STUDIA INFORMATICA 2002
Volume 23 Number 3 (50)

Jacek LACH
Politechnika Śląska, Instytut Informatyki

IDENTYFIKACJA SYSTEMU OPERACYJNEGO W TEŚCIE PENETRACYJNYM

Streszczenie. W artykule przedstawiony został przegląd technik umożliwiających zdalną, identyfikację systemu operacyjnego. Zaprezentowane zostały techniki wykorzystujące standardową konfigurację, jak również aktywnego oraz pasywnego odcisku palca. Wskazane zostały również sposoby unikania zagrożeń niesionych przez powyższe techniki.

OPERATING SYSTEM IDENTIFICATION IN PENETRATION TEST

Summary. This article describes methods used for system identification. Both methods abusing out-of-the-box configurations as well as active and passive fingerprinting were described. Some basic and more complex methods of fingerprinting evasion were also briefly presented.

1. Wstęp

Badanie zabezpieczeń systemu operacyjnego może być prowadzone na kilka sposobów. Jednym z bardziej skutecznych jest próba włamania do badanego systemu w ramach przeprowadzanego testu penetracyjnego. Rozpoznanie systemu operacyjnego oraz jego wersji jest jednym z etapów takiego testu, będąc jednocześnie jednym z pierwszych (zaraz po ustaleniu struktury sieci) kroków podejmowanych przez włamywacza. Ustalenie tych informacji umożliwia zaplanowanie kolejnych etapów działania oraz odrzucenie metod ataku nieskutecznych w przypadku rozpoznanego systemu.

2. Wykorzystanie błędów w konfiguracji

Pierwotnie metody rozpoznawania systemu opierały się na wykorzystaniu informacji podawanych przez oprogramowanie. Metoda ta jest bardzo prosta i w wielu przypadkach (niestety) nadal skuteczna. Próba połączenia się z systemem pracującym pod kontrolą starszej wersji systemu SunOS w jego standardowej konfiguracji za pomocą programu telnet spowoduje uzyskanie następującej informacji:

```
$ telnet 192.168.1.1
Trying 192.168.1.1...
Connected to hostname.domain.
Escape character is '^]'.

UNIX(r) System V Release 4.0 (hostname)

login:
```

Ta sama próba skierowana na serwer pracujący pod kontrolą systemu Linux umożliwia uzyskanie znacznie dokładniejszej informacji na temat działającego systemu:

```
$ telnet 192.168.1.1
Trying 192.168.1.2...
Connected to hostname.domain.
Escape character is '^]'.

Red Hat Linux release 6.2 (Zoot)
Kernel 2.2.19 on an i686
login:
```

Jak widać powyżej, można uzyskać nie tylko informację o rodzaju systemu operacyjnego (Linux), wiadomo też, że jest to dystrybucja RedHat, wersja 6.2, dodatkowo wiadomo, że było przeprowadzone uaktualnienie jądra systemu do wersji 2.2.19. Informacje te są dużo dokładniejsze od tych, które można uzyskać za pomocą różnych, często bardzo zaawansowanych technik odcisków palca (*ang. fingerprinting*). Może się zdarzyć, że informacje te nie są prawdziwe, a zamieszczone zostały tylko w celu zmylenia intruza, jednak takie sytuacje zdarzają się bardzo rzadko. Najczęściej informacje te są prawdziwe i wyświetlane po standardowej instalacji systemu. Powyższy sposób uzyskiwania dodatkowych informacji o systemie dotyczy również oprogramowania, często krytycznego z punktu widzenia działania systemu.

```
$ telnet 192.168.1.1 25
Trying 192.168.1.1...
Connected to hostname.domain.
Escape character is '^]'.
220 (none) ESMTP Sendmail 8.9.3+3.2W/8.9.3/Debian 8.9.3-21; Tue,
27 Nov 2001 10:34:23 +0100
```

Powyższy przykład pokazuje, jak łatwo dowiedzieć się, jaki program obsługuje pocztę na danym serwerze (sendmail) oraz jaka jest jego wersja (8.9.3). Dodatkowo można zauważyć, że działającym systemem operacyjnym jest najprawdopodobniej Linux - dystrybucja Debian.

3. Techniki „odcisku palca"

Dużo bardziej rozwiniętą techniką rozpoznania systemu operacyjnego jest badanie różnic w działaniu stosu TCP/IP różnych systemów. Technika ta nosi nazwę badania "odcisku palca" systemu operacyjnego (*ang. OS fingerprinting*). Omawiana technika może zostać wykonana w sposób aktywny bądź w sposób pasywny.

3.1. Techniki aktywne

Aktywna odmiana badania odcisku palca (*ang. active fingerprinting*) polega na wysyłaniu do badanego komputera odpowiednio spreparowanych pakietów w celu analizy pakietu zwrotnego. Technika ta została dokładnie opisana w [1], do zestawu podstawowych testów należą:

- Test FIN (*FIN Probe*). Test polega na zbadaniu zachowania zgodności z wytycznymi zawartymi w RFC793, wg którego po otrzymaniu pakietu FIN nie powinna pojawić się żadna odpowiedź, wiele systemów operacyjnych odpowiada w takiej sytuacji pakietem RST.
- Test nieprawidłowej flagi (*BOGUS Flag Probe*). Polega na ustawieniu niezdefiniowanej flagi w nagłówku TCP i sprawdzeniu odpowiedzi systemu na taki pakiet. Niektóre systemy odpowiadają bez kasowania nieprawidłowej flagi.
- Próbkowanie ISN (*TCP ISN Sampling*). Polega na znalezieniu sekwencji numerów ISN w ciągu pakietów wysyłanych podczas transmisji. Numery te mogą być (tak jak w starszych implementacjach) przydzielane sekwencyjnie lub zgodnie z pewnym algorytmem (mniej lub bardziej losowo).
- Test bitu fragmentacji (*Don't fragment bit*). Niektóre systemy standardowo ustawiają ten bit, co może być wykorzystane jako test pomocniczy.

- Test rozmiaru okna początkowego (*TCP Initial Window*). Polega na sprawdzeniu początkowego rozmiaru okna podczas przesyłania pakietów. Umożliwia stosunkowo dokładną identyfikację rodzaju systemu.
- Test wartości potwierdzenia (*ACK Value*). Polega na sprawdzaniu numeru potwierdzenia TCP po wysłaniu odpowiednio spreparowanego pakietu.
- Test obsługi fragmentów (*Fragmentation handling*). Polega na sprawdzeniu sposobu składania nachodzących na siebie fragmentów pakietu.
- Test opcji TCP. Polega na sprawdzeniu obsługiwanych opcji TCP.
- Reakcja na fałszowane pakiety SYN. Część systemów blokuje połączenia po nadejściu kilku sfałszowanych pakietów SYN. Po wysłaniu takiej sekwencji można sprawdzić poprawność połączenia z wybranym hostem.
- Test tłumienia komunikatów ICMP (*ICMP Error Message Quenching*). RFC 1812 sugeruje limitowanie wysyłania komunikatów ICMP o błędach. Niektóre systemy stosują się do zaleceń RFC, co umożliwia ich identyfikację.
- Cytowanie komunikatów ICMP (*ICMP Message Quoting*). Różne systemy zamieszczają w komunikacie ICMP dotyczącym błędu różną ilość bajtów pakietu powodującego błąd.
- Test integralności odesłanego pakietu (*ICMP Message Quoting Integrity*). Zawartość cytowanego pakietu w komunikacie ICMP może zostać zmodyfikowana, w zależności od sytemu zmianie mogą podlegać różne pola cytowanego pakietu.
- Test pola TOS. System Linux ustawia to pole w komunikacie ICMP port unreachable na wartość 0xC0, co umożliwia jego identyfikację.

