PORADNIKI

Omijanie firewall'i : Narzędzia i Techniki

1.Wprowadzenie

Firewall'e są uważane jako podstawowy mechanizm obronny kiedy łączymy się z Internetem. Są różne typy, ale firewall' zasadniczo kontrolują dostęp przez warstwę aplikacji i transportu, często używając informacji warstwy aplikacji dla podejmowania decyzji o kontroli dostępu. Firewall ma zatrzymywać niechciany ruch z Internetu do chronionej sieci. W podobny sposób, możemy kontrolować ruch przepływający na zewnątrz do Internetu. Firewall jest granicznym mechanizmem zabezpieczeń. Musi sterować tym co pojawia się w wewnętrznej sieci. To prosty obraz połączeń między wnętrzem a stroną zewnętrzną. Jedyna informacja jaką ma firewall o poszczególnych połączeniach ,to źródłowe i docelowe adresy i numery portów. Jest bardzo łatwo oszukać firewall zezwalając na protoków, który przypuszczalnie będzie zablokowany – albo przez zmianę numeru portu związanego z tym protokołem lub użycie tunelowania protokołu. Wprowadzimy tu pojęcie protokołu tunelowania i pokażemy jak łatwo można użyć go do omijania firewall'a

2. Protokół tunelowania

Protokół tunelu zawiera jeden protokół w drugim. Tunelowanie jest ogólną techniką która może być używana do przenoszenia protokołu przez obcą sieć. Jest często używane do połączenia dwóch izolowanych sieci z prywatnym *mostkiem* przez sieć publiczną, formując VPN (Virtual Private Network) .Najpopularniejszy ruch IP jest szyfrowany i umieszczany w strumieniu TCP, który jest przenoszony przez publiczny Internet między dwoma zdalnymi stronami. Dla tunelowania może być wykorzystany dowolny protokół. Jedynym wymaganiem jest to, aby prorokół był dozwolony przez dowolny firewall, który znajduje się między tunelema punktami. Protokoły takie jak SMTP i HTTP generalnie spełniają to wymaganie. Inne, takie jak ICPM-ECHO (protokół "ping") również są dozwolone przez większość konfiguracji.

2.1 Omijanie firewall'i

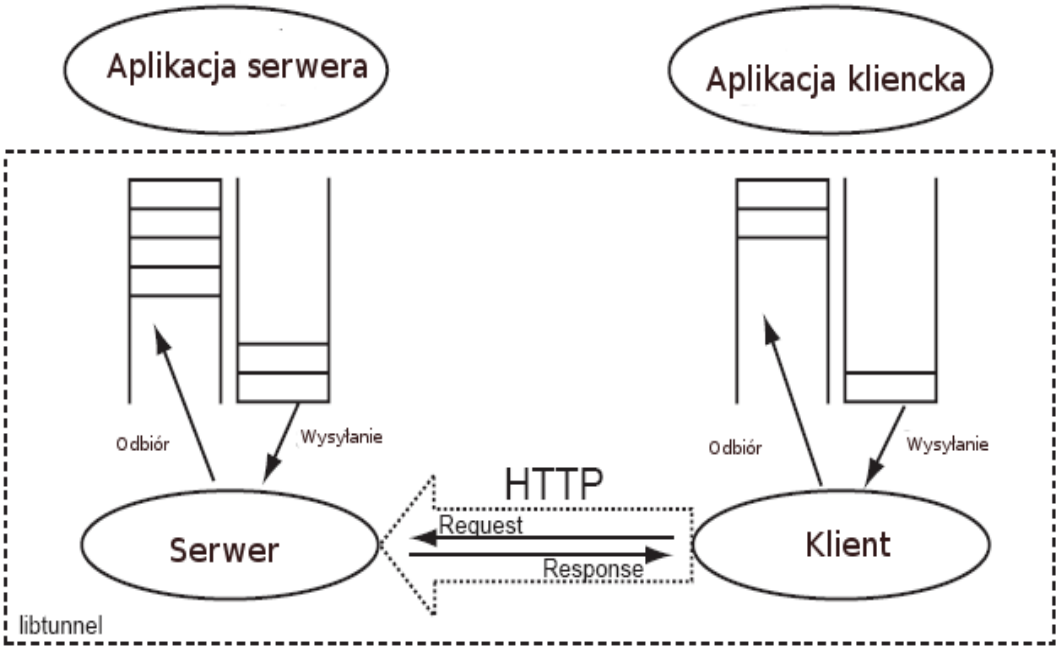
Protokół tunelowania może odwrócić protoków warstwy aplikacji (takie jak HTTP lub SMTP) do protokołu warstwy transportowej. To może utrudnić pracę firewall'owi przez który przechodzi cały ruch. Jednym z narzędzi stworzonym do wykorzystania tego faktu jest httptunnel, któy jest dostępny jako Publiczna Licencja GNU. Narzędzie to tworzy dwupunktowy tunel HTTP, ale może być użyte w połączeniu z innym oprogramowaniem (takim jak SSH i Telnet) aby uzyskać nieograniczony dostęp przez firewall. Użytkownicy po stronie z ograniczeniami firewall'a mogą wyłączyć protokoły, które są blokowane ,przez tunelowanie ich przez HTTP. Oczywisty problem powodowany przez to narzędzie jest taki ,że zasady firewall'a już dłużej nie dyktują zasad bezpieczeństwa. Chociaż użytkownicy muszą podjąć świadomą decyzję dla odwołania aplikacji takich jal htc i hts, i tak użytkownicy ostatecznie decydują jaki określony protokół będzie przechodził przez firewall.

