innym niż Twój może być niebezpieczne

W poniższym przykładzie fgdump jest używany do zrzucania skrótów z komputera z systemem Windows 10, a John the Ripper do łamania skrótów na komputerze z systemem Kali Linux:

1. Tester bezpieczeństwa uruchamia program fgdump, aby uzyskać wartości skrótów kont użytkowników na komputerze z systemem Windows. Uruchomienie programu Fgdump bez opcji powoduje zrzucanie kont użytkowników komputera do pliku 127.0.0.1.pwdump.

2. Używając programu John the Ripper z plikiem wejściowym 127.0.0.1.pwdump, tester bezpieczeństwa może użyć polecenia john -f5NT 127.0.0.1.pwdump, aby przeprowadzić atak siłowy na wartości skrótów odkryte za pomocą programu fgdump. Przełącznik -f5NT służy do określania, że hasła są w formacie uwierzytelniania NT LAN Manager (NTLM).

Ta metoda nie jest najszybszym sposobem na złamanie hasła, ale jest skuteczna. Wielu hakerów pozostawia program taki jak John the Ripper uruchomiony na komputerze przeznaczonym do łamania haseł przez wiele dni

PODSUMOWANIE MODUŁU

• Kryptografia istnieje od tysięcy lat, od egipskich hieroglifów po maszynę Enigma i dalej, aż do XXI wieku.

• Szyfrogram to dane, które zostały zaszyfrowane; tekst jawny, zwany również tekstem jawnym, to dane, które może odczytać każdy.

• Kryptografia symetryczna używa jednego klucza do szyfrowania i odszyfrowywania danych. Zarówno nadawca, jak i odbiorca muszą uzgodnić klucz przed przesłaniem danych. Dwa główne typy algorytmów symetrycznych to szyfry blokowe i szyfry strumieniowe. Szyfry blokowe, takie jak AES, działają na stałych fragmentach danych, a szyfry strumieniowe, takie jak RC4, działają na jednym bicie danych na raz.

• Kryptografia asymetryczna, zwana również kryptografią klucza publicznego, używa dwóch kluczy: jednego klucza do szyfrowania i drugiego do odszyfrowywania danych. W kryptografii klucza publicznego klucz publiczny można pobrać ze strony internetowej i jest on matematycznie powiązany z kluczem prywatnym znanym tylko właścicielowi. Klucz prywatny nigdy nie jest udostępniany.

• RSA, ECC i ElGamal używają funkcji jednokierunkowej do generowania klucza, który może być używany do podpisów cyfrowych i szyfrowania. Wymagają systemu dystrybucji kluczy, takiego jak PKI.

• Diffie-Hellman to system dystrybucji kluczy i jest jednym ze składników, który może być używany przez RSA i inne systemy kryptografii klucza publicznego.

• Digital Signature Standard (DSS) zapewnia, że podpisy cyfrowe mogą być weryfikowane. Aby utworzyć podpis cyfrowy, wartość skrótu musi być zaszyfrowana kluczem prywatnym nadawcy.

• OpenPGP to bezpłatny standard szyfrowania klucza publicznego oparty na programie szyfrowania wiadomości e-mail PGP. S/MIME to kolejny standard szyfrowania klucza publicznego, zawarty w programie Microsoft Outlook, do szyfrowania wiadomości e-mail.

• Algorytmy haszujące są używane do weryfikacji integralności danych. SHA-1 to powszechnie używany algorytm haszujący, ale ze względu na niedawno odkryte słabości NIST nie zaleca już jego używania w przypadku wrażliwych aplikacji, a agencje federalne przeszły na SHA-2 i SHA-3. • Infrastruktura klucza publicznego (PKI) to struktura składająca się z kilku komponentów służących do szyfrowania danych. PKI obejmuje protokoły, programy i zasady bezpieczeństwa oraz wykorzystuje kryptografię klucza publicznego do ochrony danych przesyłanych przez Internet.

• Certyfikat cyfrowy to plik wydany przez urząd certyfikacji (CA), który wiąże klucz publiczny z informacjami o jego właścicielu. CA to zaufana strona trzecia, która akceptuje wnioski o certyfikaty od podmiotów, uwierzytelnia aplikacje, wydaje certyfikaty i przechowuje informacje o certyfikatach.

• W kryptografii atak aktywny obejmuje wysyłanie danych wejściowych do kryptosystemu. Przykłady ataków aktywnych obejmują ataki siłowe, ataki typu man-in-the-middle, ataki typu replay i ataki słownikowe.

• Pasywny atak na kryptosystem wykorzystuje narzędzia do podsłuchiwania, takie jak Wireshark i tcpdump, w celu zbierania wiadomości z i/lub do danego kryptosystemu.