Prace nad użyciem protokołu ICMP w zakresie skanowania zostały podjęte przez Ofira Arkina. Wyniki tych prac zostały szczegółowo opisane w [2]. Protokół ICMP opisany w RFC 792 (z rozszerzeniami w RFC 1122 i RFC 1812) jest integralną częścią protokołu IP, który jednak korzysta z niego, jakby był protokołem wyższego poziomu. Został utworzony w celu przesyłania informacji dotyczących problemów z siecią. Nie jest protokołem zapewniającym dotarcie komunikatu do odbiorcy, co należy wziąć pod uwagę przy wykorzystaniu go do skanowania. Narzędzie powstałe w wyniku prac nad protokołem ICMP umożliwia bardzo szybką identyfikację docelowego systemu z dużą dokładnością. Identyfikacja docelowego systemu operacyjnego następuje po wysłaniu już 1 do 4 pakietów, co odróżnia tę metodę od wcześniej przedstawionej generującej większy ruch. Technika ta, jak poprzednia, bazuje na zróżnicowaniu reakcji na otrzymywane datagramy ICMP. Wykorzystywane są datagramy *ECHO*, *Timestamp*, sposób obsługi fragmentów. Odpowiednio skonstruowana hierarchia zapytań zaimplementowana w programie będącym wynikiem prac badawczych umożliwia szybką i skuteczną identyfikację systemu. Dokładniejszy opis znajduje się we wspomnianej

już pracy [2], a opis implementacji w [3]. Użycie datagramów ICMP można ograniczyć poprzez wprowadzenie odpowiednich reguł w urządzeniach filtrujących. Ruch datagramów ICMP *ECHO Request/ECHO Replay* można blokować na poziome hosta bądź firewalla (zalecane drugie rozwiązanie z racji zmniejszenia ruchu wewnątrz sieci). Rozpoznanie systemu za pomocą badania obsługi fragmentów (Solaris, HP) można blokować poprzez blokadę ustawiania bitu DF w odpowiedziach ICMP. Podobne działanie można zastosować przy metodzie wykorzystującej *Path MTU Discovery,* wyłączenie może mieć jednak wpływ na wydajność systemu. Rozpoznanie systemu bazujące na ustawieniu wartości TTL pakietu może być udaremnione poprzez modyfikacje tej wartości w zabezpieczanym systemie.

Istnieje wiele innych metod testowania sytemu wykorzystujących inne właściwości protokołu ICMP bądź też badanie reakcji systemu na nietypowe sformułowane pakiety ICMP, jednak ich opis wykracza poza zakres tego artykułu. Warto jednak podkreślić, że z punktu widzenia bezpieczeństwa protokół ICMP może oddać nieocenione usługi potencjalnemu włamywaczowi.

3.2. Techniki pasywne

Techniki aktywnego odcisku palca są technikami skutecznymi jednak bardzo "głośnymi". Do identyfikacji systemu konieczne jest wysłanie do niego pewnej liczby komunikatów, co może zostać zarejestrowane lub uniemożliwione przez dobrze zabezpieczony system. Istnieje również technika mniej inwazyjna nazywana pasywnym odciskiem palca (*ang. passive OS fingerprinting*). Metoda pasywna bazuje na analizie informacji pochodzącej z badanej sieci. W znacznym stopniu utrudnia to wykrycie takiego działania, ponieważ do badanego systemu nie docierają żadne informacje z komputera atakującego. Wykorzystanie ruchu istniejącego w sieci jest jednocześnie wadą metody pasywnej, ponieważ nie zawsze możliwy jest dostęp do informacji umożliwiającej identyfikację systemu w sposób najskuteczniejszy. Jakość uzyskanych informacji tą metodą zależna jest od położenia urządzenia skanującego w sieci. Największe zagrożenie stanowi oczywiście skaner umieszczony w tym samym segmencie sieci co skanowany komputer. Metoda pasywna wykazuje duże podobieństwo do systemów detekcji włamań. Oba rozwiązania analizują istniejący ruch sieciowy oraz dopasowując pewne napotkane wzorce, dokonują ich identyfikacji. Sposób oraz cel zastosowania obu rozwiązań jest jednak całkowicie odmienny. Metoda pasywna może być zastosowana na dwóch poziomach: na poziomie sieci oraz na poziomie aplikacji. Przypadek pierwszy analogiczny jest do wcześniej omówionej techniki aktywnego odcisku palca. Wykorzystuje się jednak tutaj pakiety wędrujące w sieci, a nie te, których pojawienie się zostało w pewien sposób wymuszone. Metoda ta jest jednak bardzo skuteczna, ponieważ analizie podlega tak duża ilość informacji, że nawet zmiana typowych ustawień przez świadomego administratora