2.2 Prosty tunel

Możliwe są dużo bardziej nikczemne scenariusze, takie jak zademonstrowane przez BT Laboratories. Simple Tunnel jest innym tunelem ogólnego przeznaczenia zaprojektowanym niezależnie od GNU httptunnel ,ale mniej więcej w tym samym czasie .Został zbudowany jako część demonstratora, stworzonego do podkreślenia wątku którego tunelowanie stanowi bezpieczeństwo sieciowe. Podobnie jak GNU httptunnel, Simple Tunnel również używa HTTP jako środka transportu, chociaż może byćłatwo rozszerzony na prawie każdy protokół klient – serwer. Jednak, jest spakowany jako biblioteka a nie jako aplikacja – zaprojektowany aby być wbudowanym w inne apliakcje. Ta biblioteka, nazwana libtunnel, dostarcza kanału dla przekaywania arbitralnych komunikatów między punktami końcowymi tunelu. Ten system komunikacji może być użyty w bibliotekach wysokopoziomowych i apliakcjach, dla komunikowania się ze zdalnym hostem. Działanie Simply Tnnel jest pokazane poniżej na rysunku. Klient i serwer kolejkują komunikaty przy każdym końcu tunelu. Klient tworzy okresowe połączenia z serwerem . Połączenia te są typu HTTP. Algorytm używany przez klienta jest

następujący:

1. Jeśli klient ma komunikaty do wysłania, tworzy żądanie HTTP POST do serwera. Komunikaty są kodowane i wysyłane w ciele tego żądania. W przeciwnym razie
2. jeśli klien nie ma komunikatów do wysłania, tworzy żądanie HTTP GET do serwera
3. Jeśli serwer ma jakieś komunikaty do wysłania, są one kodowane i zwracae w ciele tej odpowiedzi



Klient może być skonfigurowany do użycia proxy, które pozwala działać przez bramkę warstwy apliakcji. Dodatkowo, serwer może zarządzać tunelami wielu klientów jednocześnie. Apliakcje przy każym końcu tunelu używają prostego API do odczytu i zapisu komunikatów i również do kontroli kilku charakterystyk tuneli. Biblioteka Simple Tunnel używana była w kilku przykładowych apliakcjach. Jedną z nich jest biblioteka zdalnego gniazda, nazwanej librsocket. Używa ona komunikatów dostarczanych przez libtunnel do systemu tunelowania wywołanego sieciowy stosu klienta. Librsocket API jest bardzo podobna do standardowego gniazda API. Została wybrana ponieważ jest raczej prosta, bardzo użyteczna i zaimplementowana bardzo szerokim zakresie na platformach. Ale prawie każde API na dowolnej maszynie może być tunelowany w ten sam sposób. Ta technika daje atakującemu pełny dostęp do zasobów zdalnego hosta. Podczas gdy protokół tunelowania jest bardzo silnym narzędziem używanym przeciwko firewall'om atakujący z zewnątrz musi zainstalować oprogramowanie klenckie na maszynie za firewall'em. Szczęściem dla atakującego jest wiele sposobów na dokonanie tego, albo przez wykorzystanie słabości oprogramowania, lub ataki social-engineeringu. Oto kilka przykładów;

Zdalne wykorzystanie. Zdalnie wykorzystany błąd w programie, który pozwala atakującemu wykonać kod, może być użyte do załadowania instalacji tunelu klienta.

Trojany. Użytkownik może być zmuszony do uruchomienia tunelu klienta, sądząc ,że jest to coś innego

Infekcja wirusem. Wirus który może zainfekować programy wykonywalne może zawierać tunel klienta. Nawet wirus makro może być stworzony do przeniesienia tunelu klienta, jeśli makro ma

dostęp do funkcji sieciowych.

