O'REILLY®

Bezpieczeństwo nowoczesnych aplikacji internetowych

Przewodnik po zabezpieczeniach

Helion

Andrew Hoffman

Tytuł oryginału: Web Application Security Exploitation and Countermeasures for Modern Web Applications

Tłumaczenie: Joanna Zatorska

ISBN: 978-83-283-7005-0

Helion SA
ul. Kościuszki 1c, 44-100 Gliwice
tel. 32 231 22 19, 32 230 98 63
e-mail: *helion@helion.pl*
WWW: *http://helion.pl* (księgarnia internetowa, katalog książek)

Drogi Czytelniku!
Jeżeli chcesz ocenić tę książkę, zajrzyj pod adres
http://helion.pl/user/opinie/beznoa
Możesz tam wpisać swoje uwagi, spostrzeżenia, recenzję.

Printed in Poland.

Spis treści

Część II. Ofensywa

Część III. Obrona

ROZDZIAŁ 1.

Historia bezpieczeństwa oprogramowania

Zanim zagłębimy się w tajniki ofensywnych i defensywnych technik dotyczących bezpieczeństwa, musimy choćby pobieżnie poznać długą i ciekawą historię bezpieczeństwa oprogramowania. Krótki opis najważniejszych wydarzeń z ostatniego stulecia powinien wystarczyć do zrozumienia podstaw technologicznych współczesnych aplikacji internetowych. Zaprezentuję przy tym relacje, jakie zachodzą między rozwojem zabezpieczeń a improwizacją myślących perspektywicznie hakerów, szukających sposobów ich łamania lub obchodzenia.

Początki hakerstwa

W ostatnich dwóch dziesięcioleciach hakerzy zyskali znacznie większy rozgłos i złą sławę niż kiedykolwiek wcześniej. W efekcie osoby niezwiązane z tą branżą zakładają, że proceder hakerski jest ściśle powiązany z internetem i że większość hakerów zaczęła się parać tym zajęciem w ciągu ostatnich 20 lat.

Ale to tylko część prawdy. Chociaż liczba hakerów na świecie zdecydowanie wzrosła wraz z rozwojem technologii World Wide Web, to jednak przedstawiciele tej profesji działali już w połowie XX w., a może i wcześniej. Wszystko zależy od przyjętej definicji słowa „hakowanie". Wielu ekspertów zastanawia się nad dekadą, w której narodziło się współczesne hakerstwo, ponieważ kilka istotnych wydarzeń na początku XX w. dość dobrze przypomina obecne działania hakerów.

Przykładem są pojedyncze incydenty z pierwszej i drugiej dekady XX w., które w tamtych czasach uznano by za hakerstwo. Większość z nich dotyczyła manipulacji urządzeniami do wysyłania i odbierania kodu Morse'a lub zakłócania transmisji radiowej. Jednak chociaż te wydarzenia miały miejsce, nie były rozpowszechnione i trudno wskazać duże przedsięwzięcia, które ucierpiałyby na skutek tych działań.

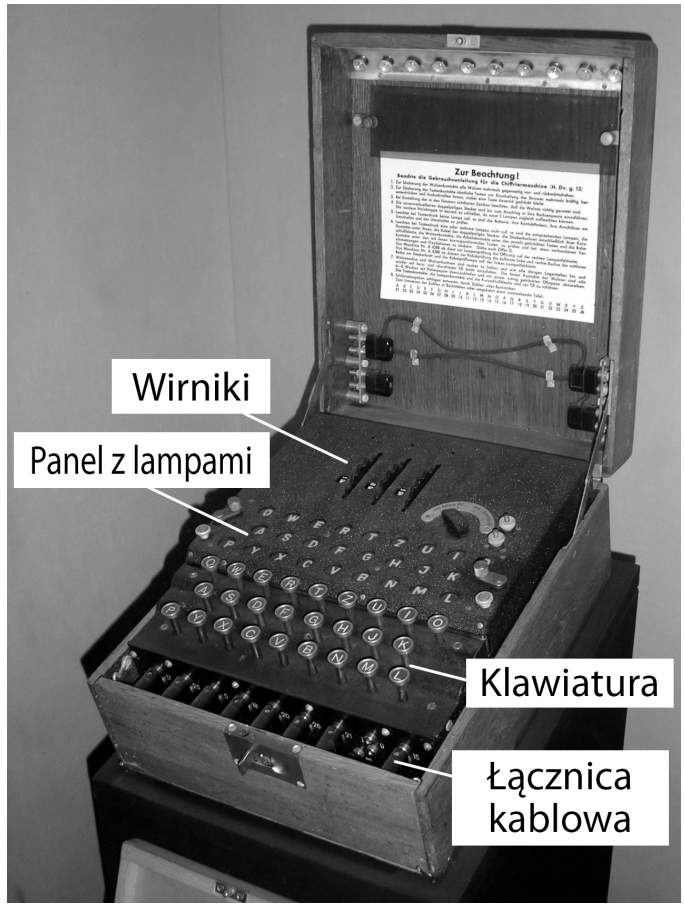
Muszę przypomnieć, że nie jestem historykiem. Jestem specjalistą ds. bezpieczeństwa. Zajmowałem się rozwiązywaniem problemów z bezpieczeństwem na poziomie architektury oraz kodu oprogramowania korporacyjnego. Wcześniej spędziłem wiele lat na stanowisku programisty. Pisałem aplikacje internetowe w różnych językach i z wykorzystaniem różnych platform. Obecnie nadal piszę oprogramowanie służące do automatyzacji zabezpieczeń, a w ramach hobby tworzę rozmaite inne projekty. Oznacza to, że nie zamierzam się rozwodzić nad szczegółami czy wymyślać alternatywne zakończenia różnych wydarzeń. Ten rozdział jest wynikiem wielu lat niezależnych badań. Chcę

pokazać, że na podstawie wydarzeń historycznych można wyciągnąć wnioski, które przydadzą się we współczesnych przedsięwzięciach.

Ponieważ w tym rozdziale nie omawiam historii kompleksowo, ale chcę tylko pobieżnie opisać najważniejsze wydarzenia, zacznę od początku lat 30. XX w. Zapraszam zatem do zapoznania się z kilkoma historycznymi wydarzeniami, które wpłynęły na kształt relacji między współczesnymi hakerami a inżynierami.

Enigma — ok. 1930 r.

Enigma była maszyną, która za pomocą elektrycznych wirników mechanicznych szyfrowała i rozszyfrowywała wiadomości tekstowe wysyłane przez fale radiowe (rysunek 1.1). Urządzenie to zostało skonstruowane w Niemczech i zyskało duże znaczenie podczas II wojny światowej.



Rysunek 1.1. Enigma

Urządzenie wyglądało jak duża prostokątna maszyna do pisania. Każde uderzenie klawisza wprawiało w ruch wirniki, co prowadziło do wybrania pozornie losowego znaku, który był przesyłany do wszystkich maszyn Enigma znajdujących się w pobliżu. Przesyłane znaki nie były jednak losowe, ale powstawały na skutek obrotu wirnika oraz przełączania różnych opcji, które można było w dowolnym czasie modyfikować. Każda maszyna Enigma skonfigurowana w odpowiedni sposób mogła odczytać, czyli „rozszyfrować", wiadomości nadawane przez maszynę z taką samą konfiguracją. Dzięki temu Enigma była niesłychanie cennym urządzeniem, umożliwiającym wysyłanie kluczowych informacji bez możliwości ich przechwycenia.

Chociaż trudno jednoznacznie stwierdzić, kto wynalazł mechanizm szyfrowania za pomocą obrotowych wirników, technologię tę spopularyzowała dwuosobowa niemiecka firma Chiffriermaschinen AG. W latach 20. XX w. właściciele Chiffriermaschinen AG podróżowali po kraju z pokazami prezentującymi możliwości maszyny. Ostatecznie zainteresowała się nią armia niemiecka, która zaczęła z niej korzystać w 1928 r. do szyfrowania ściśle tajnych wiadomości wojskowych.

Możliwość uniknięcia przechwycenia wiadomości wysyłanych na dużą odległość stała się niesłychanym osiągnięciem, które nie było wcześniej możliwe. Obecnie przechwytywanie komunikacji w świecie oprogramowania jest nadal popularną techniką, którą starają się wykorzystać hakerzy. Proceder ten określa się zwykle mianem ataków *man in the middle*. Do ochrony współczesnego oprogramowania przed tymi atakami wykorzystuje się techniki podobne (ale znacznie bardziej wyrafinowane) do stosowanych w Enigmie sto lat temu.

Chociaż technologia Enigmy była oszałamiająca, nie była pozbawiona usterek. Ponieważ jedynym kryterium przechwycenia i rozszyfrowania informacji była taka sama konfiguracja drugiej Enigmy, przechwycenie jednej wiadomości o konfiguracji (w obecnej terminologii *klucza prywatnego*) mogło zniweczyć sens całej sieci urządzeń Enigma.

Aby temu zaradzić, wszystkie grupy komunikujące się za pośrednictwem Enigmy regularnie zmieniały ustawienia. Zmiana konfiguracji Enigmy była czasochłonna. Najpierw należało osobiście pobrać zapis konfiguracji, ponieważ nie istniały jeszcze zdalne sposoby wymiany takich danych. Wymiana zapisów konfiguracji między siecią dwóch maszyn a dwoma operatorami nie jest problematyczna, jednak w przypadku większej sieci, złożonej np. z 20 maszyn, trzeba było wykorzystać kilku kurierów dostarczających dane konfiguracyjne. Każdy z nich zwiększał prawdopodobieństwo przechwycenia lub kradzieży danych konfiguracyjnych, a także ich wycieku lub sprzedaży.

Taki sposób udostępniania danych konfiguracyjnych prowadził do drugiego problemu. Otóż aby umożliwić odczyt, szyfrowanie i rozszyfrowywanie nowych komunikatów przesyłanych przez inne Enigmy, trzeba było ręcznie dostosować elementy maszyny. Wymagało to obecności wyspecjalizowanego i wyszkolonego pracownika, który mógł w razie potrzeby zmienić konfigurację. Wszystko to odbywało się w erze, w której nie istniało jeszcze oprogramowanie, a zatem poprawki w konfiguracji wymagały fizycznej zmiany układu kabli na łącznicy. Pracownik musiał mieć wykształcenie elektroniczne, które było rzadkością w początkach XX stulecia.

Ze względu na trudną i czasochłonną konfigurację tych maszyn aktualizowano je zwykle raz w miesiącu, a w przypadku linii komunikacyjnych o krytycznym znaczeniu dla misji — codziennie. Gdyby informacje o kluczu wyciekły lub zostały przechwycone, wszystkie dane przesyłane

przez pozostałą część miesiąca trafiłyby w niepożądane ręce osoby będącej odpowiednikiem współczesnego hakera.

Zastosowany w Enigmie mechanizm szyfrowania jest znany pod nazwą algorytmu klucza symetrycznego. To specjalny typ szyfru umożliwiający szyfrowanie i rozszyfrowywanie wiadomości za pomocą jednego klucza kryptograficznego. Tę rodzinę szyfrów wykorzystuje się nadal w oprogramowaniu do zabezpieczania przesyłanych danych (między nadawcą a odbiorcą). Jest to jednak znacznie ulepszona wersja klasycznego modelu, który zyskał popularność wraz z Enigmą.

Klucze wykorzystywane w oprogramowaniu mogą być dużo bardziej złożone. Nowoczesne algorytmy generowania kluczy tworzą tak skomplikowane klucze, że próba ich złamania za pomocą wszystkich możliwych kombinacji (ataki *brute force*) na najszybszych nowoczesnych komputerach zajęłaby ponad milion lat. Ponadto, w przeciwieństwie do Enigmy, obecnie wykorzystywane klucze programistyczne można szybko zmieniać.

W zależności od przypadku użycia klucze można generować dla poszczególnych sesji użytkownika (logowanie), dla każdego żądania sieciowego lub w zaplanowanych interwałach czasowych. Jeśli tego typu szyfrowanie wykorzystuje się w oprogramowaniu, a klucze są generowane dla każdego żądania, przechwycenie klucza może doprowadzić do przejęcia jednego żądania sieciowego. W najgorszym przypadku, jeśli klucze są generowane podczas logowania (na czas sesji), aplikacja będzie narażona na atak przez kilka godzin.

Jeśli prześledzimy historię nowoczesnej kryptografii, dotrzemy do II wojny światowej, która rozpoczęła się w latach 30. XX w. Można bez przesady stwierdzić, że Enigma była krokiem milowym w technologiach zabezpieczających zdalną komunikację. Okazała się podstawowym wynalazkiem w dziedzinie, którą później zaczęto nazywać bezpieczeństwem oprogramowania.

Enigma była też ważnym osiągnięciem technologicznym dla tych, których później ochrzczono mianem „hakerów". Wykorzystanie maszyn Enigma przez państwa osi podczas II wojny światowej okazało się niesłychanie silnym impulsem do rozwijania technik łamania szyfrów przez aliantów. Generał Dwight D. Eisenhower uważał, że powodzenie na tym polu jest niezbędne do zwycięstwa nad nazistami.

We wrześniu 1932 r. polski matematyk Marian Rejewski otrzymał skradzioną Enigmę. W tym samym czasie francuski szpieg Hans-Thilo Schmidt zdołał mu dostarczyć ważną konfigurację na wrzesień i październik 1932 r. Dzięki temu Rejewski mógł przechwytywać wiadomości, na podstawie których zaczął analizować tajniki szyfrowania Enigmy.

Rejewski próbował odkryć mechanizmy działania maszyny, zarówno te sprzętowe, jak i zasady matematyczne stosowanych szyfrów. Chciał zrozumieć, jak określona konfiguracja sprzętu wpływa na uzyskanie zupełnie innego szyfru.

Próby rozszyfrowania komunikatów poczynione przez Rejewskiego opierały się na kilku teoriach, które miały przewidzieć, jak określona konfiguracja sprzętu wpływa na uzyskanie określonego wyniku. Poprzez analizę wzorców w zaszyfrowanych komunikatach i formułowanie teorii na podstawie mechanizmów maszyny Rejewskiemu i jego współpracownikom: Jerzemu Różyckiemu i Henrykowi Zygalskiemu, udało się zrozumieć funkcjonowanie systemu. Dzięki dogłębnemu zrozumieniu zasad działania wirników Enigmy oraz konfiguracji złączy zespół mógł przewidzieć, jaka konfigura-

cja doprowadzi do powstania konkretnych wzorców szyfrowania. Badacze zmieniali konfigurację złączy z dość dobrą dokładnością i po kilku próbach zaczęli nasłuchiwać zaszyfrowanych komunikatów radiowych. Przed 1933 r. zespół codziennie przechwytywał i rozszyfrowywał komunikaty Enigmy.

Podobnie jak współcześni hakerzy, Rejewski i jego zespół przechwytywali komunikaty i korzystając z inżynierii odwrotnej, odczytywali schematy szyfrowania, uzyskując dostęp do cennych danych wygenerowanych przez inne źródła niż oni sami. Z tego względu uznałbym Mariana Rejewskiego i jego zespół za swego rodzaju pierwszych hakerów na świecie.

W kolejnych latach Niemcy stale podwyższali złożoność szyfrowania maszyn Enigma. W tym celu stopniowo zwiększali liczbę wirników szyfrujących znaki. Ostatecznie złożoność inżynierii odwrotnej, jakiej należało poddać konfigurację, przekroczyła możliwości zespołu Rejewskiego, który nie był już zdolny złamać kodu w rozsądnym czasie. Obrazuje to stale ewoluujące relacje między hakerami a tymi, którzy chcą zapobiec ich działalności.

