Symulacja w Badaniach i Rozwoju
Vol. 6, No. 2/2015

Rafał KASPRZYK, Marcin PAŹ
Zbigniew TARAPATA
Wojskowa Akademia Techniczna, Wydział Cybernetyki
00-908 Warszawa, ul. Kaliskiego 2
E-mail: rafal.kasprzyk@wat.edu.pl, mpaz@wat.edu.pl
zbigniew.tarapata@wat.edu.pl

Modelowanie i symulacja cyberzagrożeń typu botnet

1 Wstęp

Wyzwania związane z cyberzagrożeniami stanowią obecnie zasadniczą treść koncepcji preparacyjnych i operacyjnych strategii bezpieczeństwa większości państw oraz organizacji międzynarodowych [11]. Opracowywane strategie podkreślają potrzebę zwrócenia szczególnej uwagi na nową przestrzeń, w której funkcjonują współczesne społeczeństwa – tzw. cyberprzestrzeń. W wielu ośrodkach naukowych, jak również w armiach większości państw świata prowadzone są badania mające na celu opracowanie metod i specjalizowanych narzędzi zwiększających efektywność wykrywania, przeciwdziałania i neutralizacji skutków cyberzagrożeń [1]. Potrzeba wypracowania tego typu metod i narzędzi wynika z coraz większego uzależnienia administracji państwowej, instytucji prywatnych i całego społeczeństwa od prawidłowego funkcjonowania sieci komunikacyjnych i systemów informatycznych.

Internet, będący fundamentem cyberprzestrzeni, coraz częściej postrzegany jest jako infrastruktura niezwykle wrażliwa, od funkcjonowania której zależy bezpieczeństwo państwa, tak w sferze społecznej, gospodarczej, jak i militarnej [1]. Według raportu *We are social media* Internet posiada około 3,01 mld użytkowników, co stanowi ponad 42% wszystkich ludzi na świecie. Ponadto można dostrzec trend wskazujący na ciągły wzrost liczby użytkowników Internetu. Aby przeprowadzić skuteczny atak na tę infrastrukturę, nie trzeba mobilizować sił zbrojnych. Człowiek wyposażony w standardowe technologie komputerowe, posiadający odpowiednią wiedzę, może przeprowadzić cyberatak o skutkach wręcz katastrofalnych dla współczesnego systemu polityczno-gospodarczego. Dlatego niezwykle ważne jest, aby w porę wykryć, przeciwdziałać i neutralizować skutki tego typu zagrożeń, będących w ogólnym znaczeniu zdarzeniami w cyberprzestrzeni, które mogą powodować niepożądane skutki, powodujące szkody w systemach zarówno użytkowników indywidualnych, jak i organizacji.

2 Cyberzagrożenia typu botnet

Analizując dane historyczne dotyczące cyberataków [5], można dostrzec, iż w większości przypadków ich źródłem są sieci typu botnet, które można najprościej zdefiniować jako grupę zainfekowanych złośliwym oprogramowaniem (ang. *malware*) komputerów (ang. *zombies, bots*) i dającą jej twórcy określony poziom kontroli nad zainfekowanymi maszynami [18]. Liczba zainfekowanych komputerów w ramach jednego botnetu zwykle waha się od kilku do nawet setek tysięcy botów. Największe

zaobserwowane sieci obejmowały nawet po kilka milionów zarażonych komputerów. Taka armia botów pozwala na przeprowadzanie szeregu ataków z ich wykorzystaniem bez wiedzy użytkowników. Niski koszt utrzymania botnetu i coraz większa dostępność wiedzy wymaganej do jego zarządzania przyczyniają się do wzrostu popularności, a w konsekwencji, liczby botnetów.

Tabela 1. Udział botnetów w Polsce wg CERT Polska, Raport 2014

Table 1. A listing of botnet activities in Poland in 2014. Source: Raport 2014, CERT Polska

Lp.	Nazwa Botnetu	Liczba adresów IP	Udział procentowy
1	*Conficker*	62 221	21,19%
2	*ZeroAccess*	32 460	11,57%
3	*Zeus (w tym Citadel)*	25 311	9,03%
4	*Sality*	14 003	4,99%
5	*Zeus GameOver*	12 513	4,46%
6	*Ircbot*	10 768	3,84%
7	*Bankpatch*	6 086	2,17%
8	*Banatrix*	5 385	1,92%
9	*Virut*	4 014	1,43%
10	*Kelihos*	3 922	1,40%
	Pozostałe	103 750	37,00%

Botnety najczęściej wykorzystywane są do [16, 18]:

- Rozsyłania niechcianej poczty (SPAM) – jest to najpowszechniejszy sposób wykorzystania sieci botnet, pozwalający na wysłanie milionów wiadomości, w bardzo krótkim czasie. Szacuje się, że 80% spamu jest wysyłane przez komputery *zombie*. Adresy wykorzystywane do rozsyłania spamu zamieszczane są na czarnych listach, a same wiadomości przychodzące z tych adresów są blokowane przez serwery pocztowe. Użycie botnetu pozwala obejść ten problem, wysyłając spam z adresów e-mail należących do właścicieli zainfekowanych maszyn *zombie*.
- Przeprowadzenia ataków typu DDoS (ang. *Distributed Denial of Service*), czyli zablokowania dostępu do usług w sieci Internet poprzez generowanie sztucznego ruchu. W konsekwencji atakowany serwer zostaje przeciążony i staje się niedostępny. W zamian za zatrzymanie ataku cyberprzestępcy żądają zwykle pieniędzy. Niestety w czasach, kiedy wiele firm funkcjonuje, bazując na sieci Internet, właściciele firm często płacą okup, rezygnując jednocześnie z pomocy organów ścigania.
- Kradzieży poufnych i prywatnych danych, np. numery kart kredytowych, informacje umożliwiające uzyskanie dostępu do kont bankowych, szeroki wachlarz loginów i haseł. Zebrane dane są następnie wykorzystywane do dalszych nielegalnych działań, w tym mogą stać się przedmiotem sprzedaży.

- Generowania fałszywych kliknięć na reklamy online wystawiane w systemie PPC (ang. *Pay-Per-Click*) przez agencje reklamowe na różnych stronach internetowych. Właściciele tych stron pobierają prowizję od każdego kliknięcia. Przy pomocy sieci *zombie* możliwe jest wygenerowanie w ciągu jednego dnia tysięcy unikatowych kliknięć - każde z innej maszyny, aby nie wzbudzać podejrzeń. W ten sposób pieniądze wydane na kampanie reklamowe trafiają do kieszeni właściciela strony.

Botnety stały się źródłem dochodów dla wielkich grup cyberprzestępczych, pozwalając osiągać ogromne zyski z ich nielegalnej działalności. Dla przykładu botnet *DNSChanger* [14] liczący ponad 4 mln botów - wykorzystywany do wstrzykiwania reklam - w ciągu 5 lat działania przyniósł dochód rzędu 14 mln $, natomiast botnet *Storm* [14], którego wielkość szacowano na ok. 5 mln botów - wykorzystywany do rozsyłania SPAMu - każdego roku swego działania przynosił dochód rzędu 3,5 mln $. Dodatkowo niebezpieczeństwo związane z wykorzystaniem botnetów znacznie rośnie, gdy weźmie się pod uwagę możliwość wynajęcia istniejącej już sieci botnet do prowadzenia ataków. Szacunkowe koszty są następujące [14]: całodniowy atak DDoS to $30 - $70; email SPAM - $10/1 mln wiadomości; zakup 2000 botów: $200; zakup botnetu zdolnego wykonać efektywny atak DDoS - $700; zakup 1000 odwiedzin strony WWW to $7 - $15.

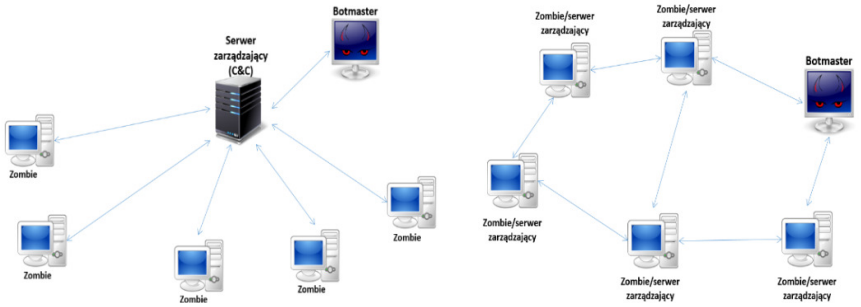
3 Klasyfikacja sieci botnet

Botnety najczęściej klasyfikuje się ze względu na ich architekturę [18] oraz protokoły sieciowe [18] wykorzystywane do komunikacji pomiędzy zainfekowanymi komputerami. Klasyfikując botnety ze względu na architekturę, wyróżnia się botnety scentralizowane oraz zdecentralizowane.

W modelu **scentralizowanym** wszystkie zainfekowane komputery komunikują się z serwerem zarządzającym zwanym C&C (*Command and Control).* Każdy zainfekowany komputer po nawiązaniu komunikacji z C&C rejestrowany jest w bazie danych, która przechowuje między innymi dane dotyczące adresów IP oraz lokalizacji komputerów stanowiących botnet. Za pomocą panelu sterowania C&C właściciel botnetu (ang. *bootmaster*) może wydawać polecenia wszystkim bądź wybranym komputerom *zombie*, spełniającym pewne kryteria (np. ze względu na lokalizację). Botnety scentralizowane są proste w implementacji oraz w późniejszym zarządzaniu. Z drugiej jednak strony, ze względu na wyróżnioną rolę C&C, stosunkowo łatwo je zneutralizować, ponieważ wystarczy unieszkodliwić lub przejąć serwer C&C, zarządzający całością sieci botnet.

W **modelu zdecentralizowanym - P2P** (ang. *peer-to-peer*) sieć botnet posiada strukturę rozproszoną, w ramach której każdy z komputerów *zombie* może pełnić rolę serwera zarządzającego. W architekturze P2P wystarczy, aby *bootmaster* miał dostęp do dowolnego komputera *zombie*. Idea takiego podejścia polega na tym, że pojedynczy bot posada listę maszyn „sąsiednich" i w przypadku otrzymania komunikatu przesyła go dalej do owych „sąsiadów". W ten sposób możliwe staje się rozpropagowanie polecenia w całej sieci botnet bez wyróżniania roli serwera C&C. W praktyce tworzenie zdecentralizowanych botnetów jest dość trudne. Każdemu nowo zainfekowanemu komputerowi należy dostarczyć listę botów - „sąsiadów", z którymi połączy się w sieć botnet. Zwalczanie zdecentralizowanych botnetów jest jednak znacznie trudniejsze

niż zwalczanie scentralizowanych sieci. Aktywny botnet P2P nie posiada żadnego wyróżnionego komputera *zombie*, którego przejęcie pozwoli na zneutralizowanie sieci botnet jako całości. Każdy z komputerów *zombie* może pełnić rolę centrum zarządzającego.



Rys. 1. Scentralizowany i zdecentralizowany model sieci botnet
Źródło: Opracowanie własne

Fig. 1. Centralized and decentralized botnet network model
Source: Own elaboration

Czasami tworzone są sieci botnet o architekturze mieszanej. Podejście takie ułatwia przekazywanie listy „sąsiadów" nowo zainfekowanym komputerom, które w pierwszej kolejności komunikują się z serwerem C&C celem otrzymania listy „sąsiadów", a następnie przełączane są na komunikację typu P2P. Tę mieszaną architekturę również kategoryzuje się zwykle jako model zdecentralizowany, mimo że na pewnym etapie „życia" botnetu wykorzystywany jest serwer C&C.

Wszystkie połączenia sieciowe opierają się na protokołach określających reguły interakcji pomiędzy urządzeniami w sieci. Biorąc pod uwagę tę cechę, można wyróżnić następujące klasy botnetów:

- Zorientowane na IRC (ang. *Internet Relay Chat*) - najczęściej wykorzystywany przez twórców botnetów typ, gdzie każdy zainfekowany komputer łączy się ze wskazanym serwerem IRC. Sterowanie botnetami odbywa się poprzez wydawanie poleceń jako rozmów na dedykowanym prywatnym kanale IRC - podłączone boty nasłuchują tekstu na kanale, a w przypadku rozpoznania tekstu jako komendy przechodzą do jej realizacji.
- Zorientowane na sieć WWW - stosunkowo nowy, ale dość popularny typ botnetów. Bazuje na protokole http działającym na zasadzie wysyłania żądań i odpowiedzi. Cechują się dość trudną wykrywalnością przez systemy ochrony. Bot łączy się z predefiniowanym serwerem, otrzymuje od niego rozkazy i, odpowiadając, przesyła do niego dane.
- Zorientowane na IM (ang. *Instant Messenger*) – dość rzadko spotykany typ, komunikujący się przez komunikatory internetowe, takie jak AOL, MSN, ICQ itd. Stosunkowo niewielka popularność takich botnetów wynika z trudności tworzenia indywidualnego konta IM dla każdej zainfekowanej maszyny.

- Inne, które komunikują się za pośrednictwem własnego protokołu opartego na stosie TCP/IP, tj. wykorzystują tylko protokoły warstwy transportowej, takie jak: TCP, ICMP oraz UDP.

4 Metody rozprzestrzeniania się sieci botnet

Jednym z ważniejszych etapów cyklu życia botnetów jest jego rozprzestrzenianie się. Metody propagacji planowane są już podczas implementacji oprogramowania stanowiącego kod botu. Najczęściej spotykanymi metodami rozprzestrzeniania się sieci botnet, a tym samym infekcji maszyn, są:

- Robaki komputerowe – będące programami rozprzestrzeniającymi się samoistnie i najczęściej wykorzystujące błędy systemów operacyjnych.
- Poczta elektroniczna i komunikatory – poprzez rozsyłanie wiadomości zawierające złośliwy kod, w formie na przykład kartki świątecznej lub okolicznościowej, treści w formacie HTML zawierający odnośnik do programów zawierających złośliwe oprogramowanie, informacji o zmianie danych logowania do konta na jednym z popularnych serwisów itp.
- Pliki pobierane ze stron zawierających nielegalne oprogramowanie (ang. *warez*), które złośliwe oprogramowanie ukrywają pod nazwą plików typu *crack* do popularnych aplikacji oraz gier.
- Portale społecznościowe - tworzone są fałszywe profile, które rozsyłają wiadomości do użytkowników z odnośnikami prowadzącymi do skryptów, z kodem włączającym do botnetu.
- Blackhat SEO *(*ang. *Search Engine Optimization*) - technika zajmująca się dostosowywaniem treści strony WWW i rozmieszczenia słów kluczowych na podstronach serwisu internetowego w celu uzyskania wyższej pozycji indeksu w wyszukiwarkach. Użytkownikowi odwiedzającemu taką witrynę zostaje zainstalowane oprogramowanie włączające go do sieci botnet.

5 Modelowanie sieci botnet

Wraz z rozwojem cyberprzestępstw, których źródłem są botnety, rośnie potrzeba opracowywania modeli, metod i narzędzi do wykrywania, przeciwdziałania i neutralizacji ich skutków. Istnieje kilka poziomów możliwej analizy zjawisk w cyberprzestrzeni [10]:

- Analiza na poziomie hosta (urządzeń) – polega przede wszystkim na budowie świadomości użytkowników o cyberzagrożeniach, instalacji oprogramowania antywirusowego oraz oprogramowania *firewall*, a także na przeprowadzaniu aktualizacji usuwających luki bezpieczeństwa w użytkowanym oprogramowaniu.
- Analiza na poziomie ruchu wchodzącego/wychodzącego – polega na monitorowaniu i analizowaniu ruchu w sieci z wykorzystaniem takich systemów, jak IDS (ang. *Intrusion Detection System*), IPS (ang. *Intrusion Prevention System*).
- Analiza na poziomie struktury sieci botnet i sposobu komunikacji pomiędzy zainfekowanymi komputerami a serwerem C&C – polega na monitorowaniu

funkcjonowania sieci Internet jako całości i możliwa jest do realizacji przez specjalizowane instytucje typu CERT, w szczególności przy założeniu ich wzajemnej współpracy, celem zapewniania cyberbezpieczeństwa w wymiarze państwowym.

W dalszej części artykułu przedstawiony zostanie **szkielet modelu cyberprzestrzeni** rozumianej intuicyjnie jako przestrzeń wytwarzania, gromadzenia, przetwarzania i wymiany informacji, która jest „generowana" przez współpracujące ze sobą systemy teleinformatyczne i byty zewnętrzne (np. ludzie) wchodzące w interakcje z tymi systemami [1]. Cyberprzestrzeń modelowana jest w celu umożliwienia opisu i analizy, w tym symulacji, cyberzagrożeń typu botnet. Model cyberprzestrzeni ma stanowić podstawę do opracowania metod wykrywania, przeciwdziałania i neutralizacji skutków cyberzagrożeń typu botnet. W chwili obecnej model, opracowane metody i skonstruowane narzędzia (środowisko symulacyjne) pozwalają na:

- analizę charakterystyk strukturalnych zidentyfikowanej/hipotetycznej sieci botnet na potrzeby szacowania rozmiaru potencjalnego ataku przeprowadzonego z wykorzystaniem danej sieci botnet (np. wielkość możliwego do wygenerowania ruchu w przypadku ataku DDoS);
- ocenę odporności sieci botnet na zdarzenia przypadkowe (np. aktualizacji oprogramowania antywirusowego przez użytkownika komputera *zombie*) i świadome działania mające na celu zwalczanie/przejęcie sieci botnet (np. wyłączenie komputera zidentyfikowanego jako C&C, wyłączenie komputerów *zombie* położonych w kluczowych miejscach sieci botnet z punktu widzenia jej struktury);
- opis i analizę, w tym symulację, procesu rozprzestrzeniania się złośliwego oprogramowania i ewolucję sieci botnet w cyberprzestrzeni;
- opis i analizę, w tym symulację, skutków wybranych ataków na rzeczywisty/hipotetyczny cel w sieci Internet (np. atak typu DDoS o określonych parametrach na sieć teleinformatyczną badanej organizacji).

Jako model cyberprzestrzeni (ang. *CyberSpace*) proponuje się następujący wektor:

$$CyberSpace(t) = \langle CNet(t), CAs(t), CTs(t), AMs(t), SMs(t) \rangle ,$$

gdzie:

$CNet(t)$ – model opisujący topologię i charakterystyki ilościowe sieci Internet (lub jej fragmentu będącego przedmiotem zainteresowania ze względu na cel modelowania);

$CAs(t)$ – aktorzy cyberprzestrzeni, np. użytkownicy, administratorzy, hakerzy;

$CTs(t)$ – cyberzagrożenia występujące lub potencjalne (np. sieci botnet, złośliwe oprogramowanie);

$AMs(t)$ – metody/mechanizmy ataków będące możliwymi realizacjami cyberzagrożeń (np. atak typu DDoS zrealizowany z wykorzystaniem cyberzagrożenia typu botnet);

$SMs(t)$ – metody/mechanizmy zabezpieczeń elementów składowych sieci Internet (np. instalacja oprogramowania antywirusowego lub firewall, systemy IDS/IPS).

Parametr $t \in T = \{1, 2, 3, \ldots\}$ oznacza zdyskretyzowany czas, gdzie:

T – zbiór dyskretnych chwil.

Sieć [15] Internet modelowana jest jako trójka uporządkowana:

$$CNet(t) = \left\langle G(t) = \left\langle V(t), B(t), I(t) \right\rangle, \{f_i(v,t)\}_{\substack{i \in \{1,\ldots,NF\} \\ v \in V(t)}}, \{h_j(b,t)\}_{\substack{j \in \{1,\ldots,NH\} \\ b \in B(t)}} \right\rangle,$$

gdzie:

$G(t)$ – graf [15] opisujący topologię sieci Internet (lub jej fragmentu będącego przedmiotem zainteresowania ze względu na cel modelowania) w chwili *t*, gdzie: $V(t)$ – zbiór wierzchołków (ang. *vertices*) grafu $G(t)$; $B(t)$ – zbiór gałęzi (ang. *branches*) grafu $G(t)$; $I(t)$ – relacja incydencji, $I(t) \subset V(t) \times B(t) \times V(t)$.

Wierzchołki (elementy aktywne sieci Internet) oraz gałęzie (łącza przewodowe i bezprzewodowe pomiędzy alemantami aktywnymi sieci Internet) opisane są zbiorem funkcji określających wartości ich atrybutów:

$f_i(v,t) : V(t) \times T \rightarrow X_i$ – *i*-ta funkcja opisana na wierzchołkach grafu $G(t)$;

$h_j(b,t) : B(t) \times T \rightarrow Y_j$ – *j*-ta funkcja opisana na gałęziach grafu $G(t)$;

NF – liczba funkcji opisanych na wierzchołkach *G(t)*;

NH – liczba funkcji opisanych na gałęziach *G(t)*.

Zbiory X_i i Y_j, czyli wartości funkcji $\{f_i(v,t)\}$ i $\{h_j(b,t)\}$, mogą być z różnych przestrzeni, w szczególności zależeć to będzie od przyjętego sposobu opisu funkcjonowania sieci Internet lub jej badanego fragmentu.

Opis formalny cyberzagrożeń musi uwzględniać cechy charakteryzujące każdy z możliwych typów cyberzagrożeń. Stąd wektor cyberzagrożeń można zdefiniować następująco:

$$CTs(t) = \left[CT(t,k)_{k \in K(t) = \{botnet, malware, \ldots\}} \right],$$

gdzie: $CT(t,k)$ – model *k*-tego typu cyberzagrożenia;

$\overline{\overline{K(t)}}$ – liczba typów zagrożeń, które wystąpiły lub mogą wystąpić.

Jako model cyberzagrożenia typu botnet proponuje się parę uporządkowaną:

$$CT(t, k = botnet) = \left\langle BN(t), Diff(t) \right\rangle,$$

gdzie: $BN(t)$ – sieć ewoluująca, opisująca topologię i charakterystyki ilościowe sieci botnet;

$Diff(t)$ – model ewolucji sieci botnet w sieci Internet.

Sieć botnet modelowana jest jako trójka uporządkowana:

$$BN(t) = \left\langle BG(t) = \left\langle BV(t), BE(t) \right\rangle, \{bf_i(bv,t)\}_{\substack{i\in\{1,\ldots,NBF\} \\ bv\in BV(t)}}, \{bh_j(be,t)\}_{\substack{j\in\{1,\ldots,NBH\} \\ be\in BE(t)}} \right\rangle$$

gdzie: $BG(t)$ – graf opisujący topologię sieci botnet w chwili *t*. Uwaga! Graf $BG(t)$ jest szkieletem podgrafu grafu $G(t)$, którego zbiorem wierzchołków są te wierzchołki $V(t)$, które są zainfekowanymi komputerami – *zombie*, a krawędzie odwzorowują kanały komunikacyjne pomiędzy *zombie*, powstałe w oparciu o gałęzie $B(t)$.

Wierzchołki oraz krawędzie grafu $BG(t)$ opisane są zbiorem funkcji określających wartości ich atrybutów:

$bf_i(bv,t): BV(t)\times T \rightarrow Z_i$ – *i*-ta funkcja opisana na wierzchołkach grafu $BG(t)$;

$bh_j(be,t): BE(t)\times T \rightarrow Q_j$ – *j*-ta funkcja opisana na krawędziach grafu $BG(t)$;

NBF – liczba funkcji opisanych na wierzchołkach *BG(t)*;

NBH – liczba funkcji opisanych na gałęziach *BG(t)*.

Zbiory Z_i i Q_j, czyli wartości funkcji $\{bf_i(bv,t)\}$ i $\{bh_j(be,t)\}$, mogą być z różnych przestrzeni. W szczególności rozważa się uwzględnienie takich atrybutów wierzchołków sieci botnet, jak: rola, stan i lokalizacja, oraz takich atrybutów krawędzi sieci botnet, jak: protokół komunikacji, częstotliwość komunikacji w określonym przedