Software 2.0 nr 9 (93), wrzesień 2002, s. 48-54

Omijanie systemów wykrywania włamań

Piotr Dorosz
email: piotr.dorosz@pyton.pl
Autor dwóch lat zajmuje się problematyką systemów wykrywania włamań. Jest także twórcą autorskiej przynęty internetowej

Przemysław Kazienko
email: kazienko@pwr.wroc.pl
Autor od kilku lat prowadzi kurs z ochrony danych na Wydziale Informatyki i Zarządzania Politechniki Wrocławskiej.

1. Co to jest system wykrywania włamań

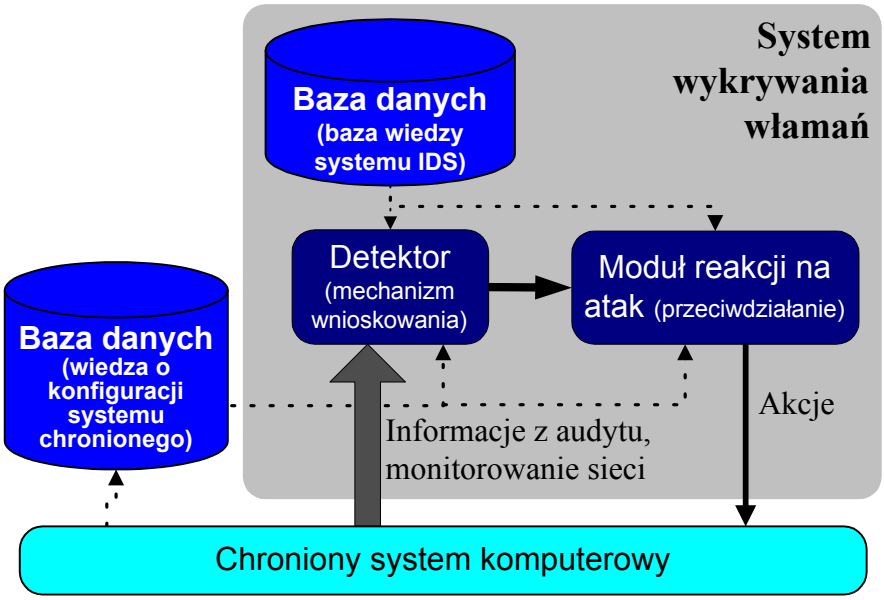
Systemy wykrywania włamań — IDS (*Intrusion Detection Systems*) służą do identyfikacji zbioru akcji, które naruszają integralność, poufność lub dostępność zasobów czyli innymi słowy mają wyłapywać podejrzane działania hakerów i ewentualnie odpowiednio na nie reagować. Ogólnie mówiąc, systemy IDS mogą działać na dwa sposoby:

- poprzez ***wykrywanie anomalii*** (*anomaly detection*, *behavior based systems*); system IDS bazuje na profilu normalnego zachowaniu systemu lub użytkowników, zaś wszystko co odbiega od tego profilu jest traktowane jako działanie podejrzane — potencjalne włamanie;
- poprzez ***wykrywanie sygnatur*** czyli ***wzorców*** (*signature detection*, *knowledge based systems*); system w celu wykrycia ataku bazuje na wiedzy o nienormalnych zachowaniach lub sygnaturach ataków.

Pierwszy sposób jest bardzo przydatny przy wykrywaniu ataków wewnętrznych, dokonywanych przez własnych pracowników. Wymaga on jednak wiedzy o normalnych zachowaniach użytkowników. Pozyskanie tej wiedzy i jej aktualizacja nie jest jednak prosta. Wykrywanie anomalii charakteryzuje się dużą liczba fałszywych alarmów.

W drugiej metodzie niezbędna jest baza o wzorcach ataków. Ponieważ jednak trudno budować profile niepoprawnych zachowań uprawnionych użytkowników, metoda ta jest szeroko wykorzystywana do detekcji ataków zewnętrznych. Wykrywanie sygnatur jest bardzo popularne w komercyjnych systemach IDS, gdyż wzorce muszą być ciągle uaktualniane (co może być źródłem niezłych dochodów) oraz łatwiej jest opisać typowy atak (taki sam na całym świecie) niż typowe zachowanie konkretnej osoby. Metoda ta często sprowadza się do porównywania ciągów znaków (z wzorca ataku z danymi dostarczanymi do systemu IDS) lub wykrywania określonych parametrów pakietów sieciowych. Wykrywanie sygnatur ma jednak poważną wadę, którą można wykorzystać przy omijaniu systemu IDS. W metodzie tej nie jest możliwe wykrycie włamania, które nie zostało umieszczone w bazie wzorców ataków. Wystarczy więc tak zmodyfikować działania wykonywane podczas ataku, aby nie pasowały one do wzorca.