Te relacje trwają do dziś. Kreatywni hakerzy nieustannie ulepszają swoje techniki umożliwiające włamywanie się do systemów oprogramowania. Z drugiej strony, inteligentni inżynierowie nieustannie rozwijają nowe techniki obronne przeciwko najbardziej kreatywnym hakerom.

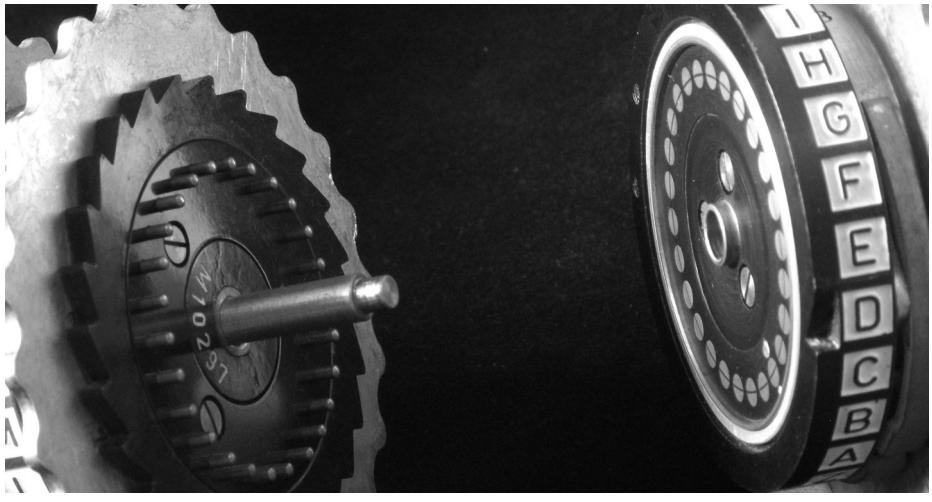
Automatyczne łamanie kodu Enigmy — ok. 1940 r.

Alan Turing był angielskim matematykiem, znanym głównie jako twórca testu nazywanego obecnie „testem Turinga". Test ten oceniał generowane przez maszyny konwersacje, szacując trudność w odróżnieniu ich od rozmów prowadzonych przez ludzi. Ten test zwykle uważa się za podstawę filozofii sztucznej inteligencji (SI).

Co prawda Alana Turinga znamy głównie ze względu na jego wkład w dziedzinie SI, ale był on również pionierem na polu kryptografii i automatyzacji. W rzeczywistości przed wybuchem II wojny światowej i w jej trakcie poświęcał się głównie kryptografii, a nie sztucznej inteligencji. Począwszy od września 1938 r., pracował na część etatu w Rządowej Szkole Kodów i Szyfrów (ang. *Government Code and Cypher School* — GC&CS). Była to założona przez armię brytyjską agencja badawcza i wywiadowcza, która mieściła się w Bletchley Park w Anglii.

Badania prowadzone przez Turinga koncentrowały się na analizie działania Enigmy. W Bletchley Park badał on kryptograficzne mechanizmy stosowane w Enigmie. Współpracował przy tym ze swoim mentorem Dillym Knoksem, który był już doświadczonym kryptografem.

Podobnie jak wcześniej polscy matematycy, Alan i Dilly chcieli znaleźć sposób na złamanie (już znacznie bardziej skomplikowanego) szyfru niemieckich urządzeń Enigma. Dzięki partnerstwu z polskim Biurem Szyfrów uzyskali dostęp do materiałów badawczych, które zespół Rejewskiego opracował niemal dekadę wcześniej. Dzięki temu mogli dogłębnie zrozumieć działanie maszyny. Znali relację między wirnikami a okablowaniem i wiedzieli, jak konfiguracja maszyny wpływa na wynik szyfrowania komunikatu (rysunek 1.2).



Rysunek 1.2. Para wirników Enigmy wykorzystywana do kalibrowania konfiguracji, czyli analogowy odpowiednik zmiany klucza głównego szyfru cyfrowego

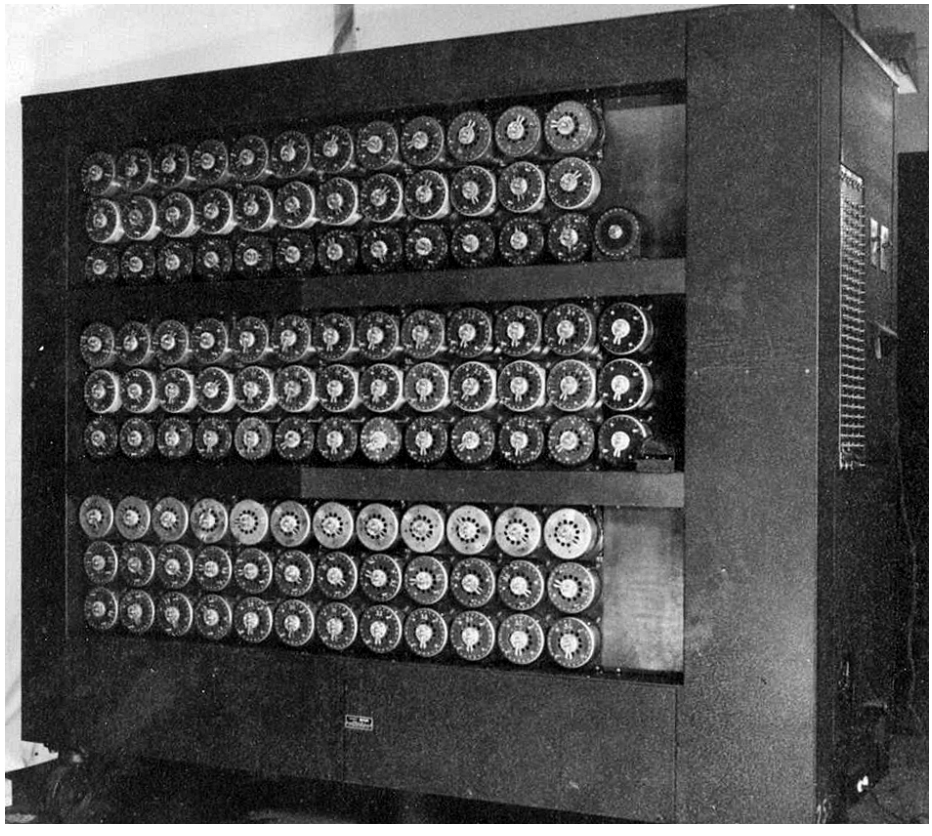
Zespół Rejewskiego zdołał dostrzec wzorce szyfrowania, na podstawie których można było przewidywać konfigurację maszyny. Jednak to rozwiązanie nie skalowało się do dziesięciokrotnie większej liczby wirników. W czasie, jaki należałoby poświęcić na wypróbowanie wszystkich możliwych kombinacji, przeciwnicy zdążyliby już zmienić konfigurację. To dlatego Alan i Dilly szukali innego rozwiązania. Zależało im na skalowalności i możliwości łamania nowych rodzajów szyfrów. Potrzebne im było rozwiązanie ogólne, a nie wysoce wyspecjalizowane.

Poznaj Bombę

Bomba była elektrycznym urządzeniem, które miało automatycznie stosować zasady inżynierii odwrotnej w celu wyznaczenia położenia mechanicznych wirników Enigmy. Było to możliwe na podstawie mechanicznej analizy wiadomości przesyłanych przez te maszyny (rysunek 1.3).

Pierwsze Bomby zbudowali Polacy, próbując zautomatyzować pracę Rejewskiego. Niestety, te urządzenia zostały zaprojektowane z myślą o znalezieniu konfiguracji Enigmy o określonej specyfikacji technicznej. Nie nadawały się do użycia z maszynami wyposażonymi w ponad trzy wirniki. Ponieważ Bomby opracowanej przez Polaków nie można było dostosować do coraz bardziej skomplikowanych wersji Enigmy, polscy kryptolodzy ostatecznie powrócili do ręcznych prób rozszyfrowywania niemieckich komunikatów wojskowych.

Alan Turing uważał, że oryginalne maszyny nie zdały egzaminu, ponieważ nie były wystarczająco ogólne. Opracowując urządzenie, które zdołałoby rozszyfrować dowolną konfigurację Enigmy (niezależnie od liczby wirników), przyjął proste założenie. Uznał, że aby zaprojektować algorytm umożliwiający rozszyfrowanie zaszyfrowanej wiadomości, trzeba znać występujące w niej słowo lub frazę, a także ich pozycję.



Rysunek 1.3. Wczesna wersja Bomby skonstruowanej w Bletchley Park i wykorzystywanej podczas II wojny światowej (warto zwrócić uwagę na wiele rzędów wirników służących do rozszyfrowywania szybko zmieniającej się konfiguracji Enigmy)

Na szczęście dla Turinga niemieccy wojskowi trzymali się bardzo surowych standardów komunikacyjnych. Każdego dnia nadawali przez radio zaszyfrowane przez Enigmę komunikaty, zawierające szczegółową regionalną prognozę pogody. Dzięki temu mieli pewność, że wszystkie jednostki wojskowe znają warunki pogodowe, a jednocześnie unikali możliwości podsłuchania ich przez radio. Niemcy nie wiedzieli jednak, że zespół Turinga zdoła na podstawie tych raportów wyznaczyć ich cel i pozycję.

Znając dane wejściowe (o pogodzie) wysyłane przez odpowiednio skonfigurowaną Enigmę, można było znacznie łatwiej wyznaczyć dane wyjściowe. Turing wykorzystał tę nową wiedzę do opracowania konfiguracji Bomby, która działała niezależnie od liczby wirników Enigmy, której kod próbowano złamać.

Naukowiec złożył wniosek o sfinansowanie Bomby, która mogłaby dokładnie wyznaczyć konfigurację potrzebną do przechwycenia i odczytania zaszyfrowanych komunikatów wysyłanych przez niemieckie Enigmy. Po uzyskaniu akceptacji budżetu Turing skonstruował Bombę składającą się ze 108 bębenków, które mogły się obracać z szybkością 120 obrotów na minutę. Ta ma-

szyna mogła przeanalizować prawie 20 000 możliwych wersji konfiguracji Enigmy w ciągu zaledwie 20 minut. Dzięki temu można było z łatwością odgadnąć każdą nową konfigurację. Szyfr Enigmy nie był już bezpiecznym środkiem komunikacji.

Podejście Turinga do inżynierii odwrotnej nazywa się obecnie atakiem ze znanym tekstem jawnym (ang. *Known Plaintext Attack* — KPA). Jest to algorytm, który działa dużo skuteczniej dzięki podaniu znanych wcześniej danych wejściowych i wyjściowych. Podobne techniki wykorzystują współcześni hakerzy do łamania szyfrów stosowanych do zabezpieczania danych zapisanych w bazie lub używanych w oprogramowaniu. Zbudowane przez Turinga urządzenie jest ważnym punktem w historii, ponieważ było to pierwsze zautomatyzowane urządzenie hakerskie.

Phreaking telefonów — ok. 1950 r.

Po okresie popularności Enigmy w latach 30. XX w. i kryptologicznej bitwie między największymi światowymi mocarstwami, kolejnym ważnym wydarzeniem jest pojawienie się telefonii. Dzięki telefonom zwykli ludzie mogli się szybko komunikować na duże odległości. W miarę rozrastania się sieci telefonicznych narodziła się potrzeba ich automatyzacji umożliwiającej działanie na coraz większą skalę.

Pod koniec lat 50. XX w. telekomy takie jak AT&T zaczęły wdrażać nowe telefony, które umożliwiały automatyczne przekierowanie rozmów do odbiorcy na podstawie sygnałów dźwiękowych emitowanych przez urządzenia. Po naciśnięciu klawisza na tarczy telefonu urządzenie emitowało dźwięki o określonej częstotliwości, które były transmitowane kablem do maszyny w centrali. Przełącznik przetwarzał te dźwięki na liczby i przekierowywał rozmowę do odpowiedniego odbiorcy.

Ten system nazywano *wybieraniem tonowym*. Stał się on podstawą rozwoju sieci telefonicznych na dużą skalę. Wybieranie tonowe znacznie zredukowało niedogodności związane z działaniem sieci telefonicznych, gdyż wyeliminowało konieczność obsługi przez operatora, który musiał ręcznie przełączać każdą rozmowę. W rezultacie jeden operator nadzorujący sieć mógł obsłużyć setki rozmów w czasie, jaki wcześniej musiałby poświęcić na obsługę jednego połączenia.

W krótkim czasie mała grupa ludzi zorientowała się, że dowolny system oparty na interpretacji sygnałów dźwiękowych można z łatwością zmanipulować. Wystarczyło nauczyć się odtwarzać dźwięki o identycznej częstotliwości w pobliżu odbiornika telefonicznego, aby zakłócić jego działanie. Hobbyści, którzy eksperymentowali w taki sposób, ostatecznie zasłynęli jako *phreakerzy*, czyli hakerzy specjalizujący się w manipulowaniu sieciami telefonicznymi oraz ich łamaniu. Prawdziwe pochodzenie słowa *phreaking* nie jest znane, chociaż rozważa się kilka możliwości. Najczęściej spotyka się wyjaśnienie, że słowo to jest wynikiem połączenia dwóch wyrazów: *freaking* (maniakalny) i *phone* (telefon).

Istnieje też alternatywne wyjaśnienie genezy tego określenia, według mnie bardziej sensowne. Wydaje mi się, że słowo *phreaking* pochodzi od frazy *audio frequency* (częstotliwość audio), ponieważ wiąże się z językami sygnałów audio wykorzystywanych w ówczesnych aparatach telefonicznych. Uważam, że to wyjaśnienie jest lepsze, ponieważ pojawienie się analizowanego terminu zbiegło się z wdrożeniem oryginalnego systemu wybierania tonowego firmy AT&T. Przed wdro-

żeniem tej technologii rozmowy telefoniczne trudno było zakłócić, gdyż każda wymagała obsługi przez operatora, który łączył ze sobą dwie linie.

Phreaking można powiązać z kilkoma istotnymi zdarzeniami, ale najgłośniejszym przykładem wczesnego zastosowania tej techniki było wykrycie i wykorzystanie tonu o częstotliwości 2600 Hz. Firma AT&T używała tej częstotliwości do sygnalizacji zakończenia rozmowy. Był to odpowiednik „polecenia administratora" wbudowany w oryginalny system wybierania tonowego. Wysłanie sygnału o częstotliwości 2600 Hz zatrzymywało obsługę otwartego połączenia przez system przełączania telekomu (rozmowa była rejestrowana jako zakończona, chociaż trwała nadal). Dzięki temu można było prowadzić kosztowne rozmowy międzynarodowe bez rejestrowania ich na rachunku dzwoniącego.

Odkrycie działania tonu o częstotliwości 2600 Hz zwykle przypisuje się dwóm zdarzeniom. Pewien młody chłopak, Jo Engressia, miał gwizdek emitujący dźwięk o częstotliwości 2600 Hz i zgodnie z zapiskami udowodnił swoim kolegom, że gwizdanie kończy rozmowy telefoniczne. Niektórzy uważają, że Jo był jednym z pierwszych phreakerów, chociaż dokonał swojego odkrycia przez przypadek.

Jakiś czas później John Draper, kolega Engressia, odkrył, że zabawkowe gwizdki dołączane do paczek płatków Cap'n Crunch wydawały dźwięk o częstotliwości 2600 Hz. Ostrożne użycie gwizdka umożliwiało też wykonywanie darmowych rozmów na dalekie odległości. Znajomość tych technik rozpowszechniła się na Zachodzie, co ostatecznie doprowadziło do opracowania sprzętu, który mógł dopasować określone częstotliwości audio poprzez naciśnięcie przycisku.

Pierwsze z tych urządzeń nazwano *niebieską skrzynką*. Niebieskie skrzynki emitowały prawie idealne sygnały o częstotliwości 2600 Hz, dzięki czemu każdy, kto dysponował tym urządzeniem, mógł wykonywać darmowe połączenia, wykorzystując błąd systemów przełączania w telekomach. Niebieskie skrzynki były jedynie pionierami spośród kolejnych automatycznych urządzeń stosowanych w phreakingu. Późniejsze generacje phreakerów manipulowały płatnymi rozmowami, zapobiegając rejestracji rozmowy bez użycia częstotliwości 2600 Hz, emulowały wojskowe sygnały komunikacyjne, a nawet fałszowały identyfikatory dzwoniącego.

Na podstawie tych wydarzeń można wysnuć wniosek, że architekci wczesnych sieci telefonicznych uwzględniali tylko zwykłych ludzi i ich potrzeby komunikacyjne. We współczesnym świecie oprogramowania tego typu podejście nazywa się obsługą „najlepszego scenariusza". Takie podejście było poważnym błędem, który jednak okazał się ważną lekcją, istotną również dzisiaj: podczas projektowania złożonych systemów zawsze należy najpierw uwzględnić najgorszy scenariusz.

Ostatecznie znajomość słabych punktów systemów wybierania tonowego stała się powszechna. W efekcie zainwestowano w rozwój środków zaradczych w celu ochrony zysków telekomów oraz uniemożliwienia naruszania integralności rozmów przez phreakerów.

Technologia antyphreakingowa — ok. 1960 r.

W latach 60. XX w. telefony były wyposażone w nową technologię wybierania dwutonowego (ang. *Dual-Tone Multifrequency* — DTMF). DTMF był językiem sygnałów audio opracowanym przez Bell Systems i opatentowanym pod szerzej znanym znakiem towarowym Touch Tones.

Technika DTMF była nieodłączną częścią znanego obecnie układu klawiszy w aparacie telefonicznym, w postaci trzech kolumn i czterech wierszy cyfr. Każdy klawisz w telefonie DTMF emitował sygnały audio o określonych częstotliwościach, w przeciwieństwie do pojedynczej częstotliwości w oryginalnych systemach wybierania tonowego.

Następująca tabela przedstawia częstotliwości audio (w hercach) emitowane po naciśnięciu klawiszy na dawnych telefonach:

```
1          2          3          (697Hz)
4          5          6          (770Hz)
7          8          9          (852Hz)
*          0          #          (941Hz)
(1209 Hz) (1336 Hz) (1477 Hz)
```

Impulsem do rozwoju DTMF było wykorzystywanie przez phreakerów słabości systemów wybierania tonowego, które były bardzo podatne na inżynierię odwrotną. W Bell Systems uważano, że ponieważ system DTMF wykorzystywał jednocześnie dwa różne tony, agresorzy będą mieli znacznie utrudnione zadanie.

Tonów DTMF nie dawało się łatwo odtworzyć ludzkim głosem ani za pomocą gwizdka, co oznaczało, że ta technika jest znacznie bezpieczniejsza niż poprzednia. DTMF była pierwszym przykładem powodzenia w rozwoju zabezpieczeń, które stanowiło odpowiedź na ataki phreakerów, czyli ówczesnych hakerów.

Mechanizm generowania tonów DTMF jest dość prosty. Pod każdym klawiszem znajduje się przełącznik, który sygnalizuje konieczność wyemitowania dwóch częstotliwości przez wewnętrzny głośnik: jednej, wynikającej z wiersza, w którym znajduje się klawisz, i drugiej, wynikającej z kolumny. Stąd wzięła się nazwa *dwutonowy*.

Technika DTMF została uznana za standard przez Międzynarodowy Związek Telekomunikacyjny (ang. *International Telecommunication Union* — ITU) i nieco później znalazła zastosowanie w telewizji kablowej (w celu ustalania przerw na reklamy).

DTMF jest istotnym wynalazkiem technologicznym, ponieważ na jego podstawie widać, że systemy można zaprojektować w sposób utrudniający naruszenia. Trzeba je tylko odpowiednio zaplanować. Warto zauważyć, że tony DTMF ostatecznie udało się zmanipulować, ale wymagało to naprawdę sporego wysiłku. Ostatecznie w centralach telefonicznych zaczęto wykorzystywać wejścia cyfrowe (a nie analogowe), co niemal całkowicie wyeliminowało phreaking.

Początki hakowania komputerów — ok. 1980 r.

W 1976 r. firma Apple zaczęła sprzedawać osobisty komputer apple 1. Nie był on wstępnie skonfigurowany i kupujący musiał się zaopatrzyć w kilka różnych komponentów, które należało połączyć z płytą główną. Zbudowano i sprzedano tylko kilkaset takich urządzeń.

W 1982 r. firma Commodore International wypuściła na rynek konkurencyjne urządzenie. Był to commodore 64, komputer osobisty całkowicie gotowy do użycia od razu po zakupie. Posiadał klawiaturę, obsługiwał dźwięk i można go było nawet podłączyć do monitorów kolorowych.

Aż do początków lat 90. XX w. sprzedawano niemal 500 000 egzemplarzy miesięcznie. Począwszy od tamtego czasu, sprzedaż komputerów osobistych stale rosła. Trend ten utrzymał się przez kilka następnych dekad. Wkrótce komputery stały się popularnym narzędziem w gospodarstwach domowych i w firmach. Zaczęły wykonywać typowe, powtarzalne zadania, takie jak zarządzanie aspektami finansowymi, zasobami ludzkimi, księgowością i sprzedażą.

W 1983 r. amerykański informatyk Fred Cohen napisał pierwszego wirusa komputerowego. Mógł on tworzyć kopie samego siebie i przenosił się z łatwością między komputerami osobistymi poprzez dyskietki. Cohen potrafił zapisać wirusa w zwykłym programie, maskując go przed wszystkimi, którzy nie mieli dostępu do kodu źródłowego. Cohen stał się potem znany jako pionier w dziedzinie bezpieczeństwa oprogramowania, pokazując, że wykrywanie wirusów w zwykłym oprogramowaniu za pomocą algorytmów było prawie niemożliwe.

Kilka lat później, w 1988 r., inny amerykański informatyk, Robert Morris, stał się pierwszą osobą, której udało się wdrożyć wirusa infekującego komputery poza laboratorium badawczym. Wirus stał się znany pod nazwą robaka Morrisa (*Morris Worm*), a słowo „robak" przeszło do słowników jako oznaczenie samoreplikującego się wirusa komputerowego. Robak Morrisa zainfekował pierwszego dnia po wypuszczeniu ok. 15 000 komputerów połączonych w sieć.

Pierwszy raz w historii rząd USA rozpoczął prace nad oficjalnymi regulacjami przeciwko hakowaniu. Government Accountability Office (GAO) w USA oszacowało szkody wywołane przez wirusa na 10 000 000 dolarów. Morris został skazany na 3 lata więzienia w zawieszeniu, 400 godzin prac społecznych i grzywnę w wysokości 10 050 dolarów. W ten sposób został pierwszym w historii hakerem skazanym w Stanach Zjednoczonych.

Obecnie większość hakerów nie tworzy wirusów infekujących systemy operacyjne. Celem ich ataku są przeglądarki internetowe. Nowoczesne przeglądarki udostępniają niesłychanie solidne mechanizmy piaskownicy (*sandbox*), które utrudniają uruchamianie kodu wykonywalnego z poziomu witryny poza przeglądarką (w systemie operacyjnym hosta) bez jawnego pozwolenia użytkownika.

Chociaż celem współczesnych hakerów są przede wszystkim użytkownicy i dane, do których dostęp można uzyskać poprzez przeglądarkę, ich działania w wielu aspektach przypominają działania hakerów atakujących system operacyjny. Skalowalność (przechodzenie od jednego użytkownika do drugiego) i kamuflaż (ukrywanie złośliwego kodu w legalnym programie) są technikami wykorzystywanymi podczas ataków na przeglądarki internetowe.

Skalowanie współczesnych ataków opiera się głównie na dystrybucji poprzez pocztę elektroniczną, serwisy społecznościowe i wiadomości tekstowe. Niektórzy hakerzy budują nawet sieci rzeczywistych witryn WWW, mających na celu promocję jednej złośliwej witryny.

Często złośliwy kod ukrywa się za legalnie wyglądającym interfejsem. Ataki phishingowe (przechwytujące dane dostępowe) odbywają się za pośrednictwem stron WWW wyglądających i działających tak samo jak witryny społecznościowe lub bankowe. Często okazuje się, że pewne wtyczki do przeglądarek kradną dane, a czasem hakerzy znajdują sposoby na uruchamianie swojego kodu na stronach, które do nich nie należą.

Rozwój sieci WWW — ok. 2000 r.

World Wide Web (WWW) rozwijała się w latach 90. XX w., ale jej popularność eksplodowała pod koniec tamtego stulecia i na początku wieku XXI.

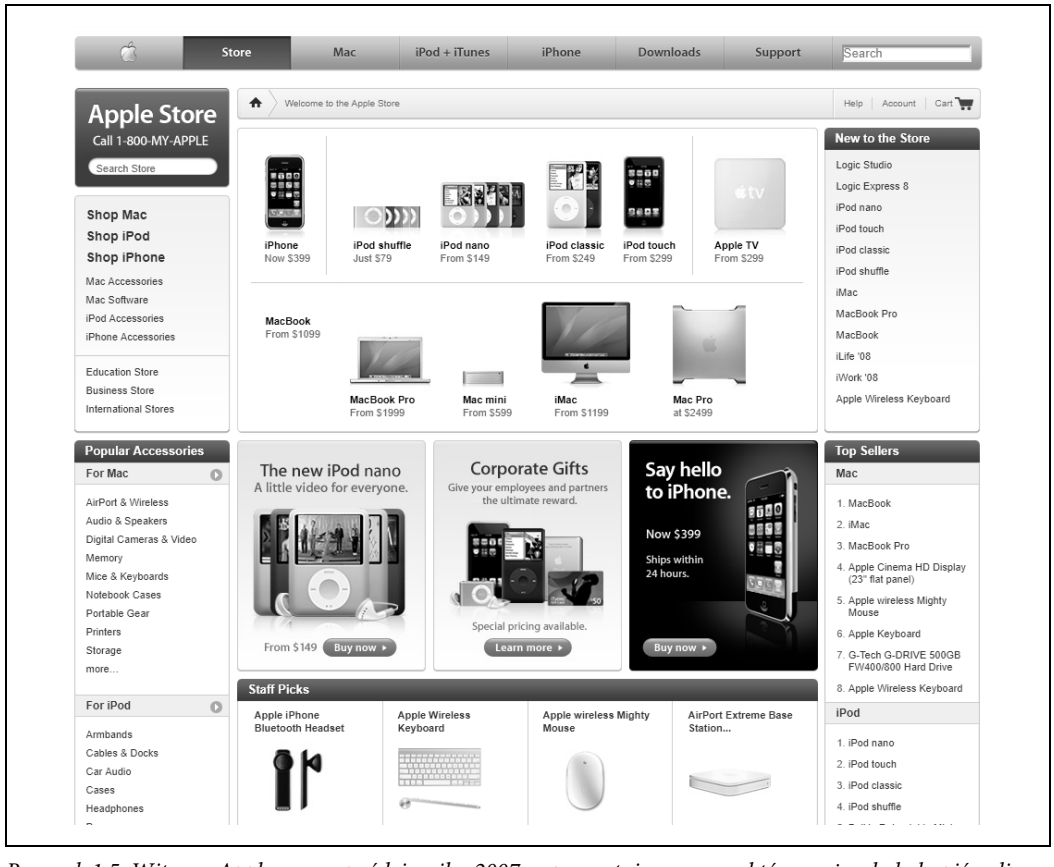
W latach 90. XX w. sieć była wykorzystywana właściwie tylko do udostępniania dokumentów w formacie HTML-a. W witrynach nie przejmowano się interfejsem użytkownika, a tylko niektóre umożliwiały wysyłanie użytkownikom danych na serwer w celu modyfikacji działania witryny. Na rysunku 1.4 widać zawierającą same informacje witrynę *Apple.com* z 1997 r.

Rysunek 1.4. Witryna Apple.com w czerwcu 1997 r.; prezentowane na niej dane miały charakter czysto informacyjny. Użytkownik nie mógł się rejestrować, logować, komentować i korzystać z danych zapisanych w sesji

Na początku XXI w. rozpoczęła się era internetowa, ponieważ witryny zaczęły gromadzić dane przesyłane przez użytkowników i na ich podstawie modyfikować działanie serwisów. Było to kluczowe osiągnięcie. Technologię tę nazwano później *Web 2.0*. Witryny tworzone na jej podstawie umożliwiały użytkownikom współpracę poprzez wysyłanie danych na serwer za pośrednictwem protokołu Hypertext Transport Protocol (HTTP). Serwer magazynował je i udostępniał innym użytkownikom na żądanie.

Ta nowa ideologia tworzenia witryn WWW umożliwiła projektowanie serwisów społecznościowych w obecnej postaci. Technologia Web 2.0 zapoczątkowała powstawanie blogów, portali typu wiki, witryn do udostępniania plików multimedialnych i wielu innych.

Dzięki tej radykalnej zmianie w ideologii przestrzeń internetowa przekształciła się z platformy do udostępniania dokumentów w platformę dystrybucji aplikacji. Rysunek 1.5 pokazuje stronę *Apple.com* w wersji z 2007 r., na której można dokonać zakupów. Warto zwrócić uwagę na link do konta użytkownika w prawym górnym rogu. Sugeruje on, że ta witryna obsługiwała konta użytkowników i zapewniała trwałość danych. Łącze do konta użytkownika istniało we wcześniejszych wersjach witryny Apple.com z początku XXI w., ale w 2007 r. umieszczono je w prawym górnym rogu interfejsu, przenosząc je z poprzedniej pozycji na dole strony. Wcześniej mogło być tylko eksperymentalną lub niezbyt często wykorzystywaną funkcją.



Rysunek 1.5. Witryna Apple.com w październiku 2007 r., prezentująca rzeczy, które można było kupić online

To ogromne przesunięcie w kierunku projektowania architektury witryn WWW zmieniło też sposób, w jaki hakerzy atakowali aplikacje internetowe. Dotychczas przeznaczano poważne nakłady na zabezpieczanie serwerów i sieci, które stanowiły dwa najważniejsze wektory ataków hakerskich w poprzedniej dekadzie. Natomiast wraz z pojawieniem się witryn przypominających aplikacje doskonałym celem hakerów stał się użytkownik.

Była to doskonała konfiguracja. Użytkownicy mieli wkrótce otrzymać dostęp do krytycznych funkcji za pośrednictwem internetu. Komunikacja wojskowa, transfery bankowe i inne miały wkrótce być obsługiwane przez aplikacje internetowe (witryny działające jak aplikacje biurkowe). Niestety w tamtych czasach istniało niewiele środków bezpieczeństwa, które mogły chronić użytkowników przed atakami. Ponadto edukacja dotycząca działań hakerów i mechanizmów funkcjonowania internetu była uboga. Tylko nieliczni użytkownicy internetu zaczynali dopiero poznawać tajniki wykorzystywanych przez nich technologii.

