KOD MOBILNY

Informatyka została przekształcona w ciągu ostatnich dwudziestu lat. W ciągu ostatnich dwóch dziesięcioleci dostęp do sieci WWW zmienił się z rzadkiej nowości komputerowej w wszechobecną; obecnie większość ludzi nosi urządzenia mobilne z pełnym bezprzewodowym dostępem do Internetu. Świat online rozkwitł w ciągu ostatnich dwóch dekad. Termin urządzenie mobilne nie jest już ironicznym oksymoronem. Dwadzieścia lat temu dzisiejszy głupi telefon miał rozmiar walizki; dziś znacznie mocniejszy smartfon z łatwością mieści się w dłoni dziecka. Dzisiejsze urządzenia przenośne są znacznie potężniejsze niż poprzednie komputery mainframe. Ich możliwości stanowią wyzwanie w zakresie bezpieczeństwa i zapewnienia informacji. Przyjazny dla konsumenta model łatwych do dodania ulepszeń do tych urządzeń komplikuje ekosystem bezpieczeństwa. Sieć WWW zaczęła się od prostych, statycznych stron. Wkrótce aplikacje zaczęły tworzyć strony na żądanie. Ewolucja technologii związanych z siecią wzrosła do znacznego polegania na JavaScript (znanym również jako ECMAScript) dzięki wykorzystaniu AJAX i innych technik, a także Java i innych wtyczek. Wyzwanie polega na tym, że taki kod jest często dostarczany przez agencje zewnętrzne, gdy jest to potrzebne (np. elementy JavaScript są pobierane od dostawcy lub innego serwera WWW podczas ładowania stron; aplikacje na urządzenia mobilne są instalowane w locie przy niewielkich lub zerowych kosztach nominalnych na życzenie właściciela urządzenia). Tak więc często nie ma lub nie ma możliwości zweryfikowania zachowań takiego kodu, a nawet zachowania pozorów kontroli. Jedna wersja kodu może być dobrze zachowana; następna wersja może umożliwić kompromis urządzenia, niezamierzony lub celowy. Apple App Store używany z iPhone'ami , iPodami i iPadami ma proces zatwierdzania i weryfikacji, ale proces ten nie był całkowicie bezbłędny. Na przykład często zdarza się odwoływać do zewnętrznych elementów JavaScript za pomocą niezaszyfrowanego transferu hipertekstu protokół (HTTP), co powoduje, że strona internetowa jest z natury podatna na maskaradę, man-in-the-middle, podszywanie się pod nazwę domeny i inne ataki. Jeśli nic innego, integralność zewnętrznego elementu zależy od integralności i bezpieczeństwa każdego elementu w sieci, co jest zniechęcającą rzeczywistością. Na najbardziej podstawowym poziomie kod mobilny to zestaw instrukcji dostarczanych do zdalnego urządzenia komputerowego w celu dynamicznego wykonania lub instalacji na urządzeniu. Takie wstępnie zapakowane elementy są dobrodziejstwem dla aplikacji na komputery stacjonarne i urządzenia mobilne; jednak nieodłączna moc tego podejścia wiąże się ze znacznymi zagrożeniami. Zagrożenia związane z kodem mobilnym wynikają z jego nieodłącznej zdolności do wykonywania więcej niż tylko wyświetlania znaków na zdalnym wyświetlaczu. Ta dynamiczna natura kodu mobilnego powoduje trudności w zakresie zasad i wdrażania. Ogólny zakaz kodu mobilnego jest bezpieczny, ale zakaz ten uniemożliwi użytkownikom dynamicznej sieci wykonywanie swoich zadań i odmówi użytkownikom urządzeń mobilnych znacznej części narzędzia urządzenia. Napięcie między integralnością, elastycznością i dynamizmem stanowi sedno problemu. Pojawienie się urządzeń masowego rynku, od urządzeń osobistych (np. smartfonów, tabletów) po gotowe urządzenia (np. serwery plików) sprawia, że powszechne zakazy są często niewykonalne. Takie urządzenia są dostarczane z paczkowanym lub pobieranym oprogramowaniem, które można następnie ulepszyć poprzez pobranie całych nowych wersji lub rozszerzeń zwanych aplikacjami. W obu przypadkach elementy oprogramowania opracowane przez producenta lub strony trzecie są fabrycznie zapakowane, bez możliwości testowania organizacji klienta lub w ogóle. Chociaż konsumeryzacja IT ma zalety, jedną z istotnych wad jest utrata wglądu użytkownika i firmy w to, jakie oprogramowanie jest używane i jaki dostęp ma oprogramowanie do informacji. Inteligentne telefony i tablety są symbolem konsumpcji IT. Bez udoskonaleń dostarczanych przez aplikacje urządzenia te zapewniają cień ich prawdziwych możliwości. Jednak niczego nie podejrzewający użytkownik może pobrać nieuczciwą lub nieuprawnioną aplikację, która zagraża poufnym danym. Aplikacje są dostarczane jako paczkowane, binarne pliki instalacyjne, zwykle dostarczane za pośrednictwem ścieżek komunikacji bezprzewodowej (np. Wi-Fi lub komórkowa sieć komunikacyjna). Kilka definicji, używanych przez siły zbrojne Stanów Zjednoczonych, ale mających zastosowanie do wszystkich, jest użytecznych przy rozważaniu treści tej części:

Enklawa. Środowisko systemu informatycznego, które jest kompleksowo kontrolowane przez jeden organ i ma jednolitą politykę bezpieczeństwa, w tym bezpieczeństwa personelu i bezpieczeństwa fizycznego. Lokalne i zdalne elementy, które uzyskują dostęp do zasobów w enklawie, muszą spełniać zasady polityki enklawy.
Kod mobilny. Oprogramowanie uzyskane ze zdalnych systemów poza granicami enklawy, przesyłane przez sieć, a następnie pobierane i uruchamiane w systemie lokalnym bez wyraźnej instalacji lub wykonania przez odbiorcę. Kod mobilny to potężne narzędzie programowe, które zwiększa możliwości różnych platform, współużytkowania zasobów i rozwiązań internetowych. Jego zastosowanie jest szeroko rozpowszechnione i rośnie zarówno w aplikacjach komercyjnych, jak i rządowych ... Kod mobilny niestety może poważnie pogorszyć… operacje, jeśli jest niewłaściwie używany lub kontrolowany.
Złośliwy kod mobilny. Mobilne moduły oprogramowania kodowego zaprojektowane, zastosowane, dystrybuowane lub aktywowane z zamiarem naruszenia wydajności lub bezpieczeństwa systemów informatycznych, zwiększenia dostępu do tych systemów, zapewnienia nieuprawnionego ujawnienia informacji, uszkodzenia informacji, odmowy usługi lub kradzieży zasobów.

KOD MOBILNY Z SIECI WWW.

