Wykorzystanie algorytmu genetycznego do wykrywania włamań do sieci

Wei Li
Department of Computer Science and Engineering
Mississippi State University, Mississippi State, MS 39762
Email: wli@cse.msstate.edu

Streszczenie

W artykule opisano technikę zastosowania algorytmu genetycznego (GA) do sieciowych systemów wykrywania włamań (IDS). Przedstawiono krótki przegląd systemu wykrywania włamań, algorytmu genetycznego i powiązanych technik wykrywania. Szczegółowo omówiono parametry i proces ewolucji GA. W przeciwieństwie do innych implementacji tego samego problemu, ta implementacja uwzględnia zarówno czasowe, jak i przestrzenne informacje o połączeniach sieciowych w kodowaniu informacji o połączeniach sieciowych w reguły w IDS. Jest to pomocne w identyfikacji złożonych anomalii. Niniejsza praca koncentruje się na protokołach sieciowych TCP/IP.

1. Wprowadzenie

W ostatnich latach system wykrywania włamań (IDS) stał się jednym z najgorętszych obszarów badawczych w dziedzinie bezpieczeństwa komputerowego. Jest to ważna technologia wykrywania i jest stosowana jako środek zaradczy w celu zachowania integralności danych i dostępności systemu podczas włamania.

Kiedy intruz próbuje włamać się do systemu informatycznego lub wykonuje działanie niedozwolone przez prawo, nazywamy to *włamaniem* (Graham, 2002; patrz także Jones i Sielken, 2000). Intruzów można podzielić na dwie grupy: *zewnętrznych* i *wewnętrznych*. Pierwsza odnosi się do tych, którzy nie mają autoryzowanego dostępu do systemu i którzy atakują przy użyciu różnych technik penetracji. Druga odnosi się do osób z uprawnieniami dostępu, które chcą wykonywać nieautoryzowane działania. Techniki włamań mogą obejmować wykorzystywanie błędów w oprogramowaniu i błędnej konfiguracji systemu, łamanie haseł, podsłuchiwanie niezabezpieczonego ruchu lub wykorzystywanie wad projektowych określonych protokołów (Graham, 2002). System wykrywania włamań to system służący do wykrywania włamań i dokładnego zgłaszania ich odpowiednim organom. Systemy wykrywania włamań są zwykle specyficzne dla systemu operacyjnego, w którym działają i są ważnym narzędziem w ogólnym wdrażaniu polityki bezpieczeństwa informacji organizacji (Jones i Sielken, 2000), która odzwierciedla oświadczenie organizacji poprzez określenie zasad i praktyk w celu zapewnienia bezpieczeństwa, obsługi włamań i odzyskiwania danych po szkodach spowodowanych naruszeniami bezpieczeństwa.

Istnieją dwie ogólnie przyjęte kategorie technik wykrywania włamań: *wykrywanie nadużyć* i *wykrywanie anomalii*. *Wykrywanie nadużyć* odnosi się do technik, które charakteryzują znane metody penetracji systemu. Te penetracje są scharakteryzowane jako "wzorzec" lub "sygnatura", której szuka IDS. Wzorzec/sygnatura może być statycznym ciągiem lub ustaloną sekwencją działań. Odpowiedzi systemu są oparte na zidentyfikowanych penetracjach. *Wykrywanie anomalii* odnosi się do technik, które definiują i charakteryzują normalne lub akceptowalne zachowania systemu (np. użycie procesora, czas wykonywania zadań, wywołania systemowe). Zachowania, które odbiegają od oczekiwanego normalnego zachowania, są uważane za włamania (Bezroukov, 2002; patrz także McHugh, 2001).

Systemy IDS można również podzielić na dwie grupy, w zależności od tego, gdzie szukają inwazyjnych zachowań: IDS *oparte na sieci (NIDS)* i *IDS oparte na hostach*. Pierwsza z nich odnosi się do systemów, które identyfikują włamania poprzez monitorowanie ruchu za pośrednictwem urządzeń sieciowych (np. karty sieciowej, NIC). *IDS oparty na hoście* monitoruje aktywność plików i procesów związanych z e środowiskiem oprogramowania powiązanym z określonym hostem. Niektóre *IDS oparte na hoście* nasłuchują również ruchu sieciowego w celu identyfikacji ataków na hosta (Bezroukov, 2002; patrz także McHugh, 2001). Istnieją również inne nowe techniki. Jednym z przykładów jest znany jako *blokujący IDS*, który łączy IDS oparty na hoście z możliwością modyfikowania reguł zapory sieciowej (Miller i Shaw, 1996). Inną jest *Honeypot,* który wydaje się być "celem" dla intruza, ale jest specjalnie zaprojektowany, aby uwięzić intruza w celu wyśledzenia jego lokalizacji i zareagowania na atak (Bezroukov, 2002).