może okazać się nieskuteczna. Metoda pasywna musi oprzeć się na informacjach, które uzyskuje, a nie tych,które chciałaby uzyskać, co prowadzi do nieco innego schematu działania - wykorzystując jednak te same reguły, o których była mowa wcześniej. Dodatkową niedogodnością tej metody jest czasochłonność; gromadzenie potrzebnych informacji może trwać dużo dłużej niż w przypadku ataku aktywnego.

Niezwykle prostą metodą pasywnego rozpoznania systemu jest metoda działająca w warstwie aplikacji. Metoda ta opiera się na wyłuskiwaniu informacji dołączanych do danych przesyłanych przez sieciowe aplikacje klienckie. Metoda przedstawiona została przez Jose Nazario w artykule [4]. Opiera się ona na (najczęściej) nadmiarowej informacji dostarczanej przez oprogramowanie klienta przy jego normalnym działaniu. Jednym z bogatych źródeł informacji jest poczta elektroniczna. Niechlubnym przykładem jest popularny program Pine, dostarczający wraz z przesyłką informacji dotyczących systemu, na którym działa. W przykładzie poniżej jest to HP-UX 10.x (dokładnie 10.20):

```
Message-ID: <Pine.HPX.4.44.0203291854390.4201-100000@host.domain>
```

Oczywiście wiele innych programów również dostarcza takich informacji w mniej lub bardziej skuteczny sposób umożliwiając identyfikację systemu:

```
X-mailer: Pegasus Mail v3.50 (NDS)
X-Mailer: KMail [version 1.3.1]
X-Mailer: The Bat! (v1.53d)
```

Podobny nadmiar informacji prezentuje oprogramowanie obsługi grup news. Nagłówki postaci:

```
X-Newsreader: Yanoff 1.5.4 PalmPilot
X-Newsreader: Microsoft Outlook Express 5.00.2919.6600
X-Mailer: Mozilla 4.76 [en] (Win98; U)
```

są standardem, jednak zgodnie ze specyfikacją NNTP (RFC 2980) żaden z nich nie jest wymagany. Kolejnym zagrożeniem wymienianym we wspomnianym artykule jest analiza ruchu generowanego przez przeglądarki WWW. Również w ich przypadku dostarczane są informacje dotyczące zarówno programu klienta, jak i systemu operacyjnego, na którym klient działa. O ile w tym przypadku przesyłanie informacji o rodzaju oraz wersji klienta może być konieczne ze względu na zróżnicowanie możliwości poszczególnych klientów, o tyle informacja o systemie operacyjnym, pod kontrolą którego klient pracuje, jest już nadmiarowa.