Na początku XXI w. pierwsze szeroko nagłośnione ataki typu *Denial of Service* (DoS) doprowadziły do całkowitych awarii Yahoo!, Amazona, eBaya i innych popularnych witryn. W 2002 r. wtyczka do przeglądarek ActiveX firmy Microsoft okazała się zawierać lukę, która umożliwiała zdalne przesyłanie i pobieranie plików przez złośliwe strony. Do połowy pierwszej dekady XXI w. hakerzy regularnie korzystali z technologii „phishingu" witryn WWW w celu kradzieży danych dostępowych. Nie istniały wówczas żadne sposoby zabezpieczenia użytkowników przed tego typu procederem.

W tamtych czasach szerzyły się ataki typu *Cross-Site Scripting* (XSS). Hakerzy wykorzystywali lukę umożliwiającą uruchamianie kodu podczas sesji użytkownika w przeglądarce wyświetlającej legalną stronę. Było to możliwe, gdyż producenci przeglądarek nie zabezpieczyli ich jeszcze przed atakami tego typu. Wiele prób włamań na początku XXI w. było możliwych, ponieważ technologia wykorzystywana przez witryny internetowe została zaprojektowana z myślą o jednym użytkowniku (właścicielu witryny). Technologie te zawodziły, jeśli za ich pomocą budowano systemy umożliwiające współdzielenie danych między wieloma użytkownikami.

Hakerzy w nowoczesnej erze — po ok. 2015 r.

Omawiając hakerską działalność w poprzednich epokach, chciałem przygotować bazę do dalszych rozważań prowadzonych w tej książce.

Rozwój i kryptoanalizę Enigmy w latach 30. XX w. opisałem, aby podkreślić znaczenie bezpieczeństwa. Okazuje się, że istnieją ludzie skłonni do ogromnych wysiłków mających na celu złamanie tego bezpieczeństwa.

W latach 40. XX w. zaczęto stosować automatyzację działań związanych z bezpieczeństwem. W tym przypadku dotyczyło to starcia między agresorami a obrońcami. Potrzeba automatyzacji wynikała ze znacznych ulepszeń Enigmy, uniemożliwiających niezawodne łamanie szyfrów za pomocą technik ręcznej kryptoanalizy. Alan Turing postawił na automatyzację, aby pokonać wspomniane ulepszenia technologiczne.

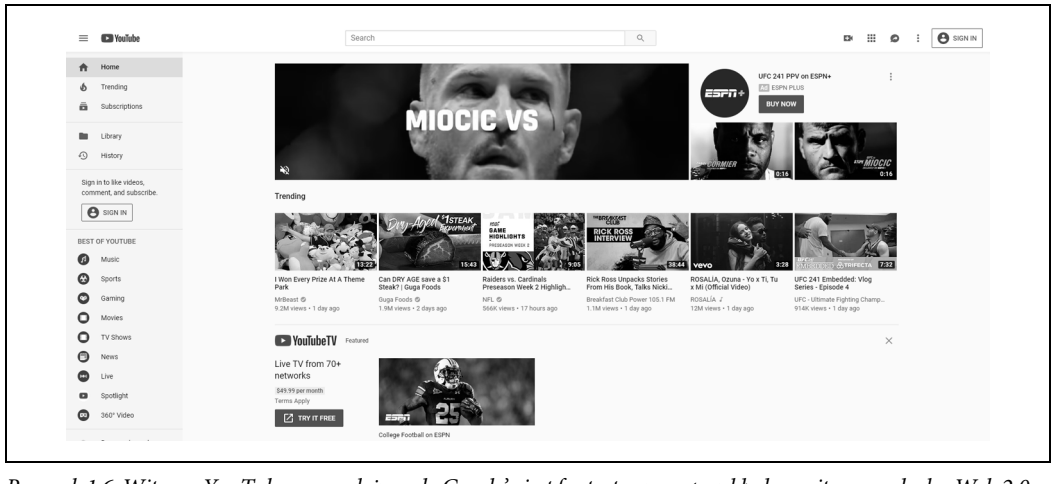
Lata 50. i 60. XX w. dowiodły, że hakerzy i myśliciele mają ze sobą wiele wspólnego. Okazało się, że projektowanie bez uwzględnienia złośliwych użytkowników prowadzi do powstania technologii, do których ostatecznie ktoś się włamie. Dlatego podczas projektowania technologii wdrażanej na dużą skalę i dla szerokiej rzeszy użytkowników trzeba zawsze uwzględniać najgorszy możliwy scenariusz.

W latach 80. XX w. popularność zaczęły zyskiwać komputery osobiste. W tym samym czasie rozpoczęła się znana obecnie działalność hakerów. Wykorzystywali oni możliwości oprogramowania, ukrywając wirusy w legalnych aplikacjach, a także rozprzestrzeniając je w sieci.

Wreszcie pojawienie się sieci World Wide Web i szybki wzrost jej popularności doprowadziły do rozwoju technologii Web 2.0, która zmieniła sposób postrzegania internetu. Z medium służącego do udostępniania dokumentów internet przekształcił się w platformę udostępniania aplikacji. W efekcie zaczęły się pojawiać nowe typy ataków — skierowane na użytkownika, a nie na sieć i serwer. Była to fundamentalna zmiana, której skutki odczuwamy do dziś. Większość współczesnych hakerów skupia się na aplikacjach internetowych działających w przeglądarkach, porzucając ataki na programy biurkowe i systemy operacyjne.

Przenieśmy się do roku 2019, w którym zacząłem pisać tę książkę. Kiedy piszę te słowa, w sieci działają tysiące stron obsługiwanych przez firmy o kapitale wynoszącym miliony i miliardy dolarów. Wiele firm czerpie wszystkie zyski ze swoich witryn internetowych. Niektóre z nich są powszechnie znane, np.: Google, Facebook, Yahoo!, Reddit, Twitter.

YouTube umożliwia wzajemne interakcje między użytkownikami oraz między użytkownikami a aplikacją (rysunek 1.6). Witryna obsługuje komentarze, przesyłanie filmów oraz obrazów. Przesyłanie każdego pliku wymaga różnych uprawnień, które umożliwiają oznaczenie przez autora, kto będzie mógł zobaczyć daną treść. Większość hostowanych danych jest zapisana na stałe i dostępna w różnych sesjach, a niektóre cechy zmieniają się w zależności od użytkownika w czasie rzeczywistym (poprzez powiadomienia). Również znaczna część krytycznych funkcji została przeniesiona na klienta (przeglądarkę) i nie jest już obsługiwana przez serwery.



Rysunek 1.6. Witryna YouTube.com należąca do Google'a jest fantastycznym przykładem witryny w duchu Web 2.0

Niektóre tradycyjne firmy produkujące oprogramowanie biurkowe próbują dostosować swoje produkty do działania w internecie, przenosząc je do środowiska znanego dzisiaj pod nazwą *chmury*. Jest to po prostu złożona sieć serwerów. Przykładem takiej firmy jest Adobe, która opracowała Creative Cloud, usługę subskrypcji udostępniającą programy Photoshop i inne za pośrednictwem sieci. Również Microsoft Office udostępnia programy Word i Excel w postaci aplikacji internetowych.

Ze względu na ogromne kwoty przewijające się przez aplikacje internetowe stawka jest wyższa niż kiedykolwiek wcześniej. Oznacza to, że współczesne aplikacje WWW stanowią doskonały cel naruszeń, a korzyści z tego procederu są niebotyczne.

Żyjemy w epoce, która oferuje najlepsze możliwości hakerom i inżynierom zajmującym się kwestiami bezpieczeństwa. Na obydwu tych obszarach istnieje wysokie zapotrzebowanie na pracowników i dotyczą one działalności po dwóch stronach prawa.

Przeglądarki są znacznie bardziej zaawansowane niż dziesięć lat temu. Temu rozwojowi towarzyszyło powstanie nowych funkcji zabezpieczeń. Postęp dokonał się też w obszarze protokołów sieciowych wykorzystywanych w dostępie do internetu.