Intelligent Intrusion Detection System (IIDS) jest projektem realizowanym w Center for Computer Security Research (CCSR) na Mississippi State University. Architektura ta łączy w sobie wiele różnych podejść do problemu IDS i obejmuje różne techniki sztucznej inteligencji, które pomagają identyfikować natrętne zachowania (Bridges i Vaughn, 2001). Wykorzystuje zarówno techniki wykrywania anomalii, jak i wykrywania nadużyć i jest systemem opartym zarówno na sieci, jak i na hoście. W ramach ogólnej architektury IIDS, niektóre narzędzia oprogramowania do wykrywania włamań typu open-source są zintegrowane do użytku jako czujniki bezpieczeństwa (Li, 2002), takie jak Bro (Paxson, 1998) i Snort (Roesch, 1999). Techniki zaproponowane w tym artykule są częścią wysiłków badawczych IIDS.

Algorytm genetyczny (GA) był wykorzystywany na różne sposoby w systemach IDS. Laboratoria Badań Stosowanych Uniwersytetu Teksańskiego w Austin (Sinclair, Pierce i Matzner 1999) wykorzystują różne techniki uczenia maszynowego, takie jak maszyna stanów skończonych, drzewo decyzyjne i GA, do generowania reguł sztucznej inteligencji dla IDS. Jedno połączenie sieciowe i związane z nim zachowanie można przetłumaczyć na regułę oceniającą, czy połączenie w czasie rzeczywistym jest uważane za włamanie. Reguły te mogą być modelowane jako chromosomy wewnątrz populacji. Populacja ewoluuje do momentu spełnienia kryteriów oceny. Wygenerowany zestaw reguł może być wykorzystany jako wiedza wewnątrz IDS do oceny, czy połączenie sieciowe i związane z nim zachowania są potencjalnymi włamaniami (Sinclair, Pierce i Matzner 1999). Laboratorium COAST na Uniwersytecie Purdue (Crosbie i Spafford, 1995) wdrożyło IDS przy użyciu autonomicznych agentów (czujników bezpieczeństwa) i zastosowało techniki sztucznej inteligencji do ewolucji algorytmów genetycznych. Agenci są modelowani jako chromosomy, a wewnętrzny ewaluator jest używany wewnątrz każdego agenta (Crosbie i Spafford, 1995).

W opisanych powyżej podejściach IDS może być postrzegany jako system oparty na regułach (RBS), a GA może być postrzegany jako narzędzie pomagające generować wiedzę dla RBS. Podejścia te mają pewne wady. W celu wykrycia inwazyjnych zachowań w sieci lokalnej, połączenia sieciowe powinny być wykorzystywane do definiowania zachowań normalnych i anomalnych. Czasami atak może być tak prosty, jak skanowanie dostępnych portów na serwerze lub schemat zgadywania hasła. Zazwyczaj jednak są one złożone i generowane przez zautomatyzowane narzędzia, które są swobodnie dostępne w Internecie. Przykładem może być koń trojański lub backdoor, który może działać przez pewien czas lub może być inicjowany z różnych lokalizacji. Aby wykryć takie włamania, zarówno czasowe, jak i przestrzenne informacje o ruchu sieciowym powinny zostać uwzględnione w zestawie reguł. Obecne aplikacje GA nie zajmują się tymi kwestiami w szerokim zakresie. Niniejszy artykuł pokazuje, w jaki sposób informacje o połączeniach sieciowych mogą być modelowane jako chromosomy i jak parametry algorytmu genetycznego mogą być definiowane w tym zakresie. Kilka przykładów zostało wykorzystanych do zademonstrowania implementacji.

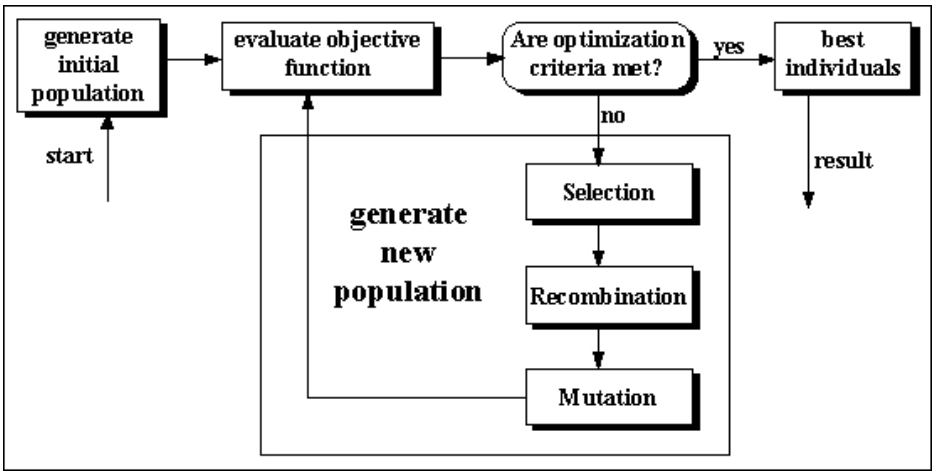
Pozostała część artykułu jest zorganizowana w następujący sposób. Sekcja 2 zawiera krótkie wprowadzenie do algorytmu genetycznego. Sekcja 3 opisuje szczegółową implementację zastosowania algorytmu genetycznego do wykrywania włamań. W sekcji 4 omówiono architekturę proponowanej implementacji. Sekcja 5 przedstawia wnioski i przyszłe prace.

2. Wprowadzenie do algorytmu genetycznego

Algorytm genetyczny to rodzina modeli obliczeniowych opartych na zasadach ewolucji i doboru naturalnego. Algorytmy te przekształcają problem w określonej dziedzinie w model przy użyciu struktury danych podobnej do chromosomu i ewoluują chromosomy przy użyciu operatorów selekcji, rekombinacji i mutacji. Zakres zastosowań algorytmów genetycznych jest dość szeroki (Sinclair, Pierce i Matzner 1999; patrz także Whitley, 1994). W zastosowaniach związanych z bezpieczeństwem komputerowym jest on głównie wykorzystywany do znajdowania optymalnych rozwiązań dla określonego problemu.

Proces algorytmu genetycznego zwykle rozpoczyna się od losowo wybranej populacji chromosomów. Chromosomy te są reprezentacją problemu, który ma zostać rozwiązany. Zgodnie z atrybutami problemu, różne pozycje każdego chromosomu są kodowane jako bity, znaki lub liczby. Pozycje te są czasami określane jako *geny* i są zmieniane losowo w pewnym zakresie podczas ewolucji. Zbiór chromosomów na danym etapie ewolucji nazywany jest *populacją*. *Funkcja oceny* jest używana do obliczania "dobroci" każdego chromosomu. Podczas oceny, dwa podstawowe operatory, *krzyżowanie* i *mutacja*, są używane do symulacji naturalnej reprodukcji i mutacji gatunków. Wybór chromosomów do przetrwania i kombinacji jest ukierunkowany na najlepiej dopasowane chromosomy.

Rysunek 1 przedstawia strukturę prostego algorytmu genetycznego. Zaczyna się od losowo wygenerowanej populacji, ewoluuje poprzez selekcję, rekombinację (krzyżowanie) i mutację. Wreszcie, najlepszy osobnik (chromosom) jest wybierany jako wynik końcowy po spełnieniu kryterium optymalizacji (Pohlheim, 2001).



Rysunek 1. Struktura prostego algorytmu genetycznego (Pohlheim, 2001)

Algorytm genetyczny jest ogólnie dość prosty, ale w większości przypadków może być złożony. Na przykład, podczas operacji krzyżowania może występować krzyżowanie jednopunktowe lub nawet krzyżowanie wielopunktowe. Istnieją również równoległe implementacje algorytmów genetycznych. Czasami szereg parametrów (na przykład szybkość mutacji, szybkość krzyżowania, rozmiar populacji, rozmiar chromosomu, liczba ewolucji lub *pokoleń* oraz sposób przeprowadzania selekcji) należy wziąć pod uwagę przy określonym procesie selekcji. Ostatecznym celem jest przeszukanie przestrzeni rozwiązań w stosunkowo krótkim czasie (Pohlheim, 2001).

3. Algorytm genetyczny zastosowany do wykrywania włamań

Zastosowanie algorytmu genetycznego do wykrywania włamań wydaje się być obiecującym obszarem. W tej sekcji omówimy motywację i szczegóły implementacji.

3.1 Przegląd

Algorytmy genetyczne mogą być wykorzystywane do tworzenia prostych reguł dla ruchu sieciowego (Sinclair, Pierce i Matzner 1999). Reguły te są wykorzystywane do rozróżniania normalnych połączeń sieciowych od połączeń anomalnych. Te anomalne połączenia odnoszą się do zdarzeń z prawdopodobieństwem włamania. Reguły przechowywane w bazie reguł mają zazwyczaj następującą postać (Sinclair, Pierce i Matzner 1999):

if { condition } then { act }

W przypadku problemów, które przedstawiliśmy powyżej, *warunek* zwykle odnosi się do zgodności między bieżącym połączeniem sieciowym a regułami w IDS, takimi jak źródłowe i docelowe adresy IP i numery portów (używane w protokołach sieciowych TCP/IP), czas trwania połączenia, używany protokół itp. Pole *act* zwykle odnosi się do akcji zdefiniowanej przez zasady bezpieczeństwa w organizacji, takiej jak zgłaszanie alertu administratorowi systemu, zatrzymywanie połączenia, rejestrowanie komunikatu w plikach audytu systemu lub wszystkie powyższe. Na przykład reguła może być zdefiniowana jako:

if {połączenie ma następujące informacje: źródłowy adres IP 124.12.5.18; docelowy adres IP: 130.18.206.55; numer portu docelowego: 21; czas połączenia: 10,1 sekundy }
następnie {zatrzymaj połączenie}

Regułę tę można wyjaśnić w następujący sposób: jeśli istnieje żądanie połączenia sieciowego ze źródłowym adresem IP 124.12.5.18, docelowym adresem IP 130.18.206.55, docelowym portem numer 21 i czasem połączenia 10,1 sekundy, to zatrzymaj nawiązywanie tego połączenia. Wynika to z faktu, że adres IP 124.12.5.18 jest rozpoznawany przez IDS jako jeden z adresów IP znajdujących się na czarnej liście; dlatego każde żądanie usługi zainicjowane z tego adresu jest odrzucane.

Ostatecznym celem zastosowania GA jest wygenerowanie reguł, które pasują tylko do anomalnych połączeń. Reguły te są testowane na połączeniach historycznych i wykorzystywane do filtrowania nowych połączeń w celu znalezienia podejrzanego ruchu sieciowego.

W tej implementacji ruch sieciowy wykorzystywany do GA jest wstępnie sklasyfikowanym zestawem danych, który odróżnia normalne połączenia sieciowe od anomalii. Ten zestaw danych jest gromadzony przy użyciu snifferów sieciowych (programów służących do rejestrowania ruchu sieciowego bez robienia czegoś szkodliwego), takich jak Tcpdump (http://www.tcpdump.com) lub Snort (http://www.snort.com). Zbiór danych jest ręcznie klasyfikowany w oparciu o wiedzę ekspertów. Jest on używany do oceny sprawności podczas wykonywania GA. Uruchamiając GA tylko z niewielkim zestawem losowo wygenerowanych reguł, możemy wygenerować większy zestaw danych, który zawiera reguły dla IDS. Reguły te są "wystarczająco dobrymi" rozwiązaniami dla GA i mogą być wykorzystywane do filtrowania nowego ruchu sieciowego.

3.2 Reprezentacja danych

Aby w pełni wykorzystać podejrzany poziom, musimy zbadać wszystkie pola związane z konkretnym połączeniem sieciowym. Dla uproszczenia bierzemy pod uwagę tylko niektóre oczywiste atrybuty dla każdego połączenia. Definicję reguł (dla protokołów TCP/IP) przedstawiono w tabeli 1.

Odpowiednią regułę dla atrybutu "Przykładowa wartość" w tabeli 1 można przetłumaczyć jako:

if {połączenie ma następujące informacje: źródłowy adres IP 209.11.?????; docelowy adres IP: 130.18.176+?.??; numer portu źródłowego: 42335; numer portu docelowego: 80; czas połączenia: 482 sekundy; połączenie zostało przerwane przez inicjatora; używany protokół to TCP; inicjator wysłał 7320 bajtów danych; a respondent wysłał 38891 bajtów danych }
następnie {zatrzymaj połączenie}

Tabela 1. Definicja reguły dla połączenia i zakres wartości każdego pola

Atrybut	Zakres wartości	Przykładowe wartości	Opisy
Źródłowy adres IP	0.0.0.0~255.255.255.255	d1.0b.**.** (209.11.????)	Podsieć z adresem IP 209.11.0.0 do 209.11.255.255
Docelowy adres IP	0.0.0.0~255.255.255.255	82.12.b*.** (130.18.176+?.??)	Podsieć z adresem IP 130.18.176.0 do 130.18.255.255
Numer portu źródłowego	0~65535	42335	Numer portu źródłowego połączenia
Numer portu docelowego	0~65535	00080	Numer portu docelowego, wskazuje jest to usługa http
Czas trwania	0~99999999	00000482	Czas trwania połączenia wynosi 482 sekundy
Stan	1~20	11	Połączenie jest przerywane przez inicjatora, dla użytek wewnętrzny
Protokół	1~9	2	Protokół dla tego połączenia to TCP
Liczba wysłanych bajtów Autor	0~9999999999	0000007320	Inicjator wysyła 7320 bajtów danych
Liczba wysłanych bajtów przez Responder	0~9999999999	0000038891	Respondenci wysyłają 38891 bajtów danych

Możemy przekonwertować powyższy przykład do postaci chromosomu, jak opisano na rysunku 3.

(d, 1, 0, b, -1, -1, -1, -1, 8, 2, 1, 2, b, -1, -1, -1, 4, 2, 3, 3,
5, 0, 0, 0, 8, 0, 0, 0, 0, 0, 0, 4, 8, 2, 1, 1, 2, 0, 0, 0,
0, 0, 0, 7, 3, 2, 0, 0, 0, 0, 0, 0, 0, 3, 8, 8, 9, 1)

Rysunek 3. Struktura chromosomu dla przykładu z tabeli 1

W sumie w każdym chromosomie znajduje się pięćdziesiąt siedem genów. Dla uproszczenia używamy szesnastkowych reprezentacji adresów IP. Regułę można wyjaśnić w następujący sposób: jeśli połączenie sieciowe ze źródłowym adresem IP 209.11.???? (209.11.0.0 ~ 209.11.255.255), docelowym adresem IP 130.18.176.??? (130.18.176.0 ~ 130.18.255.255),
numer portu źródłowego 42335, numer portu docelowego 80, czas trwania 482 sekundy, kończy się stanem 11 (połączenie zakończone przez inicjatora), używa protokołu typu 2 (TCP), a inicjator wysyła 7320 bajtów danych, respondenci wysyłają 38891 bajtów danych, to jest to podejrzane zachowanie i może być zidentyfikowane jako potencjalne włamanie. Rzeczywista ważność tej reguły zostanie sprawdzona poprzez dopasowanie zestawu danych historycznych składającego się z połączeń oznaczonych jako anomalne lub normalne. Jeśli reguła jest w stanie znaleźć anomalne zachowanie, premia zostanie przyznana bieżącemu chromosomowi. Jeśli reguła pasuje do normalnego połączenia, do chromosomu zostanie zastosowana kara. Oczywiście żadna pojedyncza reguła nie może być użyta do oddzielenia wszystkich anomalnych połączeń od normalnych. Populacja musi ewoluować, aby znaleźć optymalny zestaw reguł.

W przykładzie pokazanym w tabeli 1 użyto niektórych symboli wieloznacznych (znak "*" i znak "?"), a odpowiadające im geny w chromosomie są wyświetlane jako -1. Te symbole wieloznaczne są używane do reprezentowania odpowiedniego zakresu określonych wartości (Crosbie i Spafford, 1995). Jest to przydatne podczas reprezentowania bloku sieci (zakresu adresów IP lub numerów portów) w regule. Gdy informacje przestrzenne zostaną uwzględnione w regułach, możliwości IDS mogą zostać znacznie poprawione, ponieważ włamanie może zostać zainicjowane z wielu różnych lokalizacji. Włączenie czasu trwania połączenia sieciowego do chromosomu zapewnia uwzględnienie informacji czasowych dla połączeń sieciowych. Maksymalna wartość czasu trwania wynosi 99999999 sekund, czyli ponad rok. Jest to pomocne w identyfikacji włamań, ponieważ złożone włamania mogą obejmować godziny, dni, a nawet miesiące.

Algorytm genetyczny rozpoczyna się od populacji z losowo wybranymi regułami. Populacja może ewoluować przy użyciu operatorów krzyżowania i mutacji. Ze względu na skuteczność funkcji oceny, kolejne populacje są ukierunkowane na reguły, które pasują do inwazyjnych połączeń. Ostatecznie, gdy algorytm się zatrzymuje, reguły są wybierane i dodawane do bazy reguł IDS.

3.3 Parametry w algorytmie genetycznym

Istnieje wiele parametrów, które należy wziąć pod uwagę przy stosowaniu GA. Każdy z tych parametrów ma duży wpływ na skuteczność algorytmu genetycznego. Omówimy metodologię i powiązane parametry w następnej sekcji.

Funkcja oceny

Funkcja oceny jest jednym z najważniejszych parametrów algorytmu genetycznego. Proponowana implementacja różni się od schematu stosowanego przez (Crosbie i Spafford, 1995) tym, że definicja obliczeń *wyniku* i *kondycji* jest inna. Do obliczenia funkcji oceny wykorzystywane są następujące kroki.

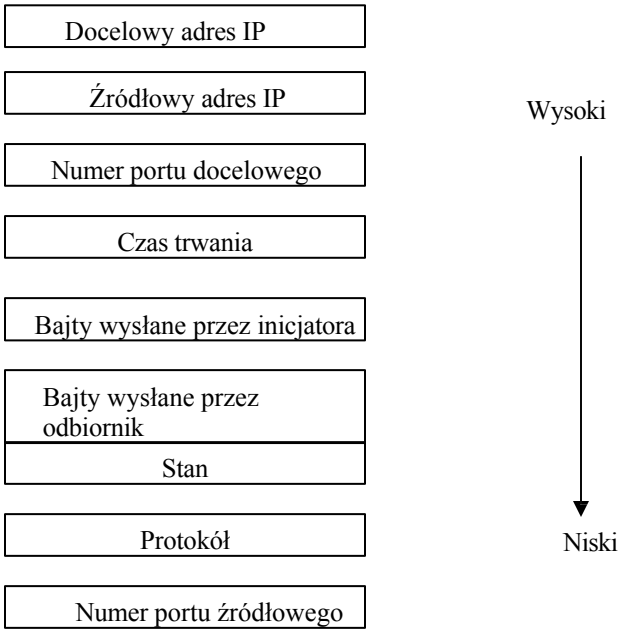
Najpierw obliczany jest ogólny *wynik* na podstawie tego, czy pole połączenia pasuje do wstępnie sklasyfikowanego zestawu danych, a następnie mnożona jest waga tego pola. Wartość *dopasowania jest* ustawiana na 1 lub 0.

$$Wynik= \sum_{i=1}^{57} Dopasowany_{*Waga}$$

Kolejność wartości wag w funkcji pokazano na rysunku 4. Kolejności te są podzielone na kategorie zgodnie z różnymi polami w rekordzie połączenia zgłaszanymi przez sniffery sieciowe. Dlatego wszystkie geny reprezentujące

pole docelowego adresu IP mają taką samą wagę. Rzeczywiste wartości można precyzyjnie dostosować w czasie wykonywania. Geny

Podstawową ideą stojącą za tą kolejnością jest znaczenie różnych pól w pakietach TCP/IP. Schemat ten jest prosty i intuicyjny. Docelowy adres IP jest celem włamania, podczas gdy źródłowy adres IP jest jego inicjatorem. Są to najważniejsze informacje potrzebne do przechwycenia włamania. Numer portu docelowego wskazuje aplikacjom, że system docelowy jest uruchomiony (na przykład usługa FTP zwykle działa na porcie 21). Niektóre adresy IP są bardziej prawdopodobnymi celami włamań - na przykład adresy IP domen wojskowych. Informacje specyficzne dla domeny są mniej ważne w porównaniu ze źródłowymi adresami IP. Inne parametry, takie jak czas trwania, bajty wysłane przez nadawcę, bajty wysłane przez odbiorcę i stan są zwykle mniej ważne niż powyższe pola, ale nadal są przydatne. Pola protokołu i numeru portu źródłowego są zwykle zbędne i służą do identyfikacji niektórych konkretnych włamań.



Rysunek 4. Kolejność wag dla pól w funkcji oceny

Bezwzględna różnica między *wynikiem* chromosomu a rzeczywistym *podejrzanym poziomem* jest następnie obliczana przy użyciu następującego równania. *Poziom* suspicious_level to próg, który wskazuje, w jakim stopniu dwa połączenia sieciowe są uważane za "pasujące". Rzeczywista wartość *suspicious_level odzwierciedla* obserwacje z danych historycznych.

$$\Delta = | wynik - podejrzany\_poziom |$$

Gdy wystąpi niezgodność, wartość kary jest obliczana na podstawie bezwzględnej różnicy. *Ranking* w równaniu wskazuje, czy włamanie jest łatwe do zidentyfikowania.

$$kara = \left(\frac{\Delta * ranking}{100}\right)$$

Fitness chromosomu jest obliczany przy użyciu powyższej *kary*: *fitness*

$$= 1 - kara$$

Oczywiście zakres wartości fitness wynosi od 0 do 1. Definiując ocenę, uwzględniliśmy zarówno czasowe, jak i przestrzenne informacje potrzebne do identyfikacji włamań do sieci.