Typowy system IDS ma postać jak na rys.1. Baza wiedzy systemu IDS to przede wszystkim baza wzorców ataków. Sygnatury w niej zawarte są zwykle różne dla różnych systemów operacyjnych, gdyż wiele ataków wykorzystuje luki i słabości konkretnych systemów.



Rysunek 1. Model systemu IDS. Grubość strzałek reprezentuje ilość przepływającej informacji [1].

Informacje przekazywane przez system chroniony do systemu IDS pochodzą z dwóch źródeł: ***monitorowania ruchu sieciowego*** lub z ***danych audytu*** (dzienników zdarzeń). Oba źródła mają swoje cechy charakterystyczne, które mogą być wykorzystane podczas omijania systemu IDS. Systemy analizujące ruch sieciowy posiadają zwykle dostęp jedynie do ostatnich pakietów i najczęściej nie są w stanie powiązać działań, które dokonały się kilka pakietów lub kilka minut wcześniej. Jeżeli więc atak zostanie „rozciągnięty" w czasie, to bardzo trudno będzie go wykryć. Natomiast systemy analizujące dane audytu nie są w stanie działać w czasie rzeczywistym — zwykle są uruchamiane okresowo. Haker może więc zdążyć zatrzeć ślady swojej działalności, np. modyfikując pliki audytu.
Właściwie każdy z elementów systemu wykrywania włamań może być podatny na ataki. Detektor, moduł odpowiedzialny za wykrycie włamania — serce systemu IDS — może zostać oszukany lub unieruchomiony (zaślepiony), np. poprzez zalanie (*flood*). Moduł reakcji na atak również może zostać unieruchomiony lub zablokowane może być jego połączenie z resztą systemu przez co np. wysłany alarm nie dotrze do adresata lub dotrze ze zbyt dużym opóźnieniem.

2. *Główne rodzaje ataków przeciwko systemom IDS*

Intruz, aby pozostać niewykrytym, może zaatakować sam system IDS za pomocą jednego z następujących, głównych rodzajów ataków [1]:

- ***Uniknięcie wykrycia przez zmianę wzoru***. Większość systemów IDS wykorzystuje wzorce ataków. Moduł detekcji przy porównywaniu sygnatur może być jednak łatwo ominięty poprzez niewielką zmianę działań intruza w trakcie ataku (*pattern change evasion*). Intruz, znając rodzaj stosowanego systemu IDS, może także znać sygnaturę ataku. Dzięki temu potrafi on tak zmodyfikować swoje działania, aby atak był niezgodny ze wzorcem, np. poprzez podjęcie nieistotnych akcji lub zmianę parametrów.
- ***Fragmentacja*** — omówiona dalej szczegółowo — jest działaniem powodującym podział pojedynczego pakietu na wiele mniejszych. Stos TCP/IP odbiorcy, łączy części pakietów, przed przekazaniem do aplikacji. Niektóre z systemów IDS nie posiadają możliwości defragmentacji pakietów, co pozwala na wykorzystanie narzędzi typu *fragrouter* do ich oszukania.
- ***Powolne skanowanie***. Ze względu na objętość przetwarzanego ruchu w sieci systemy IDS mają problemy z dłuższym przechowywaniem danych oraz korelacją odległych zdarzeń. Z

tego względu powolne skanowanie (skanowanie portów lub skanowanie *ping sweeps*), w którym napastnik skanuje jeden port na godzinę, może być bardzo trudne do wykrycia.

- ***Skoordynowany powolny atak***. Kilku intruzów może przeprowadzić skoordynowany powolny atak (skanowanie) pochodzący z wielu adresów IP. System IDS może mieć duże trudności ze skorelowaniem tych informacji.
- ***Zmiana adresów***. Jednym z zadań wykrywania włamań jest wskazanie atakującego. Może to być z wielu powodów zadanie bardzo trudne. W przypadku ataku *Smurf* (pakiety rozgłoszeniowe z adresem nadawcy zmienionym na adres ofiary) system IDS przetwarza tysiące odpowiedzi na pakiety, których ofiara nigdy nie wysłała. IDS nie jest jednak w stanie stwierdzić, skąd tak naprawdę pochodzą podrobione pakiety ICMP. Z tego powodu identyfikacja intruza może być utrudniona, co utrudnia zastosowanie mechanizmów reakcji na atak.
- ***Unikanie wartości domyślnych***. Często zapory ogniowe wykorzystywane są jako proste systemy IDS W zaporach tych port przeznaczenia zwykle określa usługę. Intruz po wdarciu się i zainstalowaniu tylnego wejścia do systemu może uruchomić usługi na nie standardowych portach. Przykładowo może on zainstalować program *BackOriffice* lecz zmienić standardowy jego port (31337). Wiele zapór i systemów IDS nie będzie w stanie poprawnie zinterpretować (i co za tym idzie zatrzymać) ruchu na ten port. Systemowi IDS może być trudno wykryć wszelkie inne ataki na standardowe usługi zainstalowane na nietypowych portach.

