ZŁOŚLIWY KOD

Złośliwa logika (lub kod) to „sprzęt, oprogramowanie lub oprogramowanie układowe, które jest celowo dołączane do systemu w nieautoryzowanym celu". W tej sekcji wymieniono typowe rodzaje złośliwego kodu, źródła złośliwego kodu, metody replikacji złośliwego kodu oraz metody wykrywania złośliwego kodu. Badanie A2011 przeprowadzone na 200 małych i średnich firmach z maksymalnie 249 pracownikami wykazało, że „… dwa na pięć małych i średnich przedsiębiorstw mają pewność, że doszło do pewnego rodzaju naruszenia bezpieczeństwa w wyniku przekierowania pracowników do witryn sieci Web zawierających zainfekowane złośliwe oprogramowanie pobierane lub zostały uszkodzone przez złośliwy kod. "Typowe typy złośliwego kodu obejmują wirusy, robaki, konie trojańskie, oprogramowanie szpiegujące, rootkity i boty. Do nowych zagrożeń złośliwego kodu należą kod kleptograficzny, kryptowirusy i sprzętowe rootkity. Współczesne zagrożenia złośliwym kodem nie zawsze pasują do schludnych kategorii, co powoduje zamieszanie podczas omawiania tematu. Nie można sklasyfikować całego kodu jako dobrego lub złośliwego. W przypadku braku męskiej lub kryminalnej intencji autora lub użytkownika kod nie jest ani dobry, ani zły. Autorzy opracowują kod, aby osiągnąć jakiś cel lub spełnić jakiś cel, tak jak użytkownicy uruchamiają kod, aby osiągnąć jakiś cel lub cel. Dlatego kontekst użycia i zamiary właściciela określają, czy kod jest złośliwy.

MODEL ZAGROŻENIA KODEM ZŁOŚLIWYM

W modelu lub profilu zagrożenia aktor wykorzystuje dostęp do ukierunkowania zasobu za pomocą akcji, aby uzyskać wynik. Aby zrozumieć zakres problemu (i możliwe zapobieganie), warto zbadać zagrożenia złośliwym kodem w tym modelu. Podmioty mogą być zorganizowanymi lub nieustrukturyzowanymi zagrożeniami ze strony osób, organizacji lub państw narodowych. Dostęp to dozwolona fizyczna lub logiczna ścieżka do docelowego zasobu. Wykonanie złośliwego kodu lub logiki jest działaniem służącym do uzyskania pożądanego rezultatu. Rezultatem może być wywiad (np. Kradzież tajemnic handlowych), inwigilacja, rozpoznanie, zakłócenie działalności, zniszczenie mienia, rozgłos z jakiejś przyczyny lub negatywny rozgłos dla ofiary.

KODD SAMOREPLIKUJĄCY

Kod samoreplikujący się nie jest z natury złośliwy. Dobra oferta badań nad sztuczną inteligencją koncentruje się na iteracyjnej samoreplikacji. Hewlett-Packard i inni badali, w jaki sposób techniki autoreplikacji mogą dostarczyć korzystnego oprogramowania, takiego jak robaki do łatania kodu. John Von Neumann zaproponował podstawową koncepcję samoreplikacji kodu w swoim artykule z 1949 r. „Teoria autoreprodukujących się automatów". W 1961 r. W Bell Labs Douglas McIlroy, Victor Vyssotsky i Robert H. Morris zagrali w grę Darwin w które dwa przeciwne programy (robaki pamięci) wejdą do systemu, a tylko jeden wyjdzie. W wywiadzie dla jednego z autorów Roberta H. Morrisa, byłego Głównego Naukowca Agencji Bezpieczeństwa Narodowego (NSA), stwierdził:

Myśleliśmy o umieszczeniu dwóch programów w maszynie, która będzie ze sobą walczyć, a jeden wygra, a drugi umrze. Zasadniczo pojęcie i program zostały napisane przez McIlroya i Wysockiego, a ja byłem po stronie. Ale potem, kilka dni później, miałem dobry pomysł i po prostu wygrałem grę, a wszyscy inni się poddali… tylko dlatego, że wpadłem na bardzo dobry pomysł na napisanie go.

Chociaż pożądane byłoby życie w świecie wolnym od wojny, stale rosnąca integracja technologii i działań wojennych wymaga rozwoju ofensywnych możliwości prowadzenia wojny informacyjnej. Implanty złośliwego kodu mogą służyć użytecznym funkcjom wywiadowczym, inwigilacyjnym i rozpoznawczym w operacjach bezpieczeństwa narodowego i ścigania. Ponadto, badając i rozwijając techniki szkodliwego kodu, specjaliści mogą lepiej przygotować się do obrony przed rozwijającymi się zagrożeniami.

AKTORZY : POCHODZENIE ZŁOŚLIWEGO KODU

Przed omówieniem konkretnych rodzajów złośliwego kodu warto zrozumieć źródło i źródło ataków złośliwego kodu. Złośliwy kod może pochodzić ze strukturalnych lub nieustrukturyzowanych zagrożeń. Zagrożenia strukturalne obejmują państwa narodowe, przestępców korporacyjnych i przestępczość zorganizowaną. Nieustrukturyzowane zagrożenia obejmują nieuczciwych aktorów, takich jak indywidualni intruzi i tak zwanI script kiddies.

AKTORZY : ZAGROŻENIE STRULTURALNE

Zagrożenia strukturalne są zorganizowane, dobrze finansowane i mogą działać w perspektywie długoterminowej. Zagrożenia związane ze złośliwym kodem strukturalnym zwykle obejmują funkcje wywiadowcze, inwigilacyjne i rozpoznawcze. Zagrożenia strukturalne mogą wykorzystywać te możliwości do angażowania się w szpiegostwo przemysłowe, kontradyktoryjne operacje informacyjne lub poważne bieżące oszustwa i kradzieże. Przestępczość zorganizowana odpowiada za 90 procent zagrożeń złośliwym kodem. Ekstresjoniści atakują branżę hazardu online za pomocą ataków DDoS z rozproszonych komputerów osobistych. Przestępcy używają skompromitowanych systemów do angażowania się w oszustwa związane z giełdami typu „zrzuć", a jeden z mediów informuje o rocznych zyskach takich grup zbliżających się do 1 miliarda dolarów rocznie. Rośnie liczba incydentów związanych z wypożyczaniem oprogramowania, co wskazuje na dojrzewający rynek złośliwego kodu. W 2006 r. Izraelski sąd skazał kilka osób na więzienie i nakazał im zapłacić grzywny za pisanie i umieszczanie złośliwego kodu wykorzystywanego do szpiegowania korporacji i osób przez członków firm telekomunikacyjnych i handlowych. Skoordynowane, systematyczne ataki pochodzące z Chin rutynowo atakują obiekty obronne oraz badawczo-rozwojowe. W tekście Unrestricted Warfare z 1999 r. Dwóch chińskich oficerów lotnictwa wezwało do transformacji działań wojennych, która pociągałaby za sobą ciągłe ataki technologiczne na zachodnie aktywa.10 W chińskiej białej księdze wojskowej wydanej w 2006 r. Wezwano do strategii, dzięki której Chiny mogłyby wygrać „wojnę informacyjną" przeciwko Zachodowi, który jest „w szybkim tempie, zaawansowany technologicznie i zdigitalizowany". Chociaż chińscy urzędnicy publicznie zaprzeczają sponsorowaniu takich ataków, brak działań organów ścigania wskazuje przynajmniej tolerancję na ataki elektroniczne na zachodnie aktywa.