Krzyżowanie i mutacja

Tradycyjne algorytmy genetyczne zostały wykorzystane do identyfikacji i konwergencji populacji hipotez kandydujących do jednego globalnego optimum. W przypadku tego problemu, zestaw reguł jest potrzebny jako podstawa dla IDS. Jak wspomniano wcześniej, nie ma sposobu, aby jednoznacznie określić, czy połączenie sieciowe jest normalne, czy anomalne, używając tylko jednej reguły. Do zidentyfikowania niepowiązanych anomalii potrzebnych jest wiele reguł, co oznacza, że kilka dobrych reguł jest bardziej skutecznych niż jedna najlepsza reguła (Sinclair, Pierce i Matzner 1999). Innym powodem znalezienia wielu reguł jest to, że ponieważ istnieje tak wiele możliwości połączeń sieciowych, mały zestaw reguł będzie dalece niewystarczający.

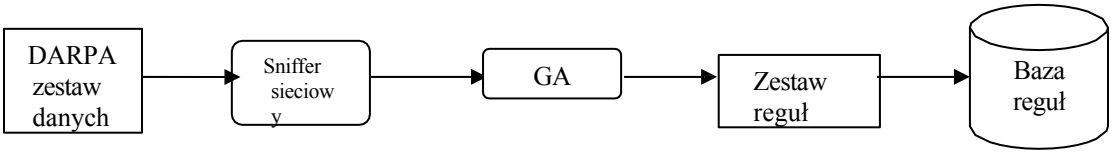
Korzystając z algorytmu genetycznego, musimy znaleźć lokalne maksima (zestaw "wystarczająco dobrych" rozwiązań) w przeciwieństwie do globalnego maksimum (najlepsze rozwiązanie) (Sinclair, Pierce i Matzner 1999). Techniki *nichingu można wykorzystać* do znalezienia wielu lokalnych maksimów (Miller i Shaw, 1996; patrz także Sinclair, Pierce i Matzner 1999). Opiera się ona na analogii do natury, w której w każdym środowisku istnieją różne podprzestrzenie (*nisze*), które mogą wspierać różne rodzaje życia. W podobny sposób algorytm genetyczny może utrzymać różnorodność każdej populacji w domenie *multimodalnej,* która odnosi się do domen wymagających identyfikacji wielu optymalnych rozwiązań. Dwie podstawowe metody, *tłumienie* i *dzielenie,* mogą być używane do *niszowania*. Metoda *tłumienia wykorzystuje* najbardziej podobnego członka do zastąpienia, aby spowolnić populację do zbiegnięcia się do jednego punktu w kolejnych pokoleniach. Metoda *dzielenia* zmniejsza kondycję osobników, które mają bardzo podobnych członków i zmusza osobniki do ewolucji do innych lokalnych maksimów, które mogą być mniej zaludnione. Metryki podobieństwa stosowane w tych technikach mogą być fenotypowe i genotypowe, takie jak odległość Hamminga między reprezentacjami bitowymi, lub fenotypowe, takie jak relacja między dwoma połączeniami sieciowymi w tym problemie. Ta ostatnia jest bardziej odpowiednia do znajdowania reguł używanych w IDS. Wadą tego podejścia jest to, że wymaga ono większej wiedzy specyficznej dla danej dziedziny (Miller i Shaw, 1996; patrz także Sinclair, Pierce i Matzner 1999).

Operacja mutacji powinna mieć znaczenie podczas ewolucji. Na przykład, każdy segment adresu IP nie powinien przekraczać 255 (reprezentacja dziesiętna). Mutacje powinny być wykonywane zgodnie z wymaganiami określonymi w Tabeli 1. Ograniczenia te mogą być egzekwowane poprzez zdefiniowanie odpowiednich reguł mutacji.

Inne parametry

Istnieją również inne parametry, które należy wziąć pod uwagę, takie jak współczynnik mutacji, współczynnik krzyżowania, liczba populacji i liczba pokoleń. Parametry te należy dostosować do środowiska aplikacji systemu i polityki bezpieczeństwa organizacji.

4. Architektura systemu



Rysunek 5. Architektura zastosowania GA do wykrywania włamań

Rysunek 5 przedstawia strukturę tej implementacji. Musimy zebrać wystarczającą ilość danych historycznych, które obejmują zarówno normalne, jak i anomalne połączenia sieciowe. Dobrym wyborem jest zestaw danych MIT Lincoln Laboratory (http://www.ll.mit.edu/) do testowania IDS, który jest reprezentowany w formacie binarnym *Tcpdump.* Jest to pierwsza część architektury systemu. Ten zestaw danych jest analizowany przez sniffery sieciowe, a wyniki są przekazywane do GA w celu oceny sprawności. Następnie GA jest wykonywany i generowany jest zestaw reguł. Reguły te są przechowywane w bazie danych i wykorzystywane przez IDS.