3. Słabości porównywania ciągów znakowych

Najczęściej wykorzystywaną metodą wykrywania włamań jest porównanie aktualnych danych pochodzących z systemu (treści lub parametrów pakietów, zapisów audytu) z wzorcem ataku. Dokonuje się tego przez proste porównanie ciągów tekstowych. Jeżeli pakiet zawiera niedozwolony, podany w sygnaturze ataku ciąg, to IDS uznaje, że wykrył określony rodzaj ataku.

Weźmy pod uwagę przykładową sygnaturę ataku dla systemu IDS o nazwie Snort (jest to system typu *open source* dostępny pod adresem http://www.snort.org) o postaci:

```
alert tcp $ZEWNETRZNA_SIEC any -> $SERWERY_HTTP 80 (msg:"Atak
przez serwer HTTP - odwolanie do /etc/passwd"; flags: A+;
content:"/etc/passwd"; nocase; classtype:atak-na-plik-hasel;)
```

Za jej pomocą program Snort powinien wykryć każde połączenie pochodzące z zewnętrznej sieci (zakres adresów zawarty w zmiennej `ZEWNETRZNA_SIEC`), na port 80 serwerów WWW w naszej sieci lokalnej (ich adresy są podane w zmiennej `SERWERY_HTTP`), w którym pakiet TCP, w pełni nawiązanym połączeniu (`A+`), zawiera ciąg znaków `/etc/passwd`, bez rozróżnienia wielkości liter. Atak jest więc wykrywany poprzez wyszukiwanie zadanego ciągu znaków w treści pakietu TCP. Jeżeli więc zmienimy ów ciąg w trakcie ataku, to włamanie pozostanie niewykryte.

Dokonać tego można zmieniając ciąg z żądania HTTP wykrywalny dla systemu IDS:

```
GET /etc/passwd
```

na ciąg niezgodny z wzorcem:

```
GET /etc/pa%73%73wd
```

w którym literki „s” zostały zakodowane szesnastkowo, lub kodując pozostałe litery:

```
GET /%65%74%63/%70%61%73%73%77%64
```

Jeszcze innym sposobem jest zmiana postaci ścieżki dostępu do pliku `passwd`, np. poprzez przejście do innej kartoteki (jest to zależne od dystrybucji systemu operacyjnego) , czyli np.:

```
GET /etc/rc1.d/../passwd
```

Przykładem tego rodzaju prób „zmylenia” systemu IDS jest również żądanie typu:

```
http://ofiara.pl/www%2findex.htm
```

w którym znak „/” został zakodowany szesnastkowo (%2f). Nic nie stoi na przeszkodzie, aby pójść krok dalej i „powędrować” po drzewie katalogowym:

```
http://ofiara.pl/www%2f..%2fwww%2findex.htm
```

Gdy dla połączeń telnetowych weźmiemy podobną do poprzedniej regułę (tylko dla pliku `/etc/shadow`):

```
alert tcp any any -> $MOJ_SERWER_TELNET 23 (msg:"Telnet –
odwolanie do /etc/shadow"; flags:  A+; content:"/etc/shadow";
classtype:atak-na-plik-hasel;)
```

to w sesji telnetowej chcąc — w niezauważalny dla systemu IDS sposób — wykonać polecenie:

```
/bin/cat /etc/shadow
```

możemy również wykorzystać odpowiednie dla atakowanego systemu operacyjnego języki programowania, poprzez które nakażemy wygenerowanie żądanego ciągu `/etc/shadow`. Ciąg ten zostanie przekazany do właściwej komendy (tutaj `cat`), a komenda będzie wywołana. Dla języka perl w linii komend wystarczy wpisać:

```
perl –e '$atak=pack("C11",47,101,116,99,47,115,104,97,100,111,
119); @komenda="/bin/cat $atak"; exec"@komenda\n";'
```

Jedenastoznakowy ciąg `/etc/shadow` jest tutaj zakodowany poprzez dziesiętne kody znaków (stąd „C11”).

W przypadku takiego żądania system IDS musiałby albo potrafić przeanalizować ciąg instrukcji (w tym wypadku działać jak interpreter języka Perl) lub zawsze alarmować o wykorzystaniu komendy perl jako podejrzanej. Pierwsza z możliwości jest w zasadzie niewykonalna, druga dyskusyjna, aczkolwiek specyficzna w danej organizacji polityka bezpieczeństwa może wykluczać stosowanie języków programowania przy telnetowym dostępie dla danego użytkownika.

Kolejnym sposobem ominięcia systemu IDS jest zastosowanie kodowania Unicode, które umożliwia inny sposób zapisu znaków [2]. Przykładowo, intruz żąda dokumentu o adresie URL o postaci:

```
http://ofiara.pl/../../winnt/system32/cmd.exe
```

System IIS wygeneruje alarm, gdyż wykryje próbę wyjścia poza główny katalog serwera WWW. Jednak po zakodowaniu podejrzanego ciągu „../..” za pomocą UTF-8 („..%C1%9C..”) wyjście poza główny katalog serwera IIS może okazać się możliwe[1]. Co więcej, znak „/” można zaszyfrować także za pomocą różnych, innych kodów Unicode, co wykorzystuje worm Nimda, np. %c0%af, %c1%pc, %c0%9v, %c0%qf, %c1%8s, %c1%1c, %e0%80%af, %f0%80%80%af, %f8%80%80%80%af, %fc%80%80%80%80%af, itd. (http://www.owasp.org/asac/canonicalization/unicode.shtml).

Konkluzja

Próby ominięcia systemów IDS poprzez kodowanie ścieżek lub znaków nie są szczególnie złożonymi ani odkrywczymi atakami. Większość z nich jest znana już od dość dawna, co powoduje że nowsze systemy IDS potrafią wykryć tego rodzaju „uniki” poprzez dokonywanie porównania ciągu znaków po zastosowaniu wszystkich konwersji. W przypadku systemu Snort stosuje się na przykład następujący zapis reguły:

```
alert tcp any any -> $SERWERY_HTTP 80 (msg: "Serwer WWW –
konwersja backslash w Unicode"; flags: AP; content:
"..|25|c1|25|9c"; nocase;)
```

[1] Błąd ten był wykorzystywany przy ataku na serwery IIS.

gdzie szukany wzorzec zapisany jest w postaci szesnastkowej (czyli tak jak intruz próbujący zmylić system IDS). Mimo to starsze systemy IDS mogą nie wykryć tego rodzaju ataku, zaś mnogość możliwości kodowania znaków dodatkowo zwiększa liczbę wzorców ataków.

4. Polimorficzne zmiany postaci ataku

Następna z możliwości oszukania ominięcia IDS przez intruza polega na zastosowaniu zmiennych wzorców ataku. Polimorfizm jest często wykorzystywany w atakach powodujących przepełnienie bufora. Za przykład niech posłużą następujące sygnatury systemu Snort dotyczące ataku *SSH CRC32 buffer overflow*:

```
alert tcp $ZEWNETRZNA_SIEC any -> $WEWNETRZNA_SIEC 22
(msg:"Wykorzystanie EXPLOIT ssh CRC32 overflow /bin/sh";
flags:A+; content:"/bin/sh"; classtype:ataki-na-ssh;)
```

```
alert tcp $ZEWNETRZNA_SIEC any -> $WEWNETRZNA_SIEC 22
(msg:"Wykorzystanie EXPLOIT ssh CRC32 overflow NOOP";
flags:A+; content:"|90 90 90 90 90 90 90 90 90 90 90 90 90 90
90";)
```

Pierwsza z nich szuka ciągu „/bin/sh” w pakietach kierowanych na port 22 (standardowy dla ssh) naszej sieci domowej. Druga z sygnatur szuka w treści pakietu pewnej liczby znaków, które w skompilowanym kodzie (dla procesorów z rodziny x86) oznaczają rozkaz NOP[2]. Biorąc pod uwagę rodzinę procesorów o architekturze x86, kod operacji NOP może być zapisany na 55 różnych sposobów (patrz http://cansecwest.com/noplist-v1-1.txt). Pozwala to intruzowi korzystać (nawet losowo) z różnych modyfikacji ataku na usługę SSH i ominięcie systemu IDS posiadającego tylko powyższy zestaw reguł.

Wykrycie tego rodzaju prób ominięcia systemu IDS jest bardzo trudnym zadaniem. Mimo to, powstał ostatnio moduł do programu Snort pozwalający na wykorzystanie opisanych powyżej mechanizmów do wykrywania ataków (spp_fnord.c autorstwa Dragos’a Ruiu — http://www.geocrawler.com/archives/3/5344/2002/3/150/8010404/).

5. Dzielenie sesji na pakiety

Kolejnym sposobem ominięcia systemu IDS jest ukrycie ataku w ramach sesji z użytkownikiem poprzez podzielenie żądania na wiele pakietów (*sesssion splicing*). Można do tego zastosować narzędzie Whisker (http://sourceforge.net/projects/whisker/), udostępniającego podział sesji na pakiety. W tym przypadku ciąg przesyłany do systemu IDS jest dzielony następująco:

Numer pakietu	1	2	3	4	5	6	7	8
Zawartość pakietu	G	E	T	20 (spacja)	/	e	t	c

Poprzez podzielenie ataku na wiele pakietów, możliwe jest uniknięcie porównania żądania z wzorcem ataku. Aby obsłużyć tego rodzaju żądanie i nie zostać oszukanym, IDS musi potrafić określić przebieg sesji użytkownika z systemem, wymusić połączenie pakietów przed porównaniem lub wprost wykryć tego rodzaju próbę ominięcia za pomocą specjalnej sygnatury. Przykładową regułą dla systemu Snort, wykrywającą ten rodzaj ataku mogłaby być:

[2] No operation – brak operacji. Rozkazy tego typu często wykorzystywane są w exploitach.

```
alert tcp $ZEWNETRZNA_SIEC any -> $SERWERY_HTTP 80
(msg:"Serwer WWW – podział sesji"; content:"|20|"; flags:A+;
dsize:1; classtype:podzial-sesji;)
```

Ta reguła wykrywa ruch TCP kierowany na port 80 serwerów HTTP, z ustawioną flagą ACK, spacją (0x20 hex) w treści pakietu oraz liczbą bajtów danych przesyłanych w pakiecie równą jeden. Sygnatura ta wykryje oczywiście tylko zastosowanie narzędzia Whisker, które dzieli sesję na 1-bajtowe pakiety. Jednakże łatwo można stworzyć podobne próby, które nie zostaną wykryte przez tę sygnaturę, np. dzieląc żądanie na pakiety o długości dwóch bajtów.

Aby bronić się przed tego rodzaju działaniami należy zmodyfikować powyższą regułę, tak aby alarmowała po przechwyceniu danych kierowanych do serwera HTTP o bardzo małej długości. Jednakże z pewnością wygeneruje to wiele fałszywych alarmów (*false positives*) a przy tym ominięcie systemu IDS ciągle będzie możliwe (choćby poprzez eksperymentalne dobranie bardzo małej wartości pola TTL[3] (*time to live*) w pakietach kierowanych do systemu, tak aby były one odbierane przez system IDS a nie dochodziły do atakowanego hosta). Aby w pełni bronić się przed tego rodzaju atakami system IDS powinien potrafić odtworzyć sesję użytkownika z systemem, co nie jest zadaniem prostym a dodatkowo jest zasobochłonne. W chwili obecnej większość systemów IDS umożliwia odtworzenie sesji użytkownika, lecz tylko przez ograniczony czas. W rzeczywistych systemach sesja użytkownika jest oceniana przez serwer jako otwarta i legalnie trwająca, jeżeli do serwera będzie wysyłany jeden bajt raz na np. 15 minut (sesja z IIS). Najnowsze wersje systemu Snort potrafią jednak nie zamykać sesji przez długi czas oraz wykrywać stosowanie takich sztuczek jak niska wartość pól TTL [4].

6. Fragmentacja

Fragmentacja działa podobnie jak zastosowanie dzielenia sesji tylko na niższym poziomie. Do niedawna jeszcze większość systemów IDS nie potrafiła odpowiednio, zdefragmentować (poskładać) pakietów przed porównaniem ich z wzorcem ataku, co powodowało, że wiele ataków pozostawało nie wykrytych. Obecnie coraz więcej systemów IDS dokonuje defragmentacji pakietów przed porównaniem ich z sygnaturą ataku, jednak istnieje kilka sposobów na ominięcie IDS za pomocą fragmentacji. Problemem dla systemu IDS jest to, że zanim pakiet zostanie całkowicie zdefragmentowany, musi on być przechowywany w pamięci, co jest możliwe tylko przez pewien czas. Jeszcze bardziej złożoną sprawą jest zagwarantowanie takiej samej defragmantacji pakietów, jaką dokona system, do którego dany pakiet jest adresowany. Wymaga to znajomości chronionego systemu. Zasadniczy problem tkwi w odmiennym traktowaniu niewłaściwej kolejności pojawiania się fragmentów oraz w obsłudze fragmentów powtarzających się. Różne systemy reagują w różny sposób na powtarzające się fragmenty: preferują pierwszy, ostatni lub dopełniają różnice[4]. Dodatkowo intruz może odpowiednio manipulować polem przesunięcia (*fragment offset*) danego fragmentu IP i powodować częściowe lub całkowite nadpisywanie danych. Generalnie atak związany z fragmentacją wykorzystuje to, że system IDS inaczej niż system chroniony łączy poszczególne (zwłaszcza duplikujące się) fragmenty w całość.
Istnieją różne techniki ataków wykorzystujące fragmentację pakietów [4]:

- fragmentacja zachodząca (*fragmentation overlap*)
- fragmentacja nadpisująca (*fragmentation overwrite*)
- fragmentacja przeterminowana (*fragmentation time-outs*)

[3] Pole TTL określa ile routerów pakiet może przejść zanim zostanie zniszczony.
[4] Przykłady różnego podejścia do fragmentów zaprezentowano w [3], np. Windows NT 4.0 preferuje stare fragmenty podczas gdy HP-UX 9.01 preferuje fragmenty nowe.

Fragmentacja zachodząca

Polega ona na tym, że po zdefragmentowaniu część kolejnego pakietu pokrywa dane wysłane w pakiecie poprzednim. Weźmy dwa pakiety IP z dwoma fragmentami [4].

Pakiet pierwszy: `GET x.idd` *ustawiony bit MF (more fragments)*
Pakiet drugi: `a?`*kod_przepełnienia_bufora*

Jeżeli pakiety będą zdefragmentowane w ten sposób, że pakiet numer 2 nadpisuje ostatni bajt pakietu pierwszego (uzyskać można to przez odpowiedni dobór pola *Fragment Offset* pakietu IP), wtedy komputer, do którego kierowany jest pakiet otrzymuje żądanie:

`GET x.ida?`*kod_przepełnienia_bufora*

Jest to atak (stosowany m.in. w jednej z odmian worma *Code Red*) wykorzystujący lukę przepełnienia bufora w usłudze indeksowania serwera IIS (http://www.eeye.com/html/Research/Advisories/AD20010618.html). System wykrywania włamań, jeżeli inaczej połączy fragmenty niż system docelowy lub w ogóle ich nie połączy (jest to warunek powodzenia ataku), to nie wykryje włamania, gdyż wynikowy ciąg będzie inny niż dostępny dla systemu IDS wzorzec ataku.

Fragmentacja nadpisująca

Jest to technika podobna do poprzedniej, działająca z tą różnicą, że jeden z fragmentów pakietu całkowicie nadpisuje inny z nich. Wystarczy stosownie dobrać kolejność pakietów (odpowiednio do traktowania tych samych fragmentów przez system ofiary).

Pakiet pierwszy: `GET x.id`
Pakiet drugi: *dowolny ciąg*
Pakiet trzeci: `a?`*kod_przepełnienia_bufora*

W zależności od tego jak komputer, do którego kierowane są pakiety, dokona ich łączenia (uważając za ważniejsze nowsze lub starsze fragmenty pakietów), poniższe dane będą mogły być odebrane przez niego jako: atak wykorzystujący przepełnienie bufora (pakiet trzeci nadpisujący pakiet drugi) lub niegroźne żądanie HTTP (pakiet trzeci zignorowany). Oczywiście także tutaj warunkiem powodzenia jest to, że łączenie fragmentów inaczej przebiega w systemie IDS i w systemie ofiary.

Fragmentacja przeterminowana

Technika ta zależy od tego, jak długo IDS będzie trzymał w pamięci fragmenty pakietów zanim zostaną one z niej usunięte. Większość systemów uznaje, że należy usunąć fragmenty pakietów, które nie tworzą całości, po 60 sekundach. Jeżeli system IDS nie czeka na pełną defragmentację pakietów przez całe 60 sekund (np. ze względu na zbyt duże wymagania dotyczące pamięci), możliwe jest ominięcie go poprzez wysłanie następującego żądania:

Pakiet pierwszy: `GET x.id`
Pakiet drugi (59 sekund później): `a?`*kod_przepełnienia_bufora*

System ofiary połączy oba pakiety, natomiast IDS potraktuje je oddzielnie — nie wykryje ataku.