AKTORZY : NIEUSTRUKTURYZOWANE ZAGROŻENIA

Nieustrukturyzowane zagrożenia obejmują nieuczciwych aktorów, którzy nie działają w porozumieniu lub koordynacji z większymi podmiotami. Chociaż poważne ataki mogą wynikać z nieustrukturyzowanych zagrożeń, nie stanowią one takich samych długoterminowych wyzwań, jak zagrożenia strukturalne. W zeznaniu przed Kongresem Stanów Zjednoczonych były dyrektor NSA Kenneth Minihan stwierdził:

Nieustrukturyzowane zagrożenie jest losowe i względnie ograniczone. Składa się z przeciwników o ograniczonych środkach i organizacji oraz krótkoterminowych celach. Chociaż stanowi zagrożenie dla operacji systemu, bezpieczeństwo narodowe nie jest celem. To dziś najbardziej oczywiste zagrożenie. Zagrożenie strukturalne jest znacznie bardziej metodyczne i dobrze obsługiwane. Chociaż zagrożenie nieustrukturyzowane jest obecnie najbardziej oczywistym zagrożeniem, dla celów bezpieczeństwa narodowego zajmujemy się przede wszystkim zagrożeniem strukturalnym, ponieważ stanowi ono największe ryzyko.

DOSTĘP KONTRA DZIAŁANIE : WEKTOR KONTRA ŁADUNEK

Ataki złośliwego kodu obejmują wektor i ładunek. W biologii wektor jest czynnikiem, który przenosi (potencjalnie szkodliwy) materiał (kod) z jednej lokalizacji do drugiej. W atakach komputerowych wektor jest drogą dostępu, taką jak dozwolona ścieżka poprzez dostęp fizyczny lub sieć. Dostęp fizyczny

odbywa się za pośrednictwem personelu wewnętrznego lub innych osób mających dostęp do pomieszczeń. Dostęp przez sieć może odbywać się za pośrednictwem dozwolonej ścieżki do serwera sieci Web, dozwolonej ścieżki od złośliwego serwera sieci Web do klienta sieci Web, użytkownika za pośrednictwem załącznika do wiadomości e-mail lub innego procesu programowego przez dostępny port. Ładunek to funkcja (akcja) umieszczona w systemie w celu osiągnięcia pewnego celu. Ładunki mogą obejmować dodatkową złośliwą logikę, oprogramowanie do zdalnego dostępu (rootkity i tym podobne) lub oprogramowanie do zdalnego sterowania (robot lub bot) w celu osiągnięcia pewnego celu (spamowanie, ataki DDoS, i tak dalej).

PRZEGLĄD ZŁOŚLIWEGO KODU

WIRUSY

W biologii wirus jest „organizmem zakaźnym, który jest zwykle submikroskopowy, może namnażać się tylko w niektórych żywych komórkach gospodarza i jest obecnie rozumiany jako struktura niekomórkowa pozbawiona jakiegokolwiek wewnętrznego metabolizmu i zwykle zawierająca rdzeń DNA lub RNA wewnątrz płaszcza białkowego ". W kategoriach komputerowych wirus to samoreplikujący się kod, który wymaga pliku wykonywalnego lub dokumentu hosta oraz pomocy człowieka do replikacji. Joe Dellinger stworzył pierwszego wirusa dla systemu operacyjnego Apple w 1982 roku, podczas gdy student Texas A & M.14 Fred Cohen stworzył pierwszego wirusa komputerowego VAX w 1983 roku w ramach swoich badań doktoranckich15 Len Adleman, A in RSA, pierwszy zastosował biologicznego wirusa metafory do opisania wyników. Typy wirusów to sektor rozruchowy, infektor plików, wirus makr, bomba logiczna, wirusy skryptów krzyżowych (naprawdę forma robaka), wirusy polimorficzne i kryptowirusy. Jednak współczesne zagrożenia złośliwym kodem nie dzielą się tak starannie na kategorie.

WIRUS SEKTORA ROZRUCHOWEGO

Gdy użytkownicy uruchamiają komputer z nośnika cyfrowego, pozwalają na uruchomienie kodu znajdującego się w sektorze rozruchowym nośnika na mikroprocesorze przy bardzo niewielkim pośrednictwie. Wirusy sektora rozruchowego w latach 80. i 90. XX wieku wykorzystywały ten mechanizm do rozprzestrzeniania się za pośrednictwem zainfekowanych nośników wymiennych, takich jak (obecnie wymarłe) dyskietki. Jeśli użytkownik błędnie uruchomi się z zainfekowanej dyskietki, dysku kompaktowego (CD), DVD lub dysku flash, wirus skopiuje się do sektora rozruchowego dysku twardego. Następnie szkodliwy kod programu kopiuje się na każdy nośnik wstawiony do systemu. Chociaż podobno nadal istnieją, wirusy sektora rozruchowego stanowią bardzo niewiele współczesnych zagrożeń złośliwym kodem. Virus Bulletin donosi: „Ostatnie wirusy sektora startowego spadły z WildList na początku 2006 roku, ale w naszych raportach dotyczących rozpowszechniania nadal pojawiają się różne typy, a partia laptopów zainfekowanych„ Stoned.Angelina "została wydana w Niemczech i Danii w połowie 2007 roku . Niedawno zaobserwowano [T] rojany stosujące podobne techniki do sadzenia rootkitów w celu ukrycia swojej działalności

WIRUS INFEKUJĄCY PLIKI

Wirusy infekujące pliki wstawiały się do programów obecnych w systemie hosta. Za każdym razem, gdy użytkownik uruchomił program hosta (zwykle plik .EXE lub .COM), złośliwy kod wykorzystywał okazję do włożenia się do pamięci o swobodnym dostępie (RAM), a następnie replikacji do innych plików w systemie. Chociaż prawdopodobnie nadal istnieją, wirusy infekujące pliki stanowią stosunkowo niewiele współczesnych zagrożeń złośliwym kodem.

MAKROWIRUSY

Makrowirusy rozprzestrzeniają się za pośrednictwem języków definicji makr używanych przez niektóre aplikacje. Najczęściej nadużywany jest język skryptowy Visual Basic for Applications (VBA) opracowany przez firmę Microsoft w celu umożliwienia automatyzacji funkcji w pakiecie produktów pakietu Office. Każda funkcja pozwalająca na automatyzację jest prawdopodobnym punktem ataku. Deweloperzy powinni dokładnie rozważyć bezpieczeństwo i łatwość użytkowania, dołączając takie funkcje do produktów. Pierwszy wirus makr atakujący Microsoft Word pojawił się na wolności w 1995 roku. Od tego czasu makrowirusy stanowiły znaczną liczbę udanych ataków.

BOMBY LOGICZNE

Bomba logiczna jest formą funkcji złośliwego kodu wbudowaną czasami w wirusy, które czekają na aktywację sekwencji zdarzeń, takich jak zniknięcie rekordu pracownika z bazy danych zasobów ludzkich lub określonej daty i godziny. Kabay podał kilka przykładów wczesnych bomb logicznych w artykule Network World z 2002 roku:

W 1985 r. Niezadowolony funkcjonariusz ds. Bezpieczeństwa komputerowego w firmie pośrednictwa ubezpieczeniowego w Teksasie założył złożoną serię programów Job Control Language (JCL) i RPG (stary język programowania), opisanych później jako „tripwires i bomby zegarowe". Na przykład zmodyfikowano funkcję rutynowego wyszukiwania danych, aby spowodować wyłączenie komputera klasy średniej EBMSystem / 38. Zaprogramowano inną procedurę usuwania losowych fragmentów pamięci głównej, zmiany własnej nazwy i resetowania, aby wykonać miesiąc później. W 1992 roku programista komputerowy został ukarany grzywną w wysokości 5000 $ za pozostawienie bomby logicznej w General Dynamics. Zamierzał wrócić po tym, jak jego program usunie krytyczne dane i dostanie dużo pieniędzy, aby rozwiązać problem. Bomby zegarowe są podklasą bomb logicznych, które „wybuchają" w określonym czasie. Niektóre z pierwszych wirusów, napisanych w latach 80., były bombami zegarowymi. Na przykład niesławny wirus „Piątek 13" był bombą zegarową; powielało się w każdy piątek i 13 dnia miesiąca, powodując spowolnienie systemu. Ponadto w każdy piątek trzynastego roku uszkodził również wszystkie dostępne dyski. Wirus Michała Anioła z początku lat 90. XX wieku - jeden z pierwszych wirusów, który dostał się do publicznej wiadomości z powodu doniesień prasowych - próbował uszkodzić katalogi dysków twardych 6 marca. Wirus Win32.Kriz.3862, odkryty w 1999 roku, detonuje w Boże Narodzenie; jego ładowność obejmuje ogromne nadpisywanie danych we wszystkich jednostkach pamięci, a także uszkodzenie systemu BIOS. W 2000 roku mężczyzna z Stamford w stanie Connecticut został oskarżony w Sądzie Najwyższym stanu Nowy Jork na Manhattanie pod zarzutem nieautoryzowanych modyfikacji systemu komputerowego i wielkiej kradzieży. Oskarżony pracował w Deutsche Morgan Grenfell od 1996 roku jako programista. Do końca 1996 r. Stał się sprzedawcą papierów wartościowych. W akcie oskarżenia postawiono programistyczną bombę zegarową w modelu ryzyka, nad którym pracował jako programista; datą uruchomienia był lipiec 2000 r. Nieautoryzowany kod został odkryty przez innych programistów, którzy najwyraźniej musieli spędzać miesiące naprawiając program z powodu nieautoryzowanych zmian, które rzekomo wprowadził oskarżony.

W 2002 roku pracownik UBS PaineWebber podłożył bombę logiczną na serwerach swojego pracodawcy w ramach próby manipulacji zapasami. Sprawca, Roger Duronio, nabył liczne kontrakty na akcje opcji sprzedaży, pozwalając mu na sprzedażUBSstock po stałej, wysokiej cenie. 4 marca 2002 roku, o godzinie 9:30, bomba logiczna zaczęła usuwać pliki na ponad 1000 komputerów, powodując szkody o wartości 3 milionów dolarów. Duronio pomyślał, że skutki bomby logicznej obniżyłyby wartość akcji UBS, pozwalając mu na osiągnięcie dużych zysków z opcji na akcje. Spisek Duronio nie powiódł się, a jego zysk się nie zmaterializował. Władze federalne oskarżyły go później o papiery wartościowe i oszustwa komputerowe. W 2006 r. Sędzia skazał Duronio na 97 miesięcy więzienia za atak. W marcu 2013 r. Bomba logiczna nadpisała dane na dyskach twardych w bankach i nadawcach Korei Południowej. Kim Zetter z WIRED napisał:

Według Richarda Hendersona, badacza zagrożeń dla FortiGuard Labs z Vancouver, działu badań firmy ochroniarskiej Fortinet, bomba logiczna podyktowała datę i czas, kiedy złośliwi zaczęli wymazywać dane z maszyn, aby koordynować zniszczenie wielu ofiar. Atak, który uderzył w maszyny 20 marca, zniszczył dyski twarde i główny rekord rozruchowy co najmniej trzech banków i dwóch firm medialnych jednocześnie. Ataki podobno spowodowały, że niektóre bankomaty przestały działać, uniemożliwiając Koreańczykom południowym wycofanie z nich gotówki. Złośliwe oprogramowanie składało się z czterech plików, w tym jednego o nazwie AgentBase.exe, który uruchomił czyszczenie. W tym pliku znajdował się ciąg szesnastkowy (4DAD4678) wskazujący datę i godzinę rozpoczęcia ataku - 20 marca 2013 r. O godzinie 14.00 czasu lokalnego (2013-3-20 14:00:00). Gdy tylko wewnętrzny zegar na maszynie wybił godzinę 14:00:01, wycieraczka została uruchomiona, aby zastąpić twarde
napęd i główny rekord rozruchowy na komputerach z systemem Microsoft Windows, a następnie ponownie uruchom system

WIRUSY SKRYPTÓW KRZYŻOWYCH (LUB ROBAKI)

Wirusy (lub robaki) skryptów cross-site scripting (XSS) replikowane są przez wadliwe serwery aplikacji Web i kod klienta. W 2005 r. Pojawił się exploit dotyczący skryptów krzyżowych obejmujący serwis społecznościowy MySpace oraz błędy w przeglądarce Microsoft Internet Explorer, w których użytkownik o imieniu Samy zgromadził ponad milion znajomych w ciągu nocy, używając błędu wstawiania JavaScript22. Sąd Najwyższy w Los Angeles skazał autora Samy Kamkara na trzyletni okres próbny i 90 dni pracy w społeczności za jego czyny. W doskonałej recenzji XSS Sherif Koussa napisał:
Skrypty między witrynami to atak wymierzony w użytkowników aplikacji. Prostota skryptów między witrynami jest również powodem, dla którego jest tak potężny. Jeśli aplikacja jest podatna na skrypty między witrynami, programista nie odpowiada już za to, co działa w przeglądarce użytkownika… atakujący. Skrypty między witrynami mogą być używane w atakach takich jak przejęcie uwierzytelnienia i przejęcie sesji. Moc skryptu między witrynami przejawia się jeszcze bardziej w połączeniu z fałszowaniem żądań między witrynami.… Później przytacza przypadek ataku na Twitterze: w 2009 r. 17-letni Mikey Mooney podniósł odpowiedzialność za atakowanie Twittera za pomocą XSS techniki: „… co najmniej cztery osobne warianty oryginalnego robaka StalkDaily.com XSS trafiły na popularną witrynę mikroblogowania Twitter, automatycznie przejmując konta i reklamując witrynę autora, publikując tweety w imieniu właścicieli kont, wykorzystując skrypty między witrynami wady strony ”. Pisarz Dancho Danchev podał więcej szczegółów na temat ataku w swoim artykule ZDNet.

0WIRUSY POLIMORFICZNE

Kod polimorficzny modyfikuje się, aby uniknąć wykrycia. Kod wirusa może to osiągnąć, dynamicznie składając się ponownie w celu zmodyfikowania podstawowej struktury przy jednoczesnym zachowaniu ogólnej funkcjonalności. W innych metodach wirus jest szyfrowany, kodowany lub pakowany, a program ładujący odszyfrowuje, dekoduje lub rozpakowuje wirusa w czasie wykonywania. UPX, Ultimate Packer dla eXecutables, jest obecnie najczęściej stosowanym formatem packera. W raporcie o zagrożeniach bezpieczeństwa SOPHOS 2013 autorzy piszą:

Polimorfizm nie jest nowym pomysłem - autorzy szkodliwego oprogramowania używają go od 20 lat. Mówiąc wprost, kod polimorficzny zmienia swój wygląd, próbując uniknąć wykrycia, bez zmiany jego zachowania lub celów. Jeśli program wygląda inaczej, atakujący mają nadzieję, że oprogramowanie antywirusowe może go przegapić. Lub oprogramowanie antywirusowe może zostać zmuszone do wygenerowania zbyt wielu fałszywych alarmów, co skłoni użytkowników do jego wyłączenia. W ataku polimorficznym kod jest zwykle szyfrowany, aby wydawał się bez znaczenia, i łączony z deszyfratorem, który przekształca go z powrotem w formę, którą można wykonać. Za każdym razem, gdy jest odszyfrowywany, silnik mutacji zmienia swoją składnię, semantykę lub oba. Na przykład autorzy szkodliwego oprogramowania dla systemu Windows często używali strukturalnej obsługi wyjątków w celu zaciemnienia przepływu kontroli i utrudnienia wykonywania statycznej analizy programów przed

ich uruchomieniem. Tradycyjne wirusy polimorficzne są samowystarczalne i muszą zawierać silnik mutacji zamówienia do replikacji. Sophos i inne firmy zajmujące się bezpieczeństwem są biegłe w wykrywaniu tych form złośliwego oprogramowania. Dostęp do silnika mutacji ułatwia analizę jego zachowania. Obecnie osoby atakujące szybko przechodzą na złośliwe oprogramowanie rozproszone w Internecie, polegające na polimorfizmie po stronie serwera (SSP). Teraz silnik mutacji i powiązane narzędzia są hostowane całkowicie na serwerze. Przestępcy mogą używać tych narzędzi do tworzenia różnorodnych treści plików w locie. Odbiorcy tej zawartości (czy to Windows .exe, Adobe PDF, JavaScript, czy cokolwiek innego) widzą tylko jeden przykład tego, co silnik może stworzyć. Nie widzą samego silnika.

WIRUSY KRYTPOGRAFICZNE

Kryptowirusy używają szyfrowania do szyfrowania danych, co czyni je niedostępnymi dla użytkownika. Chociaż niektóre wirusy używają algorytmów symetrycznych do samoszyfrowania w celu uniknięcia wykrycia, takie algorytmy nie są odpowiednie dla wirusów szyfrujących dane, ponieważ każda kopia wirusa da kopię klucza symetrycznego. Z tego powodu właściwe kryptowirusy wykorzystują techniki asymetryczne. Wirus Gpcode szyfruje pliki według rozszerzenia (.xls, .doc itp.) I pozostawia wiadomość tekstową, zachęcającą użytkownika do wysłania e-maila na podany adres w celu zapłacenia okupu za dostęp do jego plików. Następna generacja kryptowirusów prawdopodobnie będzie wykorzystywać hybrydowe kryptosystemy. W raporcie z 2012 r. Dotyczącym kryptowirusów naukowcy opracowali modele wirusów typu proof-of-concept, wykorzystujące kryptosystem klucza publicznego (PKC) do ukrywania się przed skanerami, a następnie aktywowania w odpowiedzi na zaszyfrowany klucz lub bilet. Ich lista zastosowań kryptografii w złośliwym oprogramowaniu obejmuje:

* Oprzyj się inżynierii odwrotnej
* Popraw anonimowość komunikacji między kontrolerami a złośliwym oprogramowaniem
* Skuteczniejsza kradzież danych i ataki typu „odmowa usługi”
* Zdalnie sterowane tylne drzwi do wymuszenia

ROBAKI

Termin robak odnosi się do dowolnej formy samoreplikującego się kodu, który nie integruje się z kodem wykonywalnym. Typowe wektory robaków obejmują wrażliwe usługi, pocztę e-mail, komunikatory internetowe i otwarte udostępnienia plików. Wiele robaków używa wielu wektorów. Na przykład robak Nimda rozprzestrzenia się za pośrednictwem poczty e-mail, luk w zabezpieczeniach serwera WWW, hostuje złośliwe strony internetowe i otwiera udostępnienia plików. Pierwszą infekcją robaka na dużą skalę był robak Morris (znany również jako Internet), wydany przez Roberta T. Morrisa 2 listopada 1988 r., Gdy był studentem Uniwersytetu Cornell. Robak wykorzystywał wady w usługach Sendmail i finger w celu replikacji i zainfekowania prawie 10 procent hostów internetowych. Chociaż twierdził, że był to nieudany eksperyment, Morris stał się pierwszą osobą skazaną za komputer
Ustawa o oszustwach i nadużyciach. W 2001 r. Nicholas Weaver stworzył termin „Warhol Worms” 28 dla tych robaków, które miały zdolność bardzo szybkiego rozprzestrzeniania się (biorąc pod uwagę słynny żart Warhola, że w przyszłości każdy będzie miał 15 minut sławy). Robak SQL Slammer stał się pierwszym takim robakiem, gdy w ciągu 10 minut zainfekował około 90 procent podatnych systemów.29 Robak Slammer zreplikował się z powodu usterki serwera bazy danych Microsoft SQL. Ponieważ został on zainstalowany wraz z innymi komponentami, takimi jak Visual Studio, wiele osób nawet nie wiedziało, że działa serwer SQL. Slammer był niezwykle skuteczny, ponieważ pakiet serwisowy Microsoft obniżył wcześniej łataną bibliotekę linków dynamicznych (dll) do starszej, podatnej na ataki wersji. W szczególności Microsoft wydał cztery aktualizacje ssnetlib.dll w 2002 roku. Niestety w październiku poprawka Q317748 obniżyła wersję ssnetlib.dll do wersji podatnej na atak, co ironicznie czyni tych, którzy najwierniej zastosowali łaty i poprawki, są najbardziej podatni na Slammer. Robak SQL Slammer jest interesującym studium przypadku, ponieważ był wektorem bez ładowności.

376-bajtowy robak działał w pamięci bez dotykania dysku twardego, co doprowadziło niektórych do przekonania, że Slammer był eksperymentem w technikach propagacji. Ponieważ był on oparty na protokole UDP (User Datagram Protocol) i podróżował w jednym pakiecie, narzut był minimalny, a współczynnik propagacji większy niż jakiekolwiek poprzednie zagrożenie złośliwym kodem. Ciekawe jest również to, że autor (autorzy) wydali go w sobotę. Nie wiadomo, dlaczego złośliwy atakujący celowo wypuścił takiego wściekłego wirusa w ciągu tygodnia, co zmniejszyłoby ogólny wpływ na biznes. Gdyby autor wypuścił robaka w szczytowy dzień roboczy, taki jak wtorek, spowodowane szkody byłyby znacznie poważniejsze. Robaki są wykorzystywane do rozprzestrzeniania innych form złośliwego oprogramowania, takich jak trojany, oprogramowanie szpiegujące, rootkity oraz kanały zdalnego sterowania i kontroli zwane botami. Oznaką ogólnej dojrzałości ataków złośliwego kodu jest robak Bagel, który jest produkowany przynajmniej od 2004 roku. Bagel, podobnie jak wiele innych robaków pocztowych, przybywa jako wiadomość e-mail ze złośliwym załącznikiem. Gdy użytkownik uruchamia załącznik, wykonywany jest ładunek, który wykonuje szereg czynności w zależności od wariantu.

PÓŹNIEJSZE WARIANTY

Bagel instaluje otwartą strukturę aplikacji, którą zdalni napastnicy mogą aktualizować i rozszerzać. Zebrane systemy dostarczają spam (niechciane komercyjne wiadomości e-mail), zbierają dodatkowe adresy i działają jako punkt pośredni dla innych ataków. Autor (autorzy) wydają się postępować zgodnie z wyrafinowaną metodologią tworzenia oprogramowania i procesem testowania. W 2007 roku atakujący wydali 30 000 nowych wariantów w sześciotygodniowym okresie. W kwietniu 2012 r. Robak Flashback „zainfekował ponad 650 000 systemów Mac OS X, wykorzystując lukę w zabezpieczeniach Java firmy Apple". Analitycy z F-Secure napisali, że Flashback jest najbardziej zaawansowanym złośliwym oprogramowaniem dla systemu OS X, jakie kiedykolwiek widzieliśmy. Oferuje szereg nowatorskich rozwiązań tego rodzaju. Był to zarówno pierwszy, który był świadomy VMware, jak i pierwszy, który wyłączył XProtect, wbudowany program ochrony przed złośliwym oprogramowaniem OS X. Obie te funkcje zostały usunięte z późniejszych wariantów (pierwsza prawdopodobnie w celu uniknięcia wykrycia heurystycznego, a druga prawdopodobnie po tym, jak autorzy zdali sobie sprawę, że jest to niepotrzebne, ponieważ XProtect nie został zaprojektowany do ochrony przed plikami nie poddanymi kwarantannie). Ich usunięcie wskazuje, że Flashback jest aktywnie sprawdzany i ulepszany przez jego autorów. Kolejnym interesującym pierwszym jest wykorzystanie przez Flashbacka niezałatanej luki w dystrybucji Java OS X, która pozwoliła mu zainfekować ponad 650 000 komputerów Mac na całym świecie.… To spowodowało, że Flashback był mniej więcej tak powszechny na komputerach Mac, jak Conficker dla Windows.… Oznacza to, że Flashback jest nie tylko najbardziej zaawansowanym, ale także najbardziej udanym złośliwym oprogramowaniem dla systemu OS X, jakie widzieliśmy do tej pory. Strategia infekcji Flashbacka ma na celu wybranie niezabezpieczonych systemów i nie zainfekuje komputera, jeśli zostanie znalezione określone oprogramowanie zabezpieczające lub narzędzia analityczne. Oznacza to, że autorzy Flashbacka atakują użytkowników mniej świadomych bezpieczeństwa kosztem całkowitej liczby potencjalnych celów. To okazuje się być skuteczną strategią, jako badacz bezpieczeństwa . To okazuje się być skuteczną strategią, ponieważ badacze bezpieczeństwa mieli trudności z uzyskaniem wystarczającej liczby próbek od użytkowników. Autor Flashback popełnił błąd, ostrzegając użytkowników o obecności infekcji, a następnie doprowadzając do masowego odkrycia złośliwego oprogramowania.

TROJANY

Aplikacja konia trojańskiego, podobnie jak koń mitologii greckiej, pełni zarówno funkcję jawną, jak i ukrytą. Chociaż osoby atakujące mogą wykorzystywać robaki jako jeden z możliwych wektorów propagacyjnych, trojany wymagają, aby użytkownik uruchomił złośliwy program, aby był skuteczny. Trojany zwykle wykorzystują jakąś formę inżynierii społecznej lub manipulacji, aby przekonać użytkownika do uruchomienia programu. Robaki pocztowe mogą wydawać się pochodzić od znanego

współpracownika i w ten sposób oszukać użytkownika. Inne trojany mogą mieć formę gier, bezpłatnych ofert, zdjęć popularnych celebrytów lub plików w serwisach wymiany plików peer-to-peer. W jednym badaniu usługi wymiany plików peer-to-peer Limewire stwierdzono, że „68% wszystkich odpowiedzi do pobrania zawierających pliki wykonywalne, archiwalne i rozszerzenia plików Microsoft Office" zawierało złośliwe oprogramowanie. Zapytania dotyczące filmów najprawdopodobniej zawierały złośliwy kod. Tajna funkcja aplikacji konia trojańskiego jest zazwyczaj jakąś formą zdalnego trojana, keyloggera, dialera, bota IRC lub rootkita. Trojany zdalnego dostępu zapewniają pełny zdalny dostęp do systemu. Twórcy trojana Bo2 K uważają go za „najpotężniejsze narzędzie do zarządzania siecią dostępne dla środowiska Microsoft" i nalegają na fakt, że funkcjonalnie nie różni się on od innych rozwiązań zdalnego dostępu. Trojany Keylogging rejestrują naciśnięcia klawiszy i okresowo przesyłają dane do zdalnego użytkownika. Atakujący dokonali kradzieży kodu źródłowego systemu Windows 2000 przy użyciu poświadczeń skradzionych przez trojana QAZ zainstalowanego na komputerze zdalnym. Dialer to rodzaj trojana, który cicho wybiera zdalne numery opłat drogowych i pobiera duży rachunek telefoniczny dla ofiary. Boty IRC działają jako autonomiczne klienty IRC. Wczesne boty IRC zapewniały wsparcie techniczne i funkcje monitorowania kanałów, ale obecnie są szeroko wykorzystywane do szkodliwych celów, takich jak funkcje dowodzenia i kontroli. W 2013 r. Kaspersky Labs poinformował o „kompilacji trojana specjalnie dla smartfonów z systemem Android". Sean Gallagher napisał:
„25 marca konto e-mail tybetańskiego aktywisty zostało zhakowane, a następnie wykorzystane do dystrybucji złośliwego oprogramowania dla Androida na listę kontaktów aktywisty. Przynęta e-maila była oświadczeniem z ostatniej konferencji zorganizowanej przez Światowy Kongres Ujgur.… Jeśli cele otworzyły załącznik… otrzymali złośliwe oprogramowanie spakowane w pliku APK Androida. Po otwarciu trojan instaluje aplikację o nazwie „Konferencja" na komputerach z systemem Android. Jeśli aplikacja zostanie uruchomiona, wyświetli fałszywą wiadomość od przewodniczącego WUC - jednocześnie odsyła wiadomość do serwera dowodzenia i kontroli, aby zgłosić jej pomyślną instalację. Złośliwe oprogramowanie zapewnia tylne wejście do urządzenia za pośrednictwem wiadomości SMS wysyłanych przez serwer. Na polecenie zwraca listy kontaktów telefonu, dzienniki połączeń, dane o smartfonie, dane geolokalizacyjne i wszelkie wiadomości SMS zapisane na nim na serwerze za pośrednictwem przesyłania internetowego POST. Sam serwer działa na skonfigurowanym w języku chińskim komputerze z systemem Windows Server 2003, siedzącym w centrum danych w Los Angeles. Oprócz zapewnienia punktu przesyłania danych skradzionych z urządzeń ndroid, na swojej stronie głównej znajduje się także więcej złośliwego oprogramowania dla Androida i udostępnia publiczny interfejs sieci Web (w języku chińskim), który umożliwia bezpośrednią kontrolę nad telefonami zainfekowanymi tym złośliwym oprogramowaniem. Chociaż sam serwer znajduje się pod adresem IP zarejestrowanym w firmie o nazwie Emagine Concept, domena wskazana na maszynie jest zarejestrowana w chińskiej firmie Shanghai Meicheng Technology Information Development Co., Ltd., z kontaktem w Pekinie. „
W ten sposób trojan dystrybuuje oprogramowanie szpiegujące, następny temat

PROGRAMY SZPIEGUJĄCE

Termin oprogramowanie szpiegujące odnosi się do każdego oprogramowania, które gromadzi informacje o użytkownikach bez zgody. Popularne odmiany programów szpiegujących zbierają informacje o korzystaniu z Internetu, wyświetlają treści reklamowe (wyskakujące okienka), rejestrują naciśnięcia klawiszy, angażują się w oszustwa związane z kliknięciami lub monitorują użycie programu i licencje. Nieautoryzowany dostęp lub przekroczenie uprawnień w systemie komputerowym stanowi naruszenie Ustawy o oszustwach i nadużyciach komputerowych. Mimo że niektóre programy szpiegujące mogą być nielegalne, niektórzy programiści twierdzą, że prowadzą legalną działalność gospodarczą. Takie oprogramowanie szpiegujące stanowi umowę licencyjną użytkownika końcowego (EULA), która pozwala użytkownikowi zrezygnować z instalacji. Ponieważ użytkownik musi wyrazić zgodę na zainstalowanie oprogramowania, prawie na pewno jest to legalne. Jednak to, czy jest to etyczne i rozsądne, to inna sprawa. Sony przyciągnęło znaczną negatywną reklamę ze względu na wykorzystanie technologii spyware w celu ograniczenia zdolności słuchaczy do kopiowania

muzycznych płyt kompaktowych. Z powodu tej działalności Sony stanęło przed sądem w wielu stanach, a w 2005 r. rozstrzygnęła pozew zbiorowy. Świadoma zgoda jest podstawową zasadą w każdym systemie monitorowania. Organizacje powinny dążyć do otwartości w tych sprawach, aby utrzymać pozytywny profil publiczny i uniknąć problemów prawnych. W maju 2013 r. Pojawiły się informacje o wykorzystaniu oprogramowania szpiegującego do monitorowania komunikacji działacza na rzecz praw człowieka w Bahrajnie. Na rozprawie sądowej pojawiły się szczegóły opisujące, w jaki sposób dr Ala'a Shehabi była monitorowana przez infekcję jej systemów przez atak phishingowy:
Według zeznań świadka, kilka tygodni po aresztowaniu Shehabi otrzymała szereg e-maili, pierwszy rzekomo od Kahila Marzou, który był zastępcą szefa głównej partii opozycyjnej Bahrajnu, w tym zawierający wirusa. Inne wiadomości e-mail, które rzekomo pochodzą od dziennikarza Al Jazeera, również zostały zainfekowane. Badania wykazały, że e-maile zawierały produkt o nazwie FinSpy, dystrybuowany przez brytyjską firmę Gamma International. W zeznaniu świadka stwierdzono, że gdy komputer danej osoby zostanie zainfekowany FinSpy, „umożliwia dostęp do wiadomości e-mail, wiadomości w mediach społecznościowych i połączeń Skype, a także kopiowanie plików zapisanych na dysku twardym. Te produkty umożliwiają także każdemu, kto wykonuje celowanie, dowodzenie i zdalną obsługę mikrofonów i kamer na komputerach i telefonach komórkowych

ROOTKITY

Rootkit składa się z zestawu narzędzi do potajemnego naruszania systemu i utrzymywania dostępu administracyjnego (root) dla intruza. Współczesne rootkity łamią system na poziomie aplikacji, biblioteki, jądra, hiperwizora lub sprzętu. Wczesne rootkity na poziomie aplikacji atakowały systemy podobne do UNIX i zastępowały standardowe narzędzia systemowe (netstat, ps itp.) Wersjami, które pomijały informacje o intruzie, takie jak otwarte porty, uruchomione procesy, otwarte pliki i inne działania. Rootki na poziomie API modyfikują lub łatają systemową tabelę wywołań, aby przekierowywać wywołania systemowe (jak atak małpy na środku interfejsu API). Rootkity na poziomie jądra działają jako sterowniki urządzeń i dynamicznie ładują się do jądra; narażając na szwank integralność rdzenia systemu operacyjnego w celu filtrowania informacji prezentowanych użytkownikom poprzez procesy na poziomie aplikacji. Prawie wszystkie nowoczesne systemy operacyjne wykorzystują wirtualizację do zarządzania pamięcią i procesami. System operacyjny zwykle działa w pierścieniu 0 mikroprocesora; najbardziej uprzywilejowany poziom. Hiperwizor to warstwa kodu między systemem operacyjnym a sprzętem, która oszukuje system operacyjny, aby uwierzył, że działa on w pierścieniu 0. Hiperwizor monitoruje i arbitrażuje wymianę między maszynami wirtualnymi a rzeczywistym sprzętem. Projektowanie hiperwizora jest kluczową częścią zaufanego środowiska obliczeniowego, a projektanci mikroprocesorów, tacy jak Intel i AMD, mają w swoich produktach ulepszone możliwości wirtualizacji sprzętowej. Nowa generacja technologii rootkit wykorzystuje ten framework do tworzenia potencjalnie niewykrywalnego kodu. Inne potencjalne zagrożenia związane z rootkitami pochodzą od osób wewnętrznych, projektantów sprzętu i producentów. Osoba atakująca z dostępem fizycznym może wstawić komponent peryferyjny do komputera, aby utworzyć logicznie niewykrywalnego rootkita. Producent lub projektant sprzętu może zawrzeć w mikroukładzie funkcje podobne do rootkitów. Sprzedaż działu komputerów osobistych IBM w 2005 r. Chińskiemu producentowi Lenovo wzbudziła w Stanach Zjednoczonych wystarczające obawy, aby skłonić krajową ocenę bezpieczeństwa transakcji. Przegląd przeprowadzony przez House House on Foreign Investment w Stanach Zjednoczonych (CFIUS) ostatecznie zatwierdził sprzedaż pomimo obaw wyrażonych przez przedstawicieli Henry'ego Hyde'a i Dona Manzullo z Illinois oraz przedstawiciela Duncana Huntera z Kalifornii.

BOTY IRC

Boty IRC są niezależnymi agentami korzystającymi z Internet Relay Chat (IRC) w celu oferowania interaktywnych usług, takich jak monitorowanie kanałów, wsparcie, usługi informacyjne i gry. Greg Lindhal napisał pierwszy bot GM IRC (master game), który poprowadził użytkowników poprzez

tekstową grę RPG „Hunt the Wumpus”. W 1999 roku robak PrettyPark stał się pierwszym robakiem, który wykorzystywał IRC jako kanał zdalnego sterowania. Zainfekowane systemy sprawdzałyby serwer i kanał IRC, aby pobierać aktualizacje i przesyłać skradzione dane. Atakujący tworzą złośliwe boty do przeprowadzania ataków DDoS (boty DDoS), wysyłania niechcianych komercyjnych wiadomości e-mail (SpamBots) oraz angażowania się w wykorzystywanie, kradzież, lub oszustwo. Przykładami obecnych botów są GTbot, SDbot, Agobot, Goabot, Randex, Spybot i Phatbot. Obecna generacja technologii botów obejmuje keylogging, skanowanie portów, wykorzystanie, sniffowanie pakietów, ukrywanie procesów i oszustwa adware. Pojedyncza sieć tych zaatakowanych systemów może osiągnąć ponad 100 000 botów. Naukowcy oszacowali w 2004 r., Że obecnie ponad milion botów jest podłączonych do Internetu. Pasterze botów wynajmują te masowe systemy sieci rozproszonych organizacjom przestępczym. W 2006 r. Sąd w Kalifornii skazał jednego z takich botderów, Jeansona Anchetę, na pięć lat więzienia za spiskowanie w celu naruszenia Ustawy o oszustwach komputerowych i nadużyciach, spiskowanie w celu naruszenia Ustawy CAN-SPAM, powodując uszkodzenie komputerów używanych przez rząd federalny w obrona narodowa i dostęp do chronionych komputerów bez upoważnienia, w celu popełnienia oszustwa

ZŁOŚLIWY KOD MOBILNY

Serwery sieciowe mogą hostować strony zawierające złośliwy kod mobilny. Formanty ActiveX, aplety Java, JavaScript, animacje Adobe Flash i każdy inny rodzaj dynamicznego kodu wykonawczego można pobrać do systemu użytkownika i uruchomić w jego kontekście ze wszystkimi powiązanymi uprawnieniami. Złośliwe serwery WWW to jedna z metod upuszczania trojanów i botów na komputery. Osoba atakująca może użyć spamu lub ataku phishingowego, aby zachęcić użytkowników do kliknięcia łącza osadzonego w wiadomości e-mail. Innym wektorem jest użycie nazwy domeny podobnej do legalnej organizacji. Jeszcze innym wektorem witryny jest oferowanie informacji w niektórych kwestiach. Po zaindeksowaniu w głównej lub wyszukiwarkach treść ta przyciągnie wielu użytkowników do witryny.

WYKRYWANIE ZŁEGO KODU.

Typowe metody wykrywania szkodliwego kodu obejmują techniki heurystyczne oparte na sygnaturach, sieci i behawioralne. Jednak już w pracy z 1984 r. Wirusy komputerowe - teoria i eksperymenty Fred Cohen wykazał, że jedynym sposobem na uniknięcie wszelkiego możliwego kodu wirusowego jest izolacja. Chociaż istnieje wiele metod wykrywania złośliwego kodu, żadna technika nie działa na każdą odmianę ani w każdych okolicznościach, i zawsze istnieje jakaś metoda skutecznego unikania wykrycia

WYKRYWANIE ZŁOŚLIWEGO KODU NA PODSTAWIE SYGNATUR

Niektóre z najstarszych metod wykrywania złośliwego kodu to metody oparte na sygnaturach, które wykorzystują znane ciągi znaków lub wzorce w kodzie. Metody oparte na sygnaturach są łatwe do wdrożenia i nakładają na system bardzo niskie koszty ogólne, ale równie łatwo można ich uniknąć. Polimorfizm i metamorfizm to dwie metody, za pomocą których złośliwe oprogramowanie może zmieniać formę w czasie, unikając w ten sposób wykrywania opartego na sygnaturach. Systemy wykrywania mogą używać funkcji skrótu do pobierania odcisków palców danego programu binarnego lub fragmentu kodu. Jednak skrót nie będzie pasował do żadnej zmodyfikowanej wersji, co czyni tę metodę niezawodną, ale nie zawsze przydatną. Ogólnie rzecz biorąc, metody oparte na podpisach nie są strasznie niezawodne, chociaż podpisy są jedną przydatną miarą w bardziej złożonej heurystyce.

SIECIOWE WYKRYWANIE ZŁOŚLIWEGO KODU

Sieciowe metody wykrywania złośliwego kodu szukają artefaktów sieciowych związanych ze złośliwym kodem, takich jak połączenie z serwerem, jak w przypadku trojana rejestrującego keylogera lub bota

Internet Relay Chat (IRC). Wykrywanie anomalii sieciowych działa dobrze, ale jest drogie i źle zrozumiane przez wielu praktyków bezpieczeństwa. Prostą metodą wykrywania w sieci jest analiza danych przepływu sieci (netflow) w porównaniu ze statystyczną bazą danych o znanym dobrym ruchu. Jednak złośliwy kod działający normalnie może ominąć takie metody wykrywania. Przewidywanie, co jest normalne w danym środowisku, może być trudne dla osoby z zewnątrz, ale analiza niektórych działań, takich jak system nazw domen, poczta e-mail i korzystanie z Internetu, może stworzyć ogólne modele unikania.

WYKRYWANIE BEHAWIORALNEGO ZŁOŚLIWEGO KODU

Behawioralne metody wykrywania złośliwego kodu analizują działania uruchomionego oprogramowania w poszukiwaniu nielegalnych działań. Może to być otwarcie portu, połączenie ze zdalnym hostem lub modyfikacja tablicy wywołań systemowych lub innych obszarów pamięci. Behawioralne metody wykrywania zawiodą, jeśli złośliwe oprogramowanie działa normalnie lub jeśli złośliwy kod może atakować i wykorzystywać sam system wykrywania. Powolną, ale skuteczną metodą wykrywania potencjalnie złośliwego kodu jest użycie podejścia opartego na maszynie wirtualnej, dzięki czemu system pozwala na wykonanie kodu na maszynie wirtualnej w trybie piaskownicy. Pozwala to na pełną funkcjonalną analizę kodu, ale obecna przepustowość takich systemów sprawia, że nie są one przydatne w dzisiejszych szybkich środowiskach produkcyjnych.

HEURYSTYCZNE WYKRYWANIE ZŁOŚLIWEGO KODU

Bardziej złożone heurystyki mogą wykorzystywać zarówno modele statystyczne, jak i behawioralne, aby określić względny wynik w stosunku do normalnego korpusu (w tym przypadku statystycznej bazy danych) uzasadnionego zachowania. Analiza bayesowska jest szeroko stosowana w wykrywaniu spamu. Ta metoda może jednak wykrywać nowe warianty istniejącego złośliwego kodu z dużą dokładnością. Analiza N-gram jest formą analizy częstotliwości zapożyczoną z przetwarzania w języku naturalnym, która może modelować oprogramowanie i typy danych odcisków palców. Ta metoda jest przydatna do wykrywania kodu wykonywalnego osadzonego w innych obiektach danych. Krótko mówiąc, nie ma jednego monolitycznego sposobu na wykrycie całego złośliwego kodu. Spinellis wykazał, że niezawodne wykrywanie złośliwego kodu jest NP-kompletne, co oznacza, że dylematu nie da się rozwiązać w czasie wielomianowym. Jednak omawiane technologie, stosowane wspólnie, są stosunkowo skuteczne w wykrywaniu wielu typowych zagrożeń złośliwym kodem.

ZAPOBIEGANIE SZKODLIWYM ATAKOM KODOWYM

OBRONA W GŁĘBI.

Ponieważ problem złośliwego kodu jest w sposób oczywisty kompletny NP (nierozwiązywalny), pojedynczy program antywirusowy nie może chronić się przed wszystkimi zagrożeniami złośliwym kodem. Jedna strategia, zwana głęboką obroną, wykorzystuje kontrole operacyjne, ludzkie i techniczne. Inną strategią jest tworzenie sieci i aplikacji, które działają tylko w jeden prawidłowy sposób. Takie ortogonalne sieci i aplikacje są rzadkością, ponieważ są drogie i trudne do zaprojektowania, a wiele firm będzie opierać się nałożeniu sztywnych ograniczeń na przedsiębiorstwo.

KONTROLA OPERACYJNA ZŁOŚLIWGO KODU

Wszystkie organizacje muszą stworzyć pisemne zasady i procedury dotyczące wprowadzania kodu programu do środowiska operacyjnego. Polityki powinny określać, jakie osoby firma pozwala na instalowanie programów, akceptowalne wykorzystanie dostępu do Internetu, akceptowalne wykorzystanie w systemach e-mail i co zrobić, jeśli użytkownicy podejrzewają kompromis systemu lub użytkownika. Organizacje powinny poddać wszystkich nowych pracowników pewnym dochodzeniom.

Powinno to obejmować przynajmniej przeszukiwanie rejestrów karnych, weryfikację wszystkich referencji i danych uwierzytelniających oraz raport kredytowy. Pracodawcy powinni zadbać o to, aby potencjalni pracownicy byli szczerzy i zgodni z prawdą. Jeśli pracownicy kłamią przed zatrudnieniem, istnieje prawdopodobieństwo, że również będą kłamać później.

KONTROLA LUDZI POD KĄTEM ZŁOŚLIWEGO KODU

Wszyscy użytkownicy (w tym kierownictwo) powinni przejść szkolenie w zakresie zasad i procedur organizacji. Ze względu na zmieniający się charakter zagrożenia złośliwym kodem organizacja powinna co najmniej aktualizować i odświeżać to szkolenie co roku. Sesje szkoleniowe powinny zawierać informacje o typowych zagrożeniach, sposobie ich wykrywania i właściwej reakcji. Obecnie szkolenie to powinno obejmować identyfikację oszustw zaliczkowych (znanych również jako oszustwa nigeryjskie 419), prób socjotechniki oraz wykrywania złośliwych załączników. Użytkownicy powinni powiadomić dział pomocy technicznej lub inny podmiot, jeśli napotkają jakiekolwiek nietypowe zachowanie systemu lub użytkownika.

WDRAŻANIE SYSTEMÓW ANTYWIRUSOWYCH

Wdróż rozwiązanie antywirusowe (A / V), które pasuje do środowiska operacyjnego. To rozwiązanie powinno obejmować zarówno systemy sieciowe, jak i hostowe. Systemy sieciowe działają wbudowane jak brama, podczas gdy systemy oparte na hoście działają w punktach końcowych hosta. Systemy te powinny pochodzić od różnych dostawców, ponieważ korzystanie z tego samego oprogramowania w sieci i hostach jest mniej korzystne. Zróżnicowana strategia wykrywania i ograniczania pomoże firmom uniknąć pułapek związanych z aktywnym wykorzystywaniem złośliwego oprogramowania antywirusowego, co miało miejsce w 2004 r. I ponownie w 2006 r. Ostatnią rzeczą, na którą organizacje powinny pozwolić, projektując strategię A / V, jest oprogramowanie antywirusowe jako wektor dla aktywnego ataku złośliwego kodu. Rozwiązanie A / V powinno zapewniać mechanizm do dynamicznego stosowania aktualizacji i powinno to robić przynajmniej codziennie. Systemy poczty e-mail mogą wymagać oddzielnego wbudowanego urządzenia do wykrywania złośliwego kodu przenoszonego przez pocztę e-mail, a także spamu, oszustw i ataków phishingowych. Stosując różnorodne podejścia do wykrywania, organizacje mogą osiągnąć wyższe wskaźniki wykrywalności niż przy użyciu pojedynczego, niezależnego produktu.

KONTROLA KONFIGURACJI HOSTA I BEZPIECZEŃSTWO

Konfiguracja hosta może złagodzić wiele zagrożeń złośliwym oprogramowaniem przed jego pojawieniem się. Zaimplementuj formę automatycznych aktualizacji (poprawek itp.), Która obsługuje środowisko operacyjne. Wiele technicznych zagrożeń złośliwym kodem (np. Robaki) atakuje dobrze znane i załatane wady. Wyeliminuj całe niekrytyczne oprogramowanie i usługi. Pomoże to zminimalizować zagrożenia skierowane na wciąż rosnącą złożoność kodu, z którą zmagają się specjaliści od bezpieczeństwa. W jednym badaniu programiści przeszkoleni w zakresie modelu dojrzałości zdolności do bezpiecznego tworzenia oprogramowania nadal popełnili 4,5 błędów na 1000 wierszy kodu 52. W przypadku systemu operacyjnego takiego jak Microsoft Vista, który według szacunków zawiera 50 milionów wierszy kodu, ekstrapolacja tej statystyki oznacza, że prawdopodobnie występuje co najmniej 225 000 błędów kodu. Wyłączenie lub usunięcie jak największej części tego kodu jest rozsądną kontrolą zapobiegawczą. Jeśli pozwala na to środowisko, usuń wszystkie przeglądarki internetowe. Jeśli środowisko nie pozwala na to, zablokuj konfigurację przeglądarki i użyj bezpiecznego serwera proxy sieci Web.

KONTROLA BEZPIECZEŃSTWA OPARTA NA SIECI

Warstwowa obrona routerów, zapór ogniowych, serwerów proxy i przełączanych wirtualnych sieci lokalnych (VLAN) może ograniczyć rozprzestrzenianie się złośliwego kodu. Na routerze odfiltruj wszystkie przychodzące fałszywe adresy sieciowe (BOGON), adresy RFC) 1918 i sfałszowane adresy wewnętrzne według RFC 226754 i 3704. Jeśli przedsiębiorstwo stoi w obliczu określonych zagrożeń od niektórych narodów, odfiltruj bloki sieciowe przydzielone tym narodom na routerze granicznym. Skorzystaj z aktualnych najlepszych praktyk w zakresie konfiguracji zapory. (Zmieniają się one szybko.) Użyj bezpiecznego, uwierzytelnionego serwera proxy sieci Web i zmuś wszystkich klientów do korzystania z serwera proxy. (Żaden użytkownik nie powinien mieć bezpośredniego dostępu do sieci.) Wyłącz wszystkie nieużywane porty dostępu do sieci LAN lub rozważ użycie uwierzytelnienia 802.1x, aby zrobić to automatycznie. Podziel sieć na funkcjonalne grupy robocze za pomocą sieci VLAN lub fizycznie oddzielnej architektury przełączników. Jeśli to możliwe, zaimplementuj listy kontroli dostępu między sieciami VLAN lub przełączanymi segmentami. Celem jest, aby sieć była jak najbardziej ortogonalna. Jeśli sieć może funkcjonować tylko w jeden właściwy sposób, organizacja całkowicie uniknie wielu automatycznych zagrożeń złośliwym kodem.

MONITOROWANIE SIECI

Zapobieganie zagrożeniom ze strony złośliwego kodu jest idealne, ale kluczowe znaczenie ma wykrywanie. Większość ataków złośliwego kodu będzie miała pewien artefakt, zarówno oparty na hoście, jak i na sieci. Aby wykryć te artefakty, organizacje powinny ustanowić system zarządzania informacjami o bezpieczeństwie, który gromadzi dzienniki urządzeń, dzienniki serwerów, dzienniki hostów, alerty systemu wykrywania włamań i dane o przepływie sieci. Dane o przepływie sieci są użytecznym narzędziem do wykrywania nieprawidłowej aktywności sieci. Przepływ sieci to 5-krotek składający się z adresu źródłowego, adresu docelowego, portu źródłowego, portu docelowego i protokołu z powiązanym znacznikiem czasu. Jakakolwiek złośliwa aktywność sieciowa będzie związana z tym przepływem. Podstępne złośliwe oprogramowanie będzie jednak próbowało działać normalnie, aby uniknąć wykrycia. Wykrywanie nieprawidłowej aktywności wymaga od operatorów zrozumienia, co jest normalne dla środowiska. Bez historycznej bazy danych statystycznych lub dużego doświadczenia może to być trudne. Detekcja anomalii sieci (NAD) wykorzystuje metodologię modelowania statystycznego w celu wykorzystania tych danych do wykrywania odstających wartości statystycznych. To jest wyjątkowo skuteczny w wykrywaniu nowych zagrożeń złośliwym kodem, a także innych form anomalnych zachowań. Jednak brak zrozumienia wydaje się ograniczać przyjęcie NAD. Jest to obszar, w którym wiele organizacji może poprawić swoją praktykę kontrola bezpieczeństwa informacji.

WNIOSEK

Zagrożenia złośliwym kodem są tak liczne, jak różnorodność złośliwego kodu. Zapobieganie wszelkim złośliwym kodom nie jest możliwe, ponieważ problem jest w sposób oczywisty NP-zupełny.57 Jednak strategia dogłębnej obrony wykorzystująca kontrole operacyjne, ludzkie i techniczne może być stosunkowo skuteczna. Właściwie zastosowana obecna generacja kontroli technicznych dostępnych dla organizacji skutecznie powstrzymuje większość zagrożeń złośliwym kodem. Jednak zaufany informator zwykle ma dostęp do tego, aby zmienić zagrożenie w rzeczywistość ryzyka operacyjnego. Obecne trendy w zakresie zagrożeń złośliwym kodem wskazują na ciągły wzmożony udział przestępczości zorganizowanej i szpiegostwo międzynarodowe, które nadal atakują najsłabsze ogniwo łańcucha bezpieczeństwa: człowieka.