Lista tych ataków oczywiście nie jest wyczerpana, ale ilustruje fakt ,że zainstalowanie podstępnego oprogramowania klienckiego nie szczególnie trudne. Dla zademonstorwania ataku Konia Trojańskiego, librsocket została użyta do zbudowania specjalnej wersji przeglądarki Mozilla. W wersji Trojana na Macintosha, zmiana wymagała dodatkowej pojedycznej lini kodu w źródle Mozilli i ponwnego połączenia z librsocket .Wynikowa przeglądarka zachowywała się jak normalny sieciowy klient, z wyjątkiem tego,że wykonywała regularne połączenia z tunelem serwera. Ilekroć użytkownik uruchamiał przeglądarką z Trojanem, nieświadomie otwierał sieciowy stos swojej maszyny do wykorzystania przez "atakującego"

3. W stronę dogłębnej obrony

Powidziano ,że całkowite bezpieczeństwo systemu jest takie jak zabezpieczenie najsłabszego komponentu w systemie. Implikuje to to ,że słaby komponent w izolacji jest mniejszym probleme niż kiedy jest integrowany jako częśćsystemu, który zależy od zabezpieczenia słabego komponentu. Firewall może być ominięty jeśli host od strony użytkownika może być zagrożony. Sieć niezabezpieczonych systemów może nie mieć swoich zasad bezpieczeństwa wprowadzonych przez sam firewall. Ale problemem nie są możliwości firewalla, jest to zabezpeiczenie systemu za samym firewall'em. Startegiaa "dogłębnej obrony" jest krytyczna dla prewnecji przed atakami. Termin ten osiągnął popularność niedawno i często używa sięgo w odniesieniu do wykrywania wirusów lub intuzów. Ale prawdziwa obrona powinna chronić również przeciwko tym atakom. Poniżej rozpatrzymy rodzaje "obrony dogłębnej" wymaganej do prewencji przed atakami tunelowania.

3.1 Domeny

Carroll i Landwehr argumentują ,że kluczową charakterystyką bezpiecznych systemów jest ich możliwość do zarządzania domenami dla przechowywania u przetwarzania. Domena jest zbiorem informacji i autoryzowaną do ich wykorzystywania. Pojęcie to wywodzi się z zabezpieczeń wojskowych. Nie jest krytyczne jak domeny są zarządzane ale to jak mogą być chronione przed wtrącaniem się jedna w drugą i z interakcjami między nimi w sposób który mógłby naruszyć zasady bezpieczeństwa. Mechanizm obsługujący "zarządzanie domenami" może być implementowany zarówno sprzętowo jak i programowo. Sterowanie programowe możę istnieć zarówno na poziomie systemu jak i wewnątrz apliakcji. To co jest krytyczne to to czy sterowanie działa poprawnie. Może się to wydaje oczywiste – ale dowolna wada w mechanizmach niskopoziomowych może całkowicie podkopać to na wyższym poziomie. Są różne przykłady jak te pomysły mogą być stosowane w systemach sieciowych. Dalton i Clarke sugerowali jak zapewnić bezpieczny dostęp do serwisów LAN z hostów Internetowych. Choo pokazał jak rozszerzyć ten pomysł, przez segregowanie sieci wewnętrznych w VPN'y uruchmiających się na różnych poziomach wrażliwości. Te i inne schematy dostarczają ogromnego zastosowania zaufanych systemów, takich jak CMW (Compartemend Mode Workstation). CMW różni się od "konwencjonalnych" stacji roboczych na kilka sposobów:

Znakowanie informacji. Wszystkie informacje są przechowywane w segmentach z dołączoną etykieta określa wrażliwość informacji (np SECRET, TOP SECRET) jak również jej przedział (NUCLEAR, COMMAND). CMW zapewnia ,że etykiety nie mogą być modyfikowane bez poprawnej autoryzacji i ,ze są one przenoszone z informacją poprzez sieć.

Kontrola dostępu. Dodatkowo do normalnego pojęcia uprawnienia, znanego jako DAC (Discretionary Access Control), CMW wymusza jeszcze MAC (Mandatory Access Control). Jest to niezmienna zasada bezpieczeństwa używana przz wojsko do kontroli wrażliwych informacji. Zasada MAC nie może być zmieniana lub nadpisywana.