Fragmentacja połączona z TTL

Fragmentacja może być wykorzystana do ominięcia systemu IDS wraz z innymi technikami, takimi jak niskie wartości pól TTL pakietów. Aby wykorzystać tę technikę intruz musi tak dobrać wartości pól TTL niektórych pakietów, aby atakowany host nie otrzymał części z nich, zaś system IDS otrzymał wszystkie. Przykładowo:

Numer pakietu	Zawartość	TTL
1	GET x.id	> 2
2	szum	1

3	a?*kod_przepełnienia_bufora*	>2

System IDS będzie potrafił zdefragmentować żądanie, które będzie miało następującą postać:

```
GET x.idszuma?kod_przepełnienia_bufora
```

Przy takich wartościach pól TTL, atakowany host otrzyma jednak groźne żądanie:

```
GET x.ida?kod_przepełnienia_bufora
```

Oczywiście powodzenie zależy od architektury systemu chronionego, tzn. pakiety muszą najpierw przechodzić przez system IDS a następnie docierać do hosta ofiary. Dodatkowo, aby wykorzystać tą technikę omijania systemu IDS intruz musi empirycznie dobrać wartość pól TTL pakietów.

Konkluzja

Aby uchronić się przez wymienionymi wyżej rodzajami ataków, w systemie Snort można zastosować następującą regułę:

```
alert ip $ZEWNETRZNA_SIEC any -> $WEWNETRZNA_SIEC any (msg:"Za
male fragmenty"; fragbits:M; dsize:<25;
classtype:fragmentacja)
```

W niektórych rodzajach ataków, w których można dowolnie zwiększać długość żądania atakujący może jednak zadbać o odpowiednio duży rozmiar poszczególnych fragmentów, więc taki atak nie zostanie wykryty.

W systemach IDS ruch sieciowy kierowany do systemu IDS powinien być normalizowany (zgodnie z zaleceniami http://www.icir.org/vern/papers/norm-usenix-sec-01-html/) i sam system powinien potrafić dokonać takiej defragmentacji pakietów, która umożliwi mu uzyskanie dokładnie tych samych danych, które otrzyma host docelowy.

System *Fragroute* (http://www.monkey.org/~dugsong/fragroute/) pozwala na sprawdzenie wielu problemów związanych z prawidłową fragmentacją pakietów i ich poprawną deasemblacją przez IDS.

7. Ataki DoS

Inną metodą na ominięcie systemu IDS jest przeprowadzenie ataku typu odmowa usługi (*Denial of Service*), którego celem będzie sam system wykrywania intruzów, urządzenia z nim powiązane lub personel odpowiedzialny za ochronę systemów organizacji. Narzędzia typu *Stick* (http://www.eurocompton.net/stick/) czy *Snot* (http://www.sec33.com/sniph/) mogą zostać użyte przez intruza do wygenerowania dużej liczby alarmów, które [4]:

- zużywając moc obliczeniową pozwolą intruzowi na ominięcie systemu IDS (zajętego generowaniem alarmów);
- zapełnią przestrzeń dyskową co spowoduje, że „rzeczywisty” atak nie będzie zapisany w pliku alarmów systemu IDS;
- spowodują wygenerowanie dużej liczby innych alarmów, które będą musiały zostać obsłużone przez inne systemy (np. wysłanie SMS-a na komórkę administratora)
- uniemożliwią administratorom zajęcie się wszystkimi alarmami;
- doprowadzą do zawieszenia urządzeń odpowiedzialnych za ochronę systemu.

Ogólnie system IDS (aby potrafić przeanalizować kierowany do niego ruch), musi mieć w przybliżeniu, tak złożoną budowę jak złożony jest cały stos TCP/IP chronionych systemów. Jest więc on równocześnie podatny na ataki takie jak *SYN Floods* czy *Smurf*.

Zaślepienie detektora IDS

Systemy wykrywania intruzów bazujące na informacjach pochodzących z sieci lokalnej (*network based*) zwykle posiadają architekturę, która powoduje, że cały ruch sieciowy przechodzi przez system IDS. Ruch ten jest analizowany przez detektor IDS (rys. 1). W związku z tym, możliwy jest atak polegający na zalaniu (zaślepieniu) detektora (*blind the sensor*) bardzo dużą liczbą pakietów. System IDS będzie musiał pobrać wszystkie pakiety. Jednak nie będzie w stanie ich zanalizować, czego skutkiem może być utrata części pakietów i niezdolność wykrycia włamania.