4. Zmiana tożsamości

Niezwykle ciekawą metodą ochrony przed wszystkimi omawianymi technikami jest próba wprowadzenia włamywacza w błąd poprzez modyfikację/przekłamanie pewnych informacji. Pierwszym krokiem przy podjęciu tej metody jest ustalenie docelowego systemu, pod który można się podszyć. Ustalenie to powinno być dokonane na podstawie udostępnianych usług. Następnie należy zadbać o to, aby system przedstawiał się w sposób charakterystyczny dla systemu docelowego. Na poziomie aplikacji konieczna jest tu analiza działających usług pod kątem komunikatów, w których prezentowane są informacje, mogące zdradzić działający system. Jest to ta sama akcja, która wcześniej powinna być wykonana przy zabezpieczaniu systemu przed metodą pasywnego odcisku palca. Na poziomie protokołów sieciowych zabezpieczenia można dokonać w systemie Linux za pomocą projektu IPpersonality (*htpp://ippersonality.sourceforge.net*) będącego poprawką jądra, udostępniającego dodatkowy łańcuch dla iptables dostarczający możliwości zmiany typowych odpowiedzi systemu. Inne przykładowe rozwiązanie za pomocą obsługi pakietów w przestrzeni użytkownika (udostępnianej przez netfilter wbudowany w jądra serii 2.4) przedstawił Rob Beck w artykule [5]. Przedstawionej metody nie należy stosować jako jedynego, skutecznego zabezpieczenia przed atakami. Może ona jednak stanowić dodatkowy próg w zabezpieczeniach, opóźnić atak, a tym samym dać czas administratorowi na identyfikację zagrożenia. W szczególności metoda ta nie ochroni systemu przed automatycznymi skanerami zabezpieczeń testującymi konkretne luki uruchomionego na serwerze oprogramowania. Dodatkowo metoda źle zastosowana może otwierać dodatkowe luki w zabezpieczonym wcześniej systemie.

5. Podsumowanie

Zaprezentowane sposoby uzyskiwania informacji na temat działającego systemu operacyjnego stanowią duże zagrożenie dla jego bezpieczeństwa. Jest to powód, dla którego testy penetracyjne powinny stanowić integralną część polityki bezpieczeństwa. W ramach testu penetracyjnego szczególny nacisk powinien być położony na identyfikację systemu operacyjnego, ponieważ przełamanie jego zabezpieczeń stanowi bezpośrednie zagrożenie dla wszystkich usług działających pod jego kontrolą oraz danych przez niego zabezpieczanych.

LITERATURA

1. Remote OS detection via TCP/IP Stack FingerPrinting by Fyodor (http://www.insecure.org/nmap/nmap-fingerprinting-article.html) 1999.
2. ICMP Usage in Scanning - The Complete Know How, Ofir Arkin Sys-Security Group, 2000-2001.
3. X remote ICMP based OS fingerprinting techniques, Ofir Arkin, Sys-Security Group, 2001.
4. Passive System Fingerprinting using Network Client Applications, Jose Nazario, crimelabs research, 2001.
5. Passive-Aggressive Resistance: OS Fingerprint Evasion, Rob Beck, Linux Journal #89, 2001.

Recenzent: Dr inż. Krzysztof Nałęcki

Wpłynęło do Redakcji 17 kwietnia 2002 r.

Abstract

The article describes methods of identifying the operating system that controls host under attack. This action allows selecting exploits that might be used against the attacked system. Operating system identification can be made by reading system banners in poorly configured systems (so called out-of-the box configurations). More advanced techniques were also presented. Those include usage of TCP and ICMP protocols. Both protocols allow accurate identification by finding differences in TCP/IP stack implementations. This process is called fingerprinting. It can be done in active and passive ways. Active OS fingerprinting is done by generating special traffic targeted to the probed systems. The response for this traffic is analyzed in order to make precise guess of operating system vendor and version. Passive OS fingerprinting is done through sniffing traffic from the probed host. Also in this case accurate guess can be made on the basis of differences in implementation. This method takes more time but is in many cases undetectable. Methods presented in the article are implemented in many tools available on the Internet which poses an additional risk. As a countermeasure penetration tests should be performed. It is always better for the administrators to check what their systems can tell hackers before they perform the same check.