Nowoczesne przeglądarki zapewniają solidną izolację między witrynami o różnym pochodzeniu zgodnie z zasadami bezpieczeństwa *Same Origin Policy* (SOP). Oznacza to, że witryna B nie ma dostępu do witryny A, nawet jeśli obie są otwarte jednocześnie lub gdy jedna z nich jest wbudowana w element `iframe` drugiej.

Przeglądarki honorują także nową konfigurację bezpieczeństwa, zwaną *Content Security Policy* (CSP). CSP umożliwia twórcy witryny określenie różnych poziomów zabezpieczeń. Mogą one np. dotyczyć możliwości uruchamiania skryptów wbudowanych w kod HTML-a. Dzięki temu programiści zdołają w większym stopniu zabezpieczyć aplikacje przed popularnymi zagrożeniami.

Również HTTP, główny protokół do przesyłania danych w internecie, został ulepszony pod kątem bezpieczeństwa. Zaczął korzystać z protokołów SSL i TLS, które wymuszają surowe zasady szyfrowania danych przesyłanych w sieci. To znacznie utrudnia przeprowadzanie skutecznych ataków typu *man in the middle*.

W wyniku tych ulepszeń w zabezpieczeniach przeglądarek wielu najlepszych hakerów obrało sobie za cel logikę aplikacji internetowych pisaną przez programistów. Zamiast atakować przeglądarkę, znacznie łatwiej jest skutecznie zaatakować samą witrynę, wykorzystując błędy w jej kodzie. Na szczęście dla hakerów nowoczesne aplikacje WWW są wielokrotnie większe i znacznie bardziej złożone niż aplikacje w przeszłości.

Często nowoczesne, znane aplikacje internetowe są zależne od setek bibliotek *open source*, zintegrowane z innymi witrynami i wykorzystują wiele baz danych różnego typu. Ponadto mogą działać na wielu różnych serwerach WWW, znajdujących się w różnych lokalizacjach. Tego typu aplikacje najczęściej padają ofiarą skutecznych ataków i to na nich skupiam się w tej książce.

Podsumowując: nowoczesne aplikacje internetowe są znacznie większe i bardziej złożone niż ich poprzednicy. Hakerzy mogą się obecnie skoncentrować na atakach na aplikacje internetowe, wykorzystując błędy w logice kodu. Zwykle tego typu błędy są efektem ubocznym zaawansowanych interakcji z użytkownikiem, jakie umożliwia aplikacja internetowa.

Hakerzy z ubiegłej dekady spędzali większość czasu na włamywaniu się do serwerów, sieci i przeglądarek. Współcześni hakerzy większość czasu poświęcają na włamywanie się do aplikacji internetowych, wykorzystując luki znalezione w kodzie.

Podsumowanie

Początki bezpieczeństwa oprogramowania oraz procederu hakerskiego, mającego na celu obejście zabezpieczeń, sięgają co najmniej kilkuset lat. Współczesne oprogramowanie czerpie z wniosków wysnutych na podstawie technologii z przeszłości. To samo dotyczy bezpieczeństwa oprogramowania.

Hakerzy w przeszłości atakowali aplikacje inaczej niż obecnie. W miarę jak jeden aspekt stosu aplikacji stawał się coraz bezpieczniejszy, zmieniali swój cel na nowe technologie. Zwykle te technologie nie miały wbudowanych równie dobrych zabezpieczeń, a inżynierowie mogli zaprojektować i wdrożyć odpowiednie środki bezpieczeństwa poprzez szereg prób i błędów.

Podobnie do witryn WWW z przeszłości, zawierających luki bezpieczeństwa (zwłaszcza na poziomie serwera i sieci), nowoczesne aplikacje internetowe oferują nową powierzchnię ataku, którą agresorzy aktywnie wykorzystują. Ten krótki zarys historyczny jest istotny, ponieważ podkreśla, że współczesne uwarunkowania zabezpieczeń aplikacji internetowych są tylko jednym etapem cyklicznego procesu. Aplikacje internetowe w przyszłości będą bezpieczniejsze, a hakerzy prawdopodobnie przeniosą się na inną powierzchnię ataku (np. RTC lub gniazda).

Każda nowa technologia ma swoją unikalną powierzchnię ataku i słabe punkty. Jedyną drogą prowadzącą do doskonałości w profesji hakerskiej jest konieczność stałego śledzenia najnowszych technologii, zwykle zawierających luki bezpieczeństwa, które nie zostały jeszcze opublikowane lub znalezione w internecie.

Ta książka prezentuje sposoby włamywania się do nowoczesnych aplikacji internetowych oraz metody ich zabezpieczania. Jednak nowoczesne ofensywne i defensywne techniki zabezpieczeń są tylko jednym z aspektów, jakie poznasz w czasie jej lektury. Ostatecznie możliwość znalezienia własnego rozwiązania problemów bezpieczeństwa jest najcenniejszą umiejętnością specjalisty zajmującego się zabezpieczeniami. Jeśli po przeczytaniu kolejnych rozdziałów zdołasz nauczyć się krytycznego myślenia pod kątem bezpieczeństwa i nabędziesz umiejętności rozwiązywania problemów, lepiej niż inni poradzisz sobie z nowymi lub niestandardowymi atakami, a także z wcześniej nieznanymi mechanizmami zabezpieczeń.

Myśl jak haker — twórz niezawodne zabezpieczenia!

Wydawałoby się, że ze względu na szeroki dostęp do materiałów poświęconych bezpieczeństwu systemów informatycznych temat ten powinien być świetnie znany każdemu inżynierowi. Mimo to media regularnie donoszą o spektakularnych naruszeniach zabezpieczeń. Następstwem udanych ataków mogą być nie tylko straty finansowe i uszczerbek na wizerunku, ale również zagrożenie bezpieczeństwa narodowego.

Książka systematyzuje wiedzę dotyczącą ataków hakerskich i technik zabezpieczania przed nimi aplikacji internetowych. Autor dogłębnie opisuje metody ataków na poziomie kodu i architektury systemu. Sporo uwagi poświęca eksperckim technikom prowadzenia rekonesansów, dzięki którym nawet bez wiedzy o strukturze i kodzie aplikacji można samodzielnie zrozumieć sposób jej działania i zidentyfikować wrażliwe punkty systemu. Następnie omawia różne techniki ataków, począwszy od łamania zwykłych zabezpieczeń, a skończywszy na metodach obchodzenia zaawansowanych mechanizmów obronnych. Kolejne rozdziały dotyczą zapobiegania włamaniom do systemu. Jednym z ciekawszych zagadnień jest ocena kompromisu pomiędzy zapewnieniem akceptowalnego poziomu bezpieczeństwa a kosztami i wydajnością użytkowania aplikacji.

W książce między innymi:

- typowe luki bezpieczeństwa
- podstawowe techniki atakowania aplikacji
- niestandardowe metody omijania typowych zabezpieczeń
- wdrażanie zabezpieczeń aplikacji
- najlepsze praktyki bezpiecznego kodowania w cyklu programistycznym
- poprawa poziomu bezpieczeństwa aplikacji internetowych

Andrew Hoffman jest starszym inżynierem do spraw bezpieczeństwa w Salesforce.com. Specjalizuje się w zabezpieczeniach drzewa DOM i JavaScriptu. Pracował z dostawcami wszystkich najważniejszych przeglądarek oraz z organizacjami TC39 i WHATWG. Bada również zagadnienia „bezstanowych (bezpiecznych/czystych) modułów", umożliwiających wykonywanie kodu JavaScript przy znacznie zmniejszonym ryzyku.

Helion
helion.pl
HELION SA
ul. Kościuszki 1c
44-100 Gliwice
tel.: 32 230 98 63
helion@helion.pl

KOD KORZYŚCI
Sięgnij po więcej!
ISBN 978-83-283-7005-0
9 788328 370050
INFORMATYKA W NAJLEPSZYM WYDANIU
Cena: 59,00 zł