Zaślepienie systemu przechowującego zdarzenia

Atak zaślepienia systemu przechowującego (*blind the event storage* lub *snow blind*) polega na skanowaniu zdalnego komputera przy pomocy setek zmienionych pakietów przemieszanych z pakietami, które mają wpisany prawdziwy adres IP intruza i służą atakowi. Powoduje to ogromne trudności w ustaleniu, który adres jest adresem intruza. Dodatkowo, baza danych przechowująca zdarzenia dziejące się w systemie, może w takich wypadkach zostać przepełniona, co spowoduje wykasowanie części wcześniejszych zdarzeń lub nie zapisywanie nowych. W obu przypadkach tożsamość atakującego pozostanie tajemnicą. Tego typu atak jest realizowany np. przez funkcję *decoy scan* w systemie *Nmap*.

8. Inne sposoby atakowania systemu IDS

W złożonym przypadku ominięcia systemu IDS intruz wysyła pakiety TCP FIN, które zobaczy IDS, ale które nie dotrą do chronionego systemu. Powoduje to, iż IDS (w przeciwieństwie do ofiary) sądzi, że połączenie jest zamknięte. Ze względu na fakt, iż połączenie TCP nie sprawdza aktualnej aktywności połączenia, host ofiary może przez długi czas utrzymywać otwarte połączenie, co może być wykorzystane do ataku. W praktyce większość interesujących intruza serwisów zamyka połączenie po jakimś czasie braku aktywności, zawsze jednak pozostawia to włamywaczowi przynajmniej kilka minut. Podstawą ataku tego typu jest znalezienie sposobu na przesłanie pakietu do IDS, z jednoczesnym zablokowaniem jego dalszej drogi. Może to być wykonane poprzez wysłanie pakietów, które zostaną przez router usunięte. W tym celu wystarczy zastosować wspomnianą wyżej fragmentację lub nadać pakietom odpowiednio niską wartość pola TTL. Jeżeli za systemem IDS jest łącze o małej przepustowości, to intruz może zablokować je pakietami IP o wysokim priorytecie a wtedy pakiet TCP FIN kończący połączenie zostanie przez router zgubiony (jako pakiet o niskim priorytecie).

W innym przykładzie złożonego ataku wykorzystywana jest wiedza o budowie stosu TCP/IP systemu IDS oraz systemów chronionych. Różne stosy TCP zachowują się nieco inaczej przy różnych, nieodpowiednich wartościach niektórych pól w nagłówkach pakietów, co programy (takie jak *Nmap* czy *queso*) wykorzystują do wykrycia rodzaju systemu operacyjnego. Wiedza ta może być wykorzystana do przeprowadzenia ataku na moduł sieciowy IDS.

Opisana wyżej fragmentacja jest jednym ze szczególnych wystąpień z grupy ataków, w których wykorzystuje się inne traktowanie pakietów przez system IDS i system ofiary. Niektóre pakiety (o określonych wartościach niektórych pól, w szczególności błędnych wartościach) mogą być akceptowane przez ofiarę i odrzucane przez system IDS (lub odwrotnie). Inna może być także reakcja na konflikty i nietypowe sytuacje (np. usuwanie nieaktywnego połączenia). Wszystko to można wykorzystać do modyfikacji ataku, tak by był niezgodny ze wzorcem. Dokonuje się tego poprzez: wstawanie (*insertion*) niepotrzebnych danych (IDS potraktuje je poważnie, zaś ofiara je opuści) lub omijanie (*evasion*), w którym IDS (w przeciwieństwie do systemu chronionego) odrzuci niektóre dane. W pracy [3] podano

niektóre przykłady błędów w pakietach, które mogą być różnie traktowane przez systemy, np. całkowita długość pakietu mniejsza niż długość nagłówka IP, podwójny pakiet TCP ACK, itd.

Bibliografia

[1] Dorosz P., Kazienko P. *Systemy wykrywania intruzów*. VI Krajowa Konferencja Zastosowań Kryptografii ENIGMA 2002, Warszawa 14-17 maja 2002 r. , s. TIV 47-78, http://www.enigma.com.pl/konferencje/vi_kkzk/index.htm

[2] Hacker E.: *IDS Evasion with Unicode*. last updated Jan. 3, 2001, http://online.securityfocus.com/infocus/1232

[3] Ptacek T.H., Newsham T.N.: *Insertion, Evasion and Denial of Service: Eluding Network Intrusion Detection*. Secure Networks, Inc., Jan. 1988, http://www.insecure.org/stf/secnet_ids/secnet_ids.html

[4] Timm K.: *IDS Evasion Techniques and Tactics*. 2002, Security Focus Infocus, http://online.securityfocus.com/infocus/1577