Uprzywilejowanie i uwiarygadnianie

Zamiast posiadać pojedynczego "super – użytkownika", CMW ma zbiór uprzywilejowań i

uwairygodnień, które mogą być przypisane do pojedynczego użytkownika i programu. W systamch Unix,zwykłe programy takie jak ping i xterm muszą uruchamiać się jako root ponieważ wykonują uprzywilejowane operacje. Program ping, na przykład, musi otworzyć surowe gniazdo sieciowe aby wysłać datagram ICMP. W CMW, ping tylko wymaga uprzywilejowania dla otwarcia surowego gniazda. Uwiarygadnianie jest pojęciem, które daje użytkownikom możliwość uruchamiania pewnych niebezpiecznych programów. Każdy użytkownik może uruchomić program ping, ale tylko użytkownik z poprawną autoryzacją może zamontować lub odmontować system plików.Autoryzacja pozwala na administrowanie systemem różnym użytkownikom, nie tylko tym ,którzy mają pełną kontrolę nad systemem.

Kontrola funkcji. Logowanie dostarcza sposobu na skontrolowanie operacji systemu. Jądro CMW loguje cały system funkcji, które wykonują jego przetwarzanie. To logowanie nie może być wyłączone. Dodatkowo, CMW dostarcza sposobu apliakcji na logowanie swoich własnych akcji w bezpieczny sposób.

Język programowania Java, zorientowany obiektowo jest inną technologią obsługującą "zarządzanie domenami". Obecnie ta czynność jest dokonywana wewnątrz jednej apliakcji – Java Virtual Machine (JVM). Z tego powodu, środowisko Javy jest wrażliwe na błędy w architekturze podstawowej maszyny. Obiekty Javy są chronione przed wzajemną interferencją ,ale bardzo często błąd w apliakcji, który współistnieje z JVM, może interferować z obiektami Javy. Zatem bezpieczeństwo w Javie zależy jeszcze od zafunaego podstawowego systemu.

3.2 Domeny sieciowe

Bezpieczne systemy sieciowe muszą mieć możliwość zarządzania domenami , które obejmują wiele hostów. Choo propnuje rozwiązanie IPSec dla segmentowania wewnętrznych sieci. Opisuje on użycie technologii CMW dla wymuszenia tej segregacji przez uruchamianie wielu stosów IPSec, przynajmniej jeden dla każdego wrażliwego poziomu. W jego schemacie IPSec uruchamia się jako proces obszaru użytkownika, który chroni przed błędami stosu wpływającymi zarówno na inne wrażliwe poziomy jak i jądro. Zauważmy jednak ,że pojedynczy demon ISAKMP jest zaufany dla wszyskich procesów IPSec. Czyni to cały schemat bezbronnym wobec ataków kluczem,jeśli demon może być osłabiony. ISAKMP jest prawdopodobnie zbyt złożony aby uruchamiać się jako zaufany serwis, chociaż trudno jest zobaczyć jak może uruchomić się w tym schemacie.

3.3 Bump in the wire

Powyższe pomysły mogą być rozszerzone również o pozostałe systemy. Podejście to implementuje połączenia sieciowe jako bump-in-the-wire dla hostów niezaufanych. BITW zastępuje wbudowane połączenia sieciowe z zaufanymi dołączanymi wersjami, zaimplementowanymi w firmware na specjalnej karcie sieciowej. Karta ta jest faktycznie malusieńkim komputerem, uruchamiającym swój własny zaufany stos sieciowy .Zamiast pełnego stosu sieciowego, niezaufane hosty mają zbiór prostych funkcji wywołujących określone usługi na tej karcie. Cały kluczowy materiały jest podtrzymywany przez kartę i nigdy nie jest przechowywany w głównej pamięci komputera. System może System może zachować wrażliwe etykiety przez hosty. Demon ISAKMP zakłada wrażliwość końcowych punktów gniazda komunikacyjnego, więc etykiety nie muszą być dołączane do pakietów przy przesyłaniu Graniczne warunki dla użytkownika i hosta są określane przez PKI. Hosty które nie mają możliwości obsłużenia waraźliwych informacji są zabezpieczane przed zrobieniem tego przez stos sieciowy – poprzez osadzenie go na karcie może to być zabezpieczone w wysokim stopniu. BITW chroni jądro systemu operacyjnego przed błędami sieciowych połączeń. Sam host jest jeszcze wrażliwy na błędy w oprogramowaniu na nim uruchomionym – ważna jest też możliwość zarządzania domenami. Ale wpływ błędów w oprogramowaniu może być powstrzymany przez zasadę bump-in-the-wire. Jest to skuteczna izolacja od reszty maszyny. Nawet w trybie jądra, maszyna może tylko komunikować się z kartą przez bardzo ograniczony interfejs. Powinna być możliwość konfiguracji i zarządzania kartą całkowicie ze strony sieciowej, z zaufanych hostów w

sieci.