W dyskusjach na temat World Wide Web wyrażenie mobilny kod ogólnie odnosi się do kodu wykonywalnego, innego niż HTML i języki pokrewne (np. Extensible Markup Language lub XML), dostarczane drogą elektroniczną przez sieć komunikacyjną (np. Internet, e-mail) w celu wykonania w komputer klienta. Najpopularniejsze technologie pakowania kodu mobilnego to aplikacje, ActiveX, Java i JavaScript. Kod mobilny może bezpośrednio wykonywać ukryte funkcje w systemie klienta, uzyskując dostęp do informacji lub zmieniając działanie lub trwały stan systemu; może tworzyć przypadkowe lub celowe luki w zabezpieczeniach, które można wykorzystać później. Powszechne stosowanie klientów pocztowych, które obsługują wykonywalną treść wiadomości e-mail (np. wiadomości e-mail HTML z osadzonym lub przywoływanym kodem programu), stało się znaczącym źródłem podatności na zagrożenia. Wyskakujące okienka mogą być również źródłem podatności na wiele sposobów; w szczególności mogą uzyskiwać dostęp do innych witryn i stron WWW i mogą powodować zapisy w dziennikach, które mogą powodować problemy prawne. Doświadczenie Julie Amero, Norwich, Connecticut, nauczycielki zastępczej, jest tak samo niepokojące, jak ostrzegawcze. Pani Amero została oskarżona i skazana za używanie komputera w klasie w celu uzyskania dostępu do nieodpowiednich materiałów. Najgorsze zarzuty zostały następnie wycofane, gdy stało się jasne, że komputer w centrum incydentu został zainfekowany złośliwym oprogramowaniem na kilka dni przed incydentem, subskrypcja antywirusowa wygasła, a zeznania o umyślnym dostępie okazały się nieprawdziwe. Jednak Pani Amero była przedmiotem długiego postępowania karnego i piętnowana w prasie. Rozstrzygnięcie sprawy wymagało od Pani Amero przyznania się do winy za wykroczenie, rezygnacji z licencji dydaktycznej i zapłaty grzywny. Chociaż złośliwe oprogramowanie, takie jak wirusy, robaki i konie trojańskie, napisane w językach kompilowanych, interpretowanych lub skryptowych, takich jak Visual Basic, również może być uważane za kod mobilny, szkodniki te nie są ogólnie oznaczone jako takie; ta część dotyczy tylko aplikacji, kontrolek ActiveX, apletów Java i programów JavaScript. Dzisiejsze coraz bardziej dynamiczne i bogate aplikacje internetowe opierają się na AJAX, WebSockets i innych technologiach, które z kolei opierają się na JavaScript i innych technologiach kodu mobilnego. Aplikacje te zwiększają zasięg zagrożenia, utrudniając jednocześnie stosowanie wrażliwych technologii. Chociaż awarie systemów lub aplikacji są najbardziej spektakularnymi problemami z kodem mobilnym, nie są one najbardziej niebezpieczne. Stały, cichy i tajny dostęp lub modyfikacja danych systemu klienta stanowią znacznie poważniejsze zagrożenia. Na przykład niektóre witryny sieci Web potajemnie uzyskiwały adresy e-mail z przeglądarek użytkowników, co skutkowało później niechcianym komercyjnym adresem e-mail. Podobne ataki przeprowadzono na książki telefoniczne i listy kontaktów urządzeń mobilnych. Ponieważ motywacje do takich ataków zostały przeniesione z psikusów do szpiegostwa przestępczego i sponsorowanego przez państwo narodowe, odpowiednio wzrosło znaczenie niewykrywanego tajnego dostępu do danych. W przeszłości taktyka ochrony przed

złośliwym oprogramowaniem polegała w dużej mierze na powszechnej dystrybucji zagrożeń. Oczekiwane pojawienie się designerskiego kodu mobilnego, specjalnie ukierunkowanego na konkretny system lub niewielką liczbę systemów, jest niebezpiecznym trendem. Te ukierunkowane ataki często prześlizgują się przez systemy skanowania sygnatur; rozpowszechniona dystrybucja i reklama nie są częścią ich zamierzonego zastosowania. Oszukańcze ataki pocztą elektroniczną są ogólnie nazywane phishingiem, podobno wywodzącym się z kombinacji phreaking (jak w phishingu telefonicznym, z pominięciem systemów rozliczeniowych o dużej odległości) i połowów. Z pewnym poczuciem humoru ukierunkowane wyłudzanie informacji jest określane jako wyłudzanie informacji; ataki na kierowników wyższego szczebla (np. prezesów) są określane jako wielorybnictwo lub harpunowanie. Nomenklatura może być zabawna, ale dobrze przeprowadzone ataki harpunów mogą być trudne do wykrycia lub zapobiegania. Takie ataki znaleziono w centrum kilku ważnych naruszeń (np. atak typu phishing spear przeciwko RSA w 2011 r.) w dużych korporacjach. Cele o wysokiej wartości to nie tylko kadra kierownicza wyższego szczebla; osoby z dostępem do informacji o wysokiej wartości lub autorytetem są równie zagrożone. Jednym z najwcześniej udokumentowanych odcinków tego typu jest użycie oprogramowania koni trojańskich w 2004 r. w celu rozległej kradzieży danych. Ta sprawa nie była odosobnionym incydentem. Trend ukierunkowanych ataków na starszych pracowników, a nie ataków losowych, przyspieszył. Agencje śledcze również weszły do walki. W lipcu 2007 r. oświadczenie złożone przez Federalne Biuro Śledcze w związku z serią zagrożeń bombowych opisywało wykorzystanie oprogramowania szpiegującego do infiltracji komputera podejrzanego i zwrócenia informacji śledczym. Kod mobilny stanowi złożony zestaw zagadnień dla profesjonalistów z branży informatycznej. Dopuszczenie kodu mobilnego do środowiska operacyjnego zagraża enklawie; jednak nawet komercyjne programy gotowe (COTS) naruszają integralność enklawy. Różnice między kodem mobilnym a innymi formami kodu wynikają głównie ze sposobu, w jaki te zewnętrzne programy są uzyskiwane, instalowane, dokumentowane i kontrolowane. W środowisku komputerowym pozyskiwanie COTS zwykle wymaga świadomej i wyraźnej oceny kosztów i korzyści związanych z instalacją konkretnego programu o nazwie i dokumentacji. Natomiast mobilne programy kodowe są instalowane w dużej mierze bez powiadamiania użytkownika i generalnie bez dokumentacji, kontroli zmian lub przeglądu wszelkiego rodzaju. O ile zapory ogniowe system-klient nie są ustawione tak, aby automatycznie odrzucały kod mobilny (co jest coraz trudniejsze, jeśli nie jest to techniczną i praktyczną niemożliwością), administratorzy systemu nie mogą być pewni, które oprogramowanie zostało uruchomione na komputerach klienckich pod ich nominalną kontrolą. Wykorzystanie technologii opartych na Secure Sockets Layer (SSL), takich jak HTTPS, do zabezpieczania połączeń używanych przez aplikacje w inny sposób, ma również efekt uboczny polegający na zapobieganiu wykrywaniu kodu mobilnego na poziomie zapory ogniowej. Chociaż taka kontrola jest często iluzoryczna, ze względu na obchodzenie przez użytkownika ograniczeń instalacji nieautoryzowanego oprogramowania, użycie kodu mobilnego instalowanego przez zewnętrzne witryny internetowe poważnie zagraża wszelkiej pozostałej kontroli nad konfiguracjami oprogramowania pracowniczego. Kod mobilny był również wykorzystywany w niektórych przypadkach do egzekwowania praw własności do treści, tak jak miało to miejsce w sprawie z 2005 r. dotyczącej Sony Music. Sprawa Sony Music dotyczyła oprogramowania zawartego na muzycznych dyskach CD-ROM; dokładnie ten sam efekt mógł wystąpić w przypadku pobranego pliku lub strony internetowej. Ukryta instalacja wszelkiego rodzaju oprogramowania stanowi poważne zagrożenie dla integralności, bezpieczeństwa i prywatności. Nieprawidłowe działanie lub niewłaściwe użycie takiego oprogramowania prawdopodobnie mieści się w ustawach karnych określających nielegalne, nieautoryzowane modyfikacje systemów. Prokurator generalny Teksasu, a także pozwy prywatne w Nowym Jorku i Kalifornii, złożył skargi przeciwko Sony. Wszystkie te działania zostały rozstrzygnięte przez Sony BMG w grudniu 2006 r. W styczniu 2007 r. Amerykańska Federalna Komisja Handlu ogłosiła ugodę z Sony BMG dotyczącą opłat za instalowanie oprogramowania bez pozwolenia, a także dochodzenia wszczęli również prokuratorzy generalni w Massachusetts i na Florydzie za granicą we Włoszech i Kanadzie. Istnieją również doniesienia, że Sony Root Kit został wykorzystany przez Backdoora. IRC.Snyd.A exploit i inne, aby ukryć pliki przed skanowaniem w poszukiwaniu złośliwego oprogramowania. Powszechny charakter tej indukowanej podatności powinien również dać pauzę. Powszechna luka stanowi niszę ekologiczną gotową do eksploatacji.

MOTYWACJE I CELE

Motywacje i cele twórców szkodliwego oprogramowania kontynuowały ewolucyjny trend, od mimowolnie niszczącego do mściwego, mściwego i kryminalnego. Nic dziwnego, że aktorzy migrowali z nieuczciwych indywidualnych dowcipnisiów do przestępców oraz państw narodowych i pełnomocników. Na początku wiele incydentów było losowo szkodliwych lub psikusami z niezamierzonymi skutkami ubocznymi. Tak już nie jest. Teraz złośliwy kod mobilny jest często kodem przeznaczonym do tego celu. Ten cel może być zawstydzeniem, szantażem, szpiegostwem korporacyjnym lub kradzieżą. W innym wymiarze celem może być podporządkowanie skądinąd niewinnych zasobów komputerowych przedsiębiorstwu przestępczemu przeciwko niepowiązanym stronom trzecim. Zmiana celów ma również dramatyczny wpływ na strategie przeciwdziałania. Kiedy celem była masowa reklama, ta sama infekcja była szeroko rozpowszechniona, a technologie skanowania można było wykorzystać do identyfikacji znanych zagrożeń. Kiedy celem nie jest już rozgłos, rozgłos i powszechna infekcja są nieprzystosowalne. Ukryta infekcja jest wówczas znacznie bardziej atrakcyjną strategią niż dystrybucja masowa. Nie jest prawdopodobne, aby niestandardowy kod mobilny zaprojektowany w celu osiągnięcia selektywnych tajnych infekcji szybko pojawiał się na celowniku oprogramowania do skanowania. Jest to zgodne z trajektorią ewolucji powszechną w świecie biologicznym, w której patogeny z czasem mutują się w mniej śmiertelne formy. Jest nieprzystosowalny do pasożyta, którym jest większość złośliwego oprogramowania, aby śmiertelnie uszkodzić swojego gospodarza. Minusem tego efektu jest to, że przewlekłe infekcje bez widocznych skutków ubocznych są często pomijane. W przyszłości technologie i procedury operacyjne, które utrudniają nieautoryzowanemu kodowi zamieszkiwanie w trwałym stanie systemu lub narażanie go na szwank, są znacznie bardziej pożądanymi strategiami przeciwnymi niż podejścia oparte na skanowaniu w poszukiwaniu znanych infekcji.

BŁĘDY PROJEKTOWE I WYKONAWCZE

Błędy projektowe i implementacyjne przyjmują różne formy. Najprostsze przypadki dotyczą oprogramowania, które działa w sposób ciągły i przewidywalny. Bardziej szkodliwe i bardziej niebezpieczne są te błędy, które dyskretnie zagrażają ścisłemu ograniczeniu środowiska wielu użytkowników. Błędy w takich warstwach profilaktycznych, znane jako ceglane ściany lub piaskownice, zagrażają integralności schematu ochrony. W najgorszych przypadkach umożliwiają nieograniczony dostęp do zasobów na poziomie systemu przez nieuprzywilejowane programy użytkownika. Błędy projektowe i implementacyjne mogą wystąpić w dowolnym programie lub procedurze, a kod mobilny nie jest wyjątkiem. Piaskownice (nieuprzywilejowane, ograniczone środowiska operacyjne) mają na celu zapobieganie nieautoryzowanym operacjom. Uwierzytelnianie określa, która organizacja bierze odpowiedzialność za takie błędy. W tej części omówiono model bezpieczeństwa oparty na uwierzytelnianiu kodu mobilnego, a następnie zbadano, w jaki sposób ograniczone środowiska operacyjne pomagają ograniczyć szkody spowodowane szkodliwym kodem mobilnym. Obawy te dotyczą zarówno szeroko rozpowszechnionych, jak i ukierunkowanych ataków. Wyzwanie ataków ukierunkowanych leży w ich małej populacji; ukierunkowane ataki raczej nie pojawią się na radarze ogólnych programów skanujących dystrybucję.

PODPISANY KOD.

Technologie uwierzytelniania mają na celu zapewnienie, że informacje dostarczone przez organizację nie zostaną zmienione bez autoryzacji. Technologia zastosowana do wdrożenia tego zapewnienia opiera się na wykorzystaniu infrastruktury klucza publicznego (PKI). Kod uwierzytelniany przy użyciu mechanizmów opartych na PKI często jest nazywany podpisem. Podpisany kod jest zasadniczo odporny na nieautoryzowane uwierzytelnianie; jednak podpis gwarantuje integralność tylko od momentu podpisania kodu; proces podpisywania nie oznacza bezpieczeństwa ani jakości kodu przed momentem podpisania. Po podpisaniu pliku nie można zmienić bez unieważnienia oryginalnego podpisu; bez

współpracy osoby mającej dostęp do klucza prywatnego powiązanego z certyfikatem organizacji X.509 organizacji, podpisu nie można realistycznie zmodyfikować, aby wyglądał na zgodny z prawem. (Certyfikat X.509, podpisany cyfrowo przez autoryzowanego użytkownika, uwierzytelnia powiązanie między nazwą użytkownika a kluczem publicznym użytkownika.) Jednak patrząc pod powierzchnią, takie środki ostrożności nie uwzględniają różnych luk w zabezpieczeniach:

* Nieautoryzowany dostęp do kluczy prywatnych (podpisywanie)
* Dostęp do bazy kodu przed podpisaniem
* Fałszywe certyfikaty
* Błędy projektowe i implementacyjne

AUTHETICODE

Technologia Authenticode firmy Microsoft jest przykładem podejścia opartego na uwierzytelnianiu. Programiści, którzy chcą rozpowszechniać kod, otrzymują odpowiedni certyfikat cyfrowy od urzędu certyfikacji i używają certyfikatu cyfrowego do podpisania kodu. Podpis jest następnie sprawdzany przez system klienta za każdym razem, gdy kod jest wykonywany. Authenticode opiera się na kilku komponentach:

* Certyfikaty PKI i X.509 wydane przez urząd certyfikacji.
* Ograniczony dostęp do kluczy prywatnych związanych z certyfikatem X.509 organizacji wydającej. W terminologii Microsoft termin Software Publishing Certificate lub SPC odnosi się do obiektu PKCS # 7, który z kolei zawiera zbiór certyfikatów X.509 używanych do podpisywania kodu.
* Integralność procesów używanych przez urząd certyfikacji w celu zapewnienia, że żądania certyfikatów X.509 są prawidłowe.

Authenticode nie rozwiązuje problemów związanych z bezpieczeństwem lub dokładnością podpisanego kodu, a jedynie to, że jest autentyczny i niezmieniony od momentu podpisania. Na przykład podpisywanie nie chroni przed przypadkami złego traktowania pracowników.

PODSTAWOWE OGRANICZENIA PODPISANEGO KODU

Technologie podpisywania, niezależnie od kontekstu (np. E-mail, aplety i archiwa), nie odpowiadają bezpośrednio na pytania dotyczące dokładności lub poprawności; odnoszą się jedynie do kwestii pochodzenia. Największym niebezpieczeństwem związanym z podpisywaniem programów jest podejście polegające na zaufaniu „wszystko albo nic". Podpisane elementy uznaje się za godne zaufania w pełnym zakresie uprawnień użytkownika proszącego. Podpisany element może wykonać dowolną operację, którą użytkownik może wykonać. Nie ma pojęcia częściowego zaufania. Według słów pełnomocnika taka akceptacja byłaby ogólnym pełnomocnictwem. Zgodnie ze słowami bezpieczeństwa sponsorowanego przez CERT / Centrum Koordynacji (CERT / CC) w warsztacie ActiveX: „Podpis cyfrowy nie daje jednak żadnej gwarancji dobroci ani kompetencji." Jednocześnie wrodzona moc i widoczna legalność podpisu cyfrowego stanowi duże obciążenie dla sygnatariuszy i wyższych poziomów infrastruktury PKI, aby zapewnić integralność mechanizmów i tajemnic. Kluczem do integralności podpisanego kodu jest proces podpisywania i proces generujący obiekt do podpisania; bezpieczeństwo tajnych kluczy wymaganych do jego wdrożenia określa stopień zaufania w zakresie przypisywania podpisanego kodu. W najprawdziwszym sensie klucze prywatne powiązane z certyfikatem X.509 reprezentują klucze do królestwa, tak cenne jak podpis na Dalekim Wschodzie lub faksowa tablica podpisu na konto bankowe. Na poziomie praktycznym akceptacja kodu podpisanego przez organizację jest wyraźną akceptacją tego, że organizacja podpisująca ma dobrą kontrolę nad użyciem swoich kluczy podpisu. Organizacje, które poważnie podchodzą do kwestii bezpieczeństwa, segregują dostęp do uprzywilejowanych kont i kontrolują dostęp do systemów, są dobrze przygotowane do zarządzania procedurami podpisywania kodu. W związku z tym procedury i systemy

stosowane do podpisywania kodu należy traktować z taką samą ostrożnością, jak w przypadku wyżej wymienionych tablic podpisujących lub urządzeń kryptograficznych o maksymalnym bezpieczeństwie znanych osobom z obszaru bezpieczeństwa narodowego. Niestety, pomimo wielu lat rozgłosu na temat zagrożeń związanych ze wspólnymi hasłami i kontami, wiele instalacji IT nadal korzysta ze wspólnych kont i haseł. Nie ma powodu, by zakładać, że tajemnice związane z PKI są lepiej chronione, pomimo obszernych zaleceń, aby te szczegóły były dobrze strzeżone.

SPECYFICZNE PROBLEMY Z MODELEM ZABEZPIECZEŃ ACTIVE X

Warsztaty CERT / CC na temat bezpieczeństwa w ActiveX podsumowały kwestie bezpieczeństwa w trzech głównych obszarach: importowanie i instalowanie kontrolek, uruchamianie kontrolek oraz korzystanie z kontroli przez skrypty. Kluczowe wnioski z tego raportu są następujące:
* Importowanie i instalowanie elementów sterujących
-Jak omówiono, jedyną podstawą zaufania do podpisanej kontroli jest jej przypuszczalne pochodzenie. Jednak twórca kodu mógł mieć wadę projektową w kontroli lub może nie zapewnić odpowiedniej jakości oprogramowania, aby zapobiec poważnym błędom.
- Ufający użytkownik może zainstalować podpisaną kontrolkę, która zawiera lukę, dzięki czemu jest przydatna dla atakujących tylko dlatego, że jest podpisana.
- W systemach OnWindows z wieloma użytkownikami, gdy kontrola zostanie dozwolona przez jednego użytkownika, pozostaje dostępna dla wszystkich użytkowników, nawet jeśli ich poziomy bezpieczeństwa są różne.
* Kontrola działania
- Formant ActiveX nie ma ograniczeń co do tego, co może zrobić na komputerze klienckim i działa z tymi samymi uprawnieniami, co uprawnienia użytkownika, który zainicjował kontrolę.
- Chociaż środki bezpieczeństwa ActiveX są dostępne w Internet Explorerze, inne oprogramowanie klienckie może uruchamiać kontrole bez koniecznej implementacji takich zabezpieczeń. Poziomy zabezpieczeń przeglądarki Internet Explorer są zwykle zerowe lub całkowite, co utrudnia dopuszczenie określonej kontroli bez dopuszczenia wszystkich kontroli tego typu. Zdalna aktywacja kontroli może ominąć normalne obwody bezpieczeństwa, takie jak narzucone przez zapory ogniowe.
- Nie ma podstaw do podjęcia decyzji, czy dana kontrola jest bezpieczna do wykonania, czy nie, w jakimkolwiek określonym kontekście.
* Obawy związane ze skryptami
- Brak ogólnej podstawy do ograniczenia działań kontrolnych, programiści ActiveX muszą skutecznie określić własne środki ostrożności, aby zapobiec szkodliwym działaniom. Wystarczająco trudno jest opracować dobry zestaw granic działania programu, nawet jeśli używa się ogólnego modelu, takiego jak piaskownica opisana później; niezwykle trudno jest zobaczyć, w jaki sposób można oczekiwać od poszczególnych programistów lub, co ważniejsze, zaufać, że utworzą własny odpowiednik piaskownicy dla każdej kontroli. W świetle tych zagrożeń autorzy raportu CERT / CC stwierdzili, że „istnieje duża liczba potencjalnych punktów awarii".

STUDIUM PRZYPADKU

Od czasu wprowadzenia tej technologii w połowie lat 90. miało miejsce kilka naruszeń bezpieczeństwa lub demonstracje, w których pośredniczy ActiveX.
Internet Exploder.
W 1996 roku Fred McLain napisał Internet Exploder, kontrolkę ActiveX zaprojektowaną w celu zilustrowania szerokiego stopnia zaufania do kontroli ActiveX z racji jej podpisania. Exploder, po pobraniu do wykonania przez Internet Explorera, zamknie komputer przeglądarki (odpowiednik sekwencji Shutdown | Shutdown z menu Start w systemie Windows). Ta operacja jest zakłócająca działanie, ale tak naprawdę nie uszkadza systemu. McLain zauważa w swoich często zadawanych pytaniach na temat programu Exploder, że łatwo jest tworzyć destrukcyjne lub złośliwe elementy sterujące. Exploder podnosi ważne pytanie: kto i jakie są ograniczenia zaufania przy użyciu

podpisanego kodu? W normalnych sprawach handlowych istnieje duża różnica między nieautentycznym podpisem, fałszerstwem a prawidłowo podpisanym, ale nieopłacalnym czekiem. W oprogramowaniu różnica między nieautentyczną kontrolą a niebezpieczną jest znacznie mniej wyraźna.

DEMONSTRACJA CHAOS COMPUTER CLUB

27 stycznia 1997 r. Niemiecki program telewizyjny pokazał demonstrację członków Chaos Computer Club ,w jaki sposób mogliby użyć formantu ActiveX do kradzieży pieniędzy z konta bankowego. Kontrolka, dostępna w Internecie, została napisana w celu obalenia popularnego pakietu księgowego Quicken za pomocą elektronicznej wersji tailgating. Ofiara musi jedynie odwiedzić witrynę i pobrać kontrolkę ActiveX, o której mowa; formant następnie automatycznie sprawdza, czy zainstalowano Quicken. Jeśli tak, kontrola nakazała Quicken wystawić zlecenie przeniesienia, które zostanie zapisane na liście oczekujących przelewów. Następnym razem, gdy ofiara połączy się z odpowiednim bankiem i wyśle do banku wszystkie oczekujące polecenia przelewu, wszystkie przelewy zostaną wykonane jako jedna transakcja. Osobisty numer identyfikacyjny użytkownika (PIN) i numer autoryzacji transakcji (TAN) będą miały zastosowanie do wszystkich przelewów, w tym oszukańczych w stosie zamówień. Większość ofiar byłaby nieświadoma kradzieży, dopóki nie otrzyma kolejnego wydruku - jeśli wtedy. Dan Wallach z Princeton University, komentując tę sprawę, napisał:
Gdy akceptujesz kontrolkę ActiveX, zezwalasz na to, aby całkowicie dowolny kod szperał w twoim komputerze i robił wszystko, co mu się podoba. Ten sam kod może wykonywać niezwykle kosztowne rozmowy telefoniczne na numery 900 lub na duże odległości za pomocą modemu; potrafi czytać, zapisywać i usuwać dowolny plik na twoim komputerze; potrafi instalować konie trojańskie i wirusy. Wszystko bez podstępu i hakerów wymaganych do zrobienia tego w Javie. ActiveX przekazuje klucze do komputera. Odpowiadając na krytykę modelu bezpieczeństwa ActiveX, Bob Atkinson, architekt i główny implementator Authenticode, napisał długi esej wyjaśniający jego punkt widzenia. Wśród kluczowych punktów:
* Microsoft nigdy nie twierdził, że poświadcza bezpieczeństwo kodu innych osób.
* Uwierzytelnianie ma na celu wyłącznie identyfikację sprawców po wykryciu złośliwego kodu.
* Dystrybucja oprogramowania oparta na eksploratorze nie jest bardziej ryzykowna niż konwencjonalne zakupy przez sprzedawców oprogramowania.
Późniejsza korespondencja na forum RISKS karała pana Atkinsona za pominięcie kilku innych kluczowych punktów, takich jak:
* Interakcje między kontrolkami ActiveX mogą naruszać bezpieczeństwo systemu, nawet jeśli poszczególne kontrolki wydają się nieszkodliwe.
* W rzeczywistości nie ma precedensu w zakresie kładzenia odpowiedzialności u twórców oprogramowania, nawet jeśli można je znaleźć.
* Pod atakiem dowód podpisu cyfrowego prawdopodobnie wyparuje z uszkodzonego systemu.
* Opóźnienie wykonania szkodliwych ładunków skomplikuje identyfikację źródła uszkodzenia.
* Złośliwość nie jest tak ważnym zagrożeniem ze strony kodu jak niekompetencja.
* Firma Microsoft ma w historii opcje zagrażające bezpieczeństwu, takie jak automatyczne wykonywanie makr w programie Word, bez oferowania jakiegokolwiek sposobu wyłączenia tej funkcji.
* Witryna AWeb może wywoływać formant ActiveX, który znajduje się w innej witrynie lub który został już pobrany z innej witryny, i może przekazywać za pomocą tej kontroli nieoczekiwane argumenty, które mogą wyrządzić szkodę.
Krytyka Wallacha dotycząca ActiveX ma ogólne zastosowanie do technologii podpisanego kodu, a nie jest specyficzna dla ActiveX Microsoftu.

CERTYFIKATY UZYSKANE PRZEZ OSZUSTÓW

W styczniu 2001 r. Firma VeriSign wydała dwa certyfikaty cyfrowe klasy 3 do podpisywania formantów ActiveX i innego kodu osobom podszywającym się pod pracownika Microsoft. W rezultacie

użytkownicy otrzymujący kod podpisany przy użyciu tych certyfikatów otrzymaliby prośbę o zaakceptowanie lub odrzucenie certyfikatu najwyraźniej podpisanego przez Microsoft 30 lub 31 stycznia 2001 r. Gdy Russ Cooper skomentował na grupie Usenet NTBUGTRAQ, gdy wiadomości pojawiły się w marcu 2001:
Fakt, że jeśli nie sprawdzisz daty na Certyfikacie, nie będziesz wiedział, czy jest to [sic], któremu możesz zaufać, jest Złą Rzeczą (tm) [sic], ponieważ oczywiście nie wszyscy (czytaj: obok nikogo) sprawdzają każdy otrzymany certyfikat. Zastanawiasz się, jak mechanizm wydawania VeriSign może być tak źle zaprojektowany i / lub wdrożony, aby pozwolić na coś takiego.
Tymczasem Microsoft [sic] pracuje nad łatką, która wbije palec w tamę. Zasadniczo certyfikaty VeriSign do podpisywania kodu nie wykorzystują funkcji listy odwołania certyfikatów (CRL) o nazwie CDP lub punktu dystrybucji CRL, co powoduje, że certyfikat jest sprawdzany pod kątem odwołania za każdym razem, gdy jest czytany. Nawet jeśli masz włączoną listę CRL w IE, certyfikaty VeriSign do podpisywania kodu nie są sprawdzane. Aktualizacja Microsoftu będzie migać w pewnym mechanizmie, który powoduje, że niektóre / wszystkie certyfikaty podpisujące kod sprawdzają lokalny plik / klucz rejestru pod kątem listy CRL, która (przynajmniej początkowo) będzie zawierać szczegóły tych certyfikatów. Zakładając, że działa to zgodnie z reklamą, każda próba zaufania do źle wydanych certyfikatów powinna zakończyć się niepowodzeniem.

Roger Thompson, dyrektor ds. technicznych w Laboratoriach zapobiegania wykorzystywaniu luk, wyjaśnił, że motywy oszustów określają, jak złe byłyby wyniki fałszywych certyfikatów. „Jeśli był to ktoś, kto miał jakiś cel, to sześć tygodni to dużo czasu, aby coś zrobić" - powiedział. „Jeśli zadaniem było zainstalowanie sniffera, to mogło być w rezultacie zillionem backdoorów. "Opublikowane raporty wskazują, że niepowodzenie uwierzytelnienia nastąpiło z powodu błędu w procesie wydawania w VeriSign: Certyfikaty zostały wydane przed otrzymaniem wiadomości e-mail z potwierdzeniem, że oficjalny kontakt z klientem autoryzował certyfikaty. Ta sprawa była pierwszą wykrytą awarią uwierzytelnienia w ponad 500 000 certyfikatów wydanych przez VeriSign. Niektóre ostatnie przypadki dotyczące skradzionych lub sfałszowanych certyfikatów obejmują:
* W listopadzie 2011 r. sprzedawca oprogramowania antymalware Mikko Hypponen, firma F-Secure, poinformował, że „znalazł kawałek złośliwego oprogramowania, które zostało kryptograficznie podpisane fałszywym certyfikatem Adobe pochodzącym z rządu Malezji: Malezyjski Instytut Badań i Rozwoju Rolnictwa…"
* We wrześniu 2012 r. Adobe poinformowało, że „… otrzymały dwa złośliwe narzędzia, które wyglądały na podpisane cyfrowo przy użyciu ważnego certyfikatu do podpisywania kodu Adobe". Firma wyjaśniła: „Wyrafinowani aktorzy zagrożeń używają złośliwych narzędzi, takich jak podpisane próbki, podczas wysoce ukierunkowanych ataków uprzywilejowanie eskalacji i bocznego przemieszczania się w środowisku po początkowym naruszeniu bezpieczeństwa komputera. "
* W lutym 2013 r. Bit9 Systems ujawnił, że „Ze względu na nadzór operacyjny nad Bit9 nie udało nam się zainstalować własnego produktu na kilku komputerach w naszej sieci. W rezultacie złośliwa strona trzecia mogła nielegalnie uzyskać tymczasowy dostęp do jednego z naszych certyfikatów do cyfrowego podpisywania kodu, którego następnie użyła do nielegalnego podpisania złośliwego oprogramowania. Nic nie wskazuje na to, że był to problem z naszym produktem. Nasze dochodzenie pokazuje również, że nasz produkt nie został naruszony. "

FAŁSZOWANIE CERTYFIKATÓW

Fałszowanie certyfikatów to kolejne zagrożenie. Certyfikaty nie są magiczne; są jedynie udokumentowane kryptograficznie I podpisane przez uznany urząd certyfikacji. Techniki kryptograficzne stosowane do podpisywania certyfikatów podlegają atakowi. Wyniki badań wskazujące na ryzyko kolizji MD-5 prowadzących do sfałszowanych certyfikatów zostały pierwotnie opublikowane w 2008 r. Z tego powodu wiele organów zaleciło wycofanie certyfikatów podpisanych MD-5. W sierpniu 2011 r. Irańscy internauci zgłosili fałszywy certyfikat podpisany przez MD-5 dla * .google.com

OGRANICZONE ŚRODOWISKA OPERACYJNE.

Z perspektywy sIECI termin sandbox [piaskownica] definiuje coś, co można by nazwać ograniczonym środowiskiem operacyjnym. Ograniczone środowiska operacyjne nie są nowe; istnieją od prawie 50 lat w postaci systemów operacyjnych dla wielu użytkowników, w tym MULTICS, OS / 360 i jego potomków, OpenVMS, UNIX i innych. Mówiąc prościej, ograniczone lub nieuprzywilejowane środowisko operacyjne zabrania normalnym użytkownikom i ich programom wykonywania operacji, które mogą zagrozić całemu systemowi. W takim środowisku zwykłym użytkownikom nie wolno wykonywać takich operacji, jak HALT, które mają bezpośredni wpływ na sprzęt. Programy użytkownika nie mogą wykonywać instrukcji, które mogą zagrozić alokacji pamięci systemu operacyjnego i stanowi procesora, ani uzyskiwać dostępu lub modyfikować pliki należące do systemu operacyjnego lub innych użytkowników. Takie systemy, starannie wdrożone i zarządzane, są bardzo skuteczne w ochronie informacji i danych przed nieautoryzowanymi modyfikacjami i dostępem. Orange Book National Computer Security Center (NCSC) zawiera kryteria klasyfikacji i oceny zaufanych systemów. Mocne i słabe strony chronionych systemów są dobrze znane. Zezwolenie zwykłym użytkownikom na nieograniczony dostęp do plików systemowych zagraża integralności systemu. Użytkownicy uprzywilejowani (tj. Posiadający legalny dostęp do plików systemowych i sprzętu fizycznego) muszą uważać, aby uruchamiane przez nich programy nie naruszały systemu operacyjnego. Większość chronionych systemów zawiera zbiór programów wolnostojących, które implementują przydatne funkcje systemowe wymagające jakiejś formy uprawnień do działania. Często programy te były źródłem luk w zabezpieczeniach. Jest to podstawowe uzasadnienie uniwersalnej rekomendacji, że programy nie działają jako root lub Administrator lub z podobnymi uprawnieniami, chyba że jest to absolutnie konieczne.

JAVA

Java to język opracowany przez Sun Microsystems do niezależnego od platformy wykonywania kodu, zwykle w kontekście przeglądarki internetowej. Podstawowe środowisko Java obejmuje maszynę wirtualną Java (JVM) i zestaw oprogramowania pomocniczego zwanego środowiskiem Java Run Time. Aplety pobrane przez Internet (intranet lub Internet) mają ścisłe ograniczenia w dostępie do zasobów systemowych. W szczególności ograniczenia te uniemożliwiają wykonywanie zewnętrznych poleceń oraz dostęp do odczytu lub zapisu do plików. Środowisko Java zapewnia podpisane aplety, które mają szerszy dostęp do plików. Dynamicznie pobierane aplety są również ograniczone do inicjowania połączeń z systemem, który je dostarczył, teoretycznie ograniczając niektóre rodzaje ataków stron trzecich. W koncepcji podejście Java, które obejmuje również inne testy poprawności pseudokodu JVM, powinno być odpowiednie do zapewnienia bezpieczeństwa. Jednak zbiór zaufanych apletów znalezionych lokalnie w systemie klienckim i podpisane pobrane aplety reprezentują sposoby na obalenie systemu bezpieczeństwa. Bez podpisu podejście Java jest również podatne na ataki polegające na fałszowaniu przez system nazw domen (DNS). Schematy ochrony wielu użytkowników i ochrony maszyn wirtualnych są również całkowicie zależne od integralności kodu, który oddziela nieuprzywilejowanych użytkowników od uprzywilejowanych, zagrażających systemowi operacji. Java nie była wyjątkiem od tej reguły. W 1996 r. błąd w środowisku Java zawartym w Netscape Navigator wersja 2.0 umożliwiał połączenia z dowolnymi uniwersalnymi lokalizatorami zasobów (URL). Później, w 2000 r., wykryto błędy w kodzie, które chroniły różne zasoby. Chociaż środowisko Java jest mniej podatne na wykorzystywanie niż ActiveX, nadal ujawniane są luki w zabezpieczeniach. W 2007 r. US-CERT zgłosił co najmniej dwie luki w zabezpieczeniach. Co istotne, obie z tych zgłoszonych luk dotyczyły zdolności niezaufanych apletów do naruszenia koperty bezpieczeństwa. Ponadto, ponieważ niepodpisany kod może wykorzystywać błędy w podpisanym kodzie źródłowym, nie ma gwarancji, że złożone kombinacje niezaufanego i zaufanego kodu nie doprowadzą do kompromisów w zakresie bezpieczeństwa.

MASZYNY WIRTUALNE

Izolację można również realizować za pomocą maszyn wirtualnych. Teoretycznie każda maszyna wirtualna uruchamia własną kopię systemu operacyjnego na osobnej wirtualnej kopii sprzętu. Właściwie zaimplementowane systemy mainframe od dziesięcioleci obsługują wiele niezależnych maszyn w jednym systemie fizycznym. Technologia maszyn wirtualnych jest podstawową technologią dla tak zwanych chmur obliczeniowych. W takich konfiguracjach pięta achillesowa to warstwa oddzielająca różne maszyny wirtualne. Jeśli w tej warstwie lub jej interfejsach występuje niedobór, podział na przedziały może być zagrożony. W 2012 roku taki incydent miał miejsce w przypadku wielu implementacji maszyn wirtualnych implementujących maszyny wirtualne na procesorach Intel x86. W takich przypadkach specjalnie spreparowana rama stosu może doprowadzić do zerwania mechanizmów przechowujących maszynę wirtualną. Atak ten dotyczył maszyn wirtualnych z Xen, FreeBSD, Red Hat i Microsoft Windows.

PODSUMOWANIE ŚRODOWISKA OPERACYJNEGO

W tym kontekście istotne są szczegóły wirtualizacji. Błędy we wdrażaniu oprogramowania do wirtualizacji mogą powodować naruszenia bezpieczeństwa w oddzielnych, różnych maszyn wirtualnych, a nawet zagrażać integralności hosta. Takie zagrożenia nie są teoretyczne; występują w obszarach izolowanych na poziomie aplikacji i na maszynach wirtualnych na poziomie systemu.

DYSKUSJA

Bezpieczeństwo kodu mobilnego wiąże się z ważnymi problemami związanymi z obsługą relacji w coraz bardziej połączonym środowisku komputerowym.

Zaufanie asymetryczne i przechodnie lub pochodne.

To jest powszechne aby stosunki cybernetyczne były asymetryczne w odniesieniu do wielkości lub siły stron. Ten fakt zwiększa prawdopodobieństwo katastroficznych interakcji. Stwarza to również możliwości masowej infekcji ponad granicami organizacji. Duże lub krytyczne organizacje często mogą jednostronnie nakładać ograniczenia na zdolność organizacji partnerskich do obrony infrastruktury informacyjnej przed szkodami. Połączenie potężnej organizacji i niewystarczającej kontroli uprawnień do podpisywania lub alternatywnie obowiązkowego wykonywania niepodpisanych (lub samopodpisanych) kontrolek ActiveX jest receptą na poważne problemy. Potężna organizacja jest w stanie zobowiązać swoich partnerów do zaakceptowania niskiego poziomu bezpieczeństwa, na przykład wynikającego z użycia niepodpisanych kontrolek ActiveX, jednocześnie rezygnując z odpowiedzialności za wynikłe z tego skutki i reperkusje. Wszystkie organizacje powinny, ze względów bezpieczeństwa i wydajności, korzystać z technologii wymagającej najmniejszego uprzywilejowania, aby osiągnąć pożądany rezultat. JavaScript / ECMAScript może zapewniać wiele funkcji, bez potrzeby korzystania z funkcjonalności zapewnianej przez Javę, a tym bardziej ActiveX. Pozostaje wspólny dla dużych organizacji aby wymusić pobieranie formantów ActiveX do celów, które nie wymagają mocy ActiveX, wykorzystując jedynie uzasadnienie, że postrzegają Internet Explorer jako bardziej rozpowszechnioną przeglądarkę. Często wymaga to uruchomienia skryptu instalacyjnego z konta z uprawnieniami administratora, co stanowi drugie naruszenie bezpieczeństwa. Jest to szczególnie zaskakujące, ponieważ te same organizacje często oferują równoległą obsługę przeglądarek Firefox, Opera, Safari i innych przeglądarek nieobsługujących formantów ActiveX w systemach Linux, Apple i innych platformach. Ta koncepcja Trust Me wymusza ryzyko i ciężar konsekwencji dla użytkownika końcowego, który jest znacznie mniej zdolny do poradzenia sobie z konsekwencjami. Jak zauważono wcześniej w tym rozdziale, serwery WWW stanowią atrakcyjny wektor ataków. Metody podpisywania (uwierzytelniania) są sposobem kontrolowania potencjalnego uszkodzenia, jeśli mechanizmy stosowane do dopuszczania kodu wykonywalnego są odpowiednio kontrolowane. Brak kontroli tych mechanizmów prowadzi do poważnych skutków ubocznych. W rozdziale 30 niniejszego podręcznika zauważono, że ochrona serwerów sieciowych wymaga ostrożnego zarządzania zawartością serwerów. Właściwe i często konieczne jest izolowanie serwerów sieciowych w oddzielnych segmentach sieci, oddzielonych od Internetu i intranetu organizacyjnego zaporami ogniowymi. Te środki ostrożności są

jeszcze bardziej konieczne, gdy serwery są odpowiedzialne za dostarczanie kodu wykonywalnego klientom. Specjaliści ds. Bezpieczeństwa powinni dokładnie zbadać różne funkcje wykonywane przez każdy serwer. W niektórych przypadkach, takich jak hosty OpenVMS, gdzie serwery sieciowe zwykle działają jako nieuprzywilejowane procesy w osobnych kontekstach i drzewach katalogów, możliwe jest uruchomienie wielu usług na jednym serwerze. W innych systemach, takich jak UNIX i Windows, w których usługi aplikacji są wykonywane jako uprzywilejowane, z pełnym dostępem do wszystkich plików systemowych, błąd logiczny w usłudze sieciowej może zagrozić bezpieczeństwu całego serwera, w tym kolekcji plików do pobrania aplety. O wiele bardziej poważne i równie subtelne jest zaufanie przechodnie (lub pochodne): Alpha ufa Beta, która ufa Gamma. Bezpieczeństwo kompromisowe - na przykład niepodpisany aplet Java lub źle działająca lub złośliwa kontrola ActiveX dostarczana przez Beta kompromitującą gamma. Beta następnie powoduje problemy z systemami Alpha. Ta kaskada może być powtarzana wielokrotnie, co prowadzi do wielu zagrożonych usług i systemów sieciowych, które są dalekie geograficznie i organizacyjnie od pierwotnego incydentu

SPRZENIEWIERZENIE I WYWRÓCENIE

Przestrzeń zagrożenia mutowała w ciągu ostatnich kilku lat. Tam, gdzie głównym zagrożeniem ze strony kodu mobilnego były ataki na docelową maszynę, dzisiejsze zagrożenie jest znacznie bardziej zróżnicowane. W listopadzie 2007 roku John Schiefer z Los Angeles przyznał się do winy za zainstalowanie oprogramowania zaprojektowanego do przechwytywania nazw użytkowników i haseł. Według doniesień prasowych był także zaangażowany w prowadzenie wielu sieci zainfekowanych komputerów, często nazywanych botami, które są często wykorzystywane do inicjowania rozproszonej odmowy usługi (DDoS) i innych ataków. W tym konkretnym przypadku w ogłoszeniu Departamentu Sprawiedliwości USA wspomniano o dwóch konkretnych epizodach: 250 000 komputerów zainfekowanych robotami szpiegującymi w celu uzyskania nazw użytkowników i haseł do systemu PayPal i innych systemów; oraz odrębny program obejmujący holenderską firmę reklamową w Internecie, w której sieć 150 000 zainfekowanych komputerów została wykorzystana do zarejestrowania się w jednym z programów firmy reklamowej. Był to jeden z przypadków wynikających z Bot Roast II 47, operacji FBI przeciwko kilku sieciom botnetowym.

ZAGROŻENIA WIELOWYMIAROWE

Kod mobilny jest zagrożeniem wielowymiarowym, z kilkoma różnymi aspektami, z których każdy musi być traktowany osobno. Kod podpisujący, taki jak aplety Java lub formanty ActiveX, rozwiązuje problem autentyczności i uprawnień do wydania kodu. Jednak integralność mechanizmu podpisu wymaga, aby integralność infrastruktury PKI była bez zarzutu. W bardzo realnym sensie infrastruktura PKI jest poza kontrolą samej organizacji. Jakikolwiek kompromis lub poświadczenie proceduralne ze strony urzędu certyfikacji lub osoby podpisującej unieważnia domniemania bezpieczeństwa. Podpisywanie, choć w znacznym stopniu przyczynia się do rozwiązania kwestii autentyczności, nie dotyczy bezpieczeństwa ani ważności. Na przykład formant ActiveX usługi Windows Update, dystrybuowany przez firmę Microsoft jako część różnych systemów operacyjnych Windows, ma za swój podstawowy cel aktualizację systemu operacyjnego. Awaria tej kontroli byłaby katastrofalna. Na szczęście Microsoft daje użytkownikom możliwość korzystania z funkcji automatycznej aktualizacji lub ręcznego wykonywania aktualizacji. Wiele aplikacji internetowych nie jest tak dostosowanych. Problem nie polega wyłącznie na nieprawidłowym działaniu apletów. Możliwe jest, że zbiór apletów zaangażowanych w ogólną działalność biznesową klienta może kolidować w nieoczekiwany sposób, od próby użycia tego samego klucza rejestru Windows w sprzeczny sposób, po przypadkowe użycie tej samej tymczasowej nazwy pliku. Podobne problemy często występują w przypadku aplikacji, które zakładają, że mają monopol na korzystanie z systemu, co jest zbyt powszechnym syndromem. Te kwestie są w większości całkowicie niezwiązane ze sobą. Rozwiązanie w jednym obszarze nie poprawi ani nie pogorszy sytuacji w odniesieniu do innych kwestii.

OBOWIĄZKI KLIENTA

Rosnące zagrożenie stanowi wyzwanie dla osób odpowiedzialnych za zapewnienie integralności komputerów stacjonarnych. Mówiąc wprost, istnieje złożone, wielowymiarowe zagrożenie i nie można go łatwo obronić przed użyciem technik portali, zapór ogniowych i skanerów. Niebezpieczeństwo związane z przeglądaniem sieci WWW to niebezpieczeństwo, że przeglądarka zezwoli atakującemu, bezpośrednio lub pośrednio, na spowodowanie zmiany trwałego stanu systemu. Najprostszym krokiem we właściwym kierunku jest nie przeglądanie sieci WWW w kontekście ochrony, który ma dostęp do krytycznych plików systemowych i ustawień. Ograniczenie tego dostępu za pomocą nieuprzywilejowanego konta użytkownika do przeglądania znacznie zmniejsza ryzyko, pod warunkiem, że pliki systemowe są chronione przed dostępem przez takie konto. Masowa dostępność technologii maszyn wirtualnych stanowi dodatkową alternatywę. Technologia maszyn wirtualnych, której pionierem była IBM w latach 60. XX wieku na komputerach mainframe, pojawiła się w nowej formie na platformach aż do poziomu komputerów stacjonarnych. Ogólna dostępność tej możliwości w świecie komputerów stacjonarnych otwiera zupełnie nową strategię obrony przed złośliwym oprogramowaniem mobilnym: niepotrzebną przeglądarką internetową. Niezbędna przeglądarka WWW to utworzony instancja w środowisku maszyny wirtualnej ze znanego obrazu systemu. Jeśli zostanie naruszony, zostanie jedynie przepisany ze znanego, niezakłóconego obrazu systemu. Pozwala stworzyć enklawę przeglądania o niskim poziomie bezpieczeństwa i zagrożeniu w środowisku o wyższym poziomie bezpieczeństwa. Jest to podejście, które zostało zastosowane, w sensie fizycznym, dla pewności przez niektóre organizacje udowadniające publiczny dostęp do komputerów osobistych. Zamiast próbować wzmocnić maszyny przed kompromisem lub dołączeniem, są one ponownie inicjowane ze znanego obrazu po każdym użytkowniku. Pozwala to użytkownikowi końcowemu na podstępne próby partnerów handlowych narzucenia niebezpiecznych praktyk komputerowych w niepotrzebnym środowisku, które można odizolować. Korzystając z systemu Windows jako przykładu, podczas gdy instalowanie oprogramowania jako Administrator jest niebezpieczną praktyką, jest to o wiele mniej szkodliwe, jeśli robi się to na maszynie wirtualnej, gdzie można ją wygodnie usunąć z niewielkim efektem ubocznym. Podobnie, jednorazowe maszyny wirtualne mogą być używane przez zwykłych użytkowników do oddzielenia normalnego środowiska operacyjnego od środowiska, w którym zauważalny jest wzrost narażenia na ryzyko (np. Przeglądanie podejrzanych witryn; wymuszona instalacja formantów ActiveX o niepewnym pochodzeniu lub bezpieczeństwie).

OBOWIĄZKI SERWERA

Jak zauważono wcześniej , serwery WWW stanowią atrakcyjny wektor ataków. Metody podpisywania (uwierzytelniania) są sposobem kontrolowania potencjalnego uszkodzenia, pod warunkiem, że mechanizmy stosowane do przyjmowania kodu wykonywalnego są odpowiednio kontrolowane. Brak kontroli tych mechanizmów prowadzi do poważnych skutków ubocznych. Pojęcie minimalnego niezbędnego uprawnienia dotyczy kodu mobilnego. Jest mało powodów do narzucenia korzystania z ActiveX w celu zmiany koloru reklamy banerowej. Ten efekt często można osiągnąć za pomocą kaskadowych arkuszy stylów (CSS). JavaScript / ECMAScript może obsługiwać wiele zaawansowanych funkcji związanych z wyświetlaniem operacji o wysokim stopniu bezpieczeństwa w sytuacjach, w których CSS nie jest odpowiedni. Używanie Javy do obsługi koszyka (ceny, ilości i zawartości) jest rozsądne i nie wymaga użycia podpisanego apletu, co wiąże się z większymi możliwościami i ryzykiem. Na drugim końcu skali prawdopodobne jest, że funkcja aktualizacji systemu (np. funkcja Windows Update, która automatycznie pobiera i instaluje zmiany w systemie operacyjnym Windows) wymaga nieokiełznanej mocy podpisanej kontrolki ActiveX. Gdy wymagana jest moc podpisanych apletów lub elementów sterujących, praktyka dobrej inżynierii oprogramowania zapewnia doskonałe przykłady ograniczania ryzyka szkód i psot, jak omówiono w rozdziale 38 niniejszego podręcznika. Problemy te wykraczają poza bezpośrednie narażenie poszczególnych organizacji. Rosnącym trendem w ciągu ostatnich kilku lat było wykorzystywanie zewnętrznych serwerów WWW jako wektora infekcji, trampoliny do infekowania odwiedzających witrynę. Jeśli składniki są pobierane z serwerów innych

firm, integralność witryny WWW może być narażona na kompromis. Dobra implementacja oprogramowania izoluje funkcje i ogranicza zakres operacji wymagających uprzywilejowanego dostępu lub operacji. Aplikacje płac nie obsługują bezpośrednio portów drukarek, kart graficznych, kart sieciowych ani napędów dyskowych. Uprzywilejowany składniki systemu operacyjnego, takie jak sterowniki urządzeń i systemy plików, są odpowiedzialne za faktyczne działanie. Ta separacja, wraz z dokładnym sprawdzaniem parametrów przez jądro systemu operacyjnego i uprzywilejowane komponenty, zapewnia bezpieczeństwo. Tych samych technik można używać z apletami i kontrolkami. Ponieważ wymagają one większego dostępu, należy je starannie programować, stosując te same środki obronne, jakie stosuje się przy wdrażaniu uprzywilejowanych dodatków do systemów operacyjnych. Na przykład nie ma powodu, aby uprzywilejować serwer Simple Mail Transfer Protocol (SMTP). Serwer SMTP wymaga uprawnień do pojedynczej funkcji, dostarczania pojedynczej wiadomości e-mail do skrzynki pocztowej odbiorcy. Można to zrobić na dwa sposoby:
1. Wdróż aplikację w sposób nieuprzywilejowany, oznaczając pliki e-mail użytkowników i katalogi odpowiednimi uprawnieniami dostępu do programu dostarczania poczty w celu tworzenia i modyfikowania plików i katalogów e-mail. Taki mechanizm jest w pełni zgodny z poziomem bezpieczeństwa NCSC Orange Book na poziomie C2.
2. Zaimplementuj osobny podskładnik, którego wyłączną odpowiedzialnością jest faktyczne dostarczenie komunikatu w wiadomości. Podskładnik musi zostać napisany obronnie, aby sprawdzić wszystkie jego parametry, i nie zapewnia interfejsu do wykonywania dowolnego kodu. Tego podejścia używa system operacyjny HP OpenVMS. Natomiast program pocztowy sendmail w UNIX jest dużym, wielofunkcyjnym programem, który wykonuje się z uprawnieniami. sendmail jest przedmiotem wielu problemów związanych z bezpieczeństwem od ponad dekady i wzmógł starania o zapewnienie bezpieczniejszych zamienników

PODSUMOWANIE

Kod mobilny zapewnia wiele elastycznych i przydatnych funkcji. Różne mechanizmy implementacji kodu mobilnego obejmują zakres od niewinnego (HTML), do dość bezpiecznego (JavaScript / ECMAScript) oraz ze wzrostem mocy i ryzyka poprzez Javę i ActiveX. Zapewnienie bezpieczeństwa i integralności przy użyciu kodu mobilnego wymaga współpracy zarówno dostawcy, jak i klienta. Klienci nie powinni akceptować losowo podpisanego kodu i kontroli. Dostawcy mają pozytywny obowiązek:

* Postępuj zgodnie z dobrymi praktykami inżynierii oprogramowania
* Przyznaj minimum niezbędnych uprawnień i dostępu
* Użyj programowania obronnego
* Ogranicz uprzywilejowany dostęp, bez otwartych interfejsów
* Zapewnij integralność procesu podpisywania i powiązanych kluczy prywatnych

Z odpowiednią ostrożnością kod mobilny może być konstruktywną, potężną częścią aplikacji intranetowych i internetowych, zarówno w organizacji, jak i we współpracy z klientami i innymi zainteresowanymi stronami.