5. Wnioski i przyszłe prace

W niniejszym artykule omówiono metodologię zastosowania algorytmu genetycznego w technikach wykrywania włamań do sieci. Omówiono krótki przegląd systemu wykrywania włamań (IDS), algorytmu genetycznego i powiązanych technik wykrywania. Przedstawiono również architekturę systemu. Szczegółowo omówiono czynniki wpływające na GA. Ta implementacja algorytmu genetycznego jest wyjątkowa, ponieważ uwzględnia zarówno czasowe, jak i przestrzenne informacje o

połączenia sieciowe podczas kodowania problemu; dlatego powinien być bardziej pomocny w identyfikacji anomalnych zachowań sieci.

Przyszłe prace obejmują stworzenie standardowego zestawu danych testowych dla algorytmu genetycznego zaproponowanego w niniejszym artykule i zastosowanie go w środowisku testowym. Szczegółowa specyfikacja parametrów do rozważenia dla algorytmu genetycznego powinna zostać określona podczas eksperymentów. Połączenie wiedzy z różnych czujników bezpieczeństwa w standardową bazę reguł jest kolejnym obiecującym obszarem w tej pracy.

Podziękowania

Autor pragnie podziękować Ambareen Siraj z Centrum Badań nad Bezpieczeństwem Komputerowym (CCSR) na Wydziale Informatyki i Inżynierii Uniwersytetu Stanowego Mississippi za recenzję i sugestie podczas realizacji tego artykułu.

ODNIESIENIA

Bezroukov, Nikolai. 19 lipca 2003. "Wykrywanie włamań (zagadnienia ogólne)." Softpanorama: Open Source Software Educational Society. Nikolai Bezroukov. URL: http://www.softpanorama.org/Security/intrusion_detection.shtml (30 października 2003).

Bridges, Susan i Rayford B. Vaughn. 2000. "Intrusion Detection Via Fuzzy Data Mining." *In Proceedings of 12th Annual Canadian Information Technology Security Symposium*, pp. 109-122. Ottawa, Kanada.

Crosbie, Mark i Gene Spafford. 1995. "Zastosowanie programowania genetycznego do wykrywania włamań". *In Proceedings of 1995 AAAI Fall Symposium on Genetic Programming*, pp. 1-8. Cambridge, Massachusetts. URL: http://citeseer.nj.nec.com/crosbie95applying.html (30 października 2003).

Graham, Robert. Mar. 21, 2000. "FAQ: Sieciowe systemy wykrywania włamań". RobertGraham.com Homepage. Robert Graham. URL: http://www.robertgraham.com/pubs/network-intrusion-detection.html (30 października 2003).

Jones, Anita. K. i Robert. S. Sielken. 2000. "Wykrywanie włamań do systemów komputerowych: A Survey." Raport techniczny. Department of Computer Science, University of Virginia, Charlottesville, Virginia.

Li, Wei. 2002. "Integracja czujników bezpieczeństwa z inteligentnym systemem wykrywania włamań (IIDS) w środowisku klastrowym". Raport z projektu magisterskiego. Department of Computer Science, Mississippi State University.

McHugh, John, 2001. "Intrusion and Intrusion Detection." Raport techniczny. CERT Coordination Center, Software Engineering Institute, Carnegie Mellon University.

Miller, Brad. L. i Michael J. Shaw. 1996. "Genetic Algorithms with Dynamic Niche Sharing for Multimodal Function Optimization". *In Proceedings of IEEE International Conf. on Evolutionary Computation*, s. 786-791. Uniwersytet Nagoya, Japonia.

Paxson, Vern. 1998. "Bro: A System for Detecting Network Intruders in Real-time". *In Proceedings of 7th USENIX Security Symposium*, pp. 31-51. San Antonio, Texas.

Pohlheim, Hartmut. 30 października 2003. "Algorytmy genetyczne i ewolucyjne: Zasady, metody i algorytmy". Genetic and Evolutionary Algorithm Toolbox. Hartmut Pohlheim. URL: http://www.geatbx.com/docu/algindex.html.

Roesch, Martin. 7-12 listopada 1999. "Snort - Lightweight Intrusion Detection for Networks." *In Proceedings of 13th Systems Administration Conf. (LISA '99),* pp. 229-238.Seattle, Washington.

Sinclair, Chris, Lyn Pierce i Sara Matzner. 1999. "An Application of Machine Learning to Network Intrusion Detection". *In Proceedings of 1999 Annual Computer Security Applications Conf. (ACSAC)*, pp. 371-377. Phoenix, Arizona. URL: http://www.acsac.org/1999/papers/fri-b-1030-sinclair.pdf (30 października 2003).

Whitley, Darrell. 1994. "A Genetic Algorithm Tutorial." *Statistics and Computing* 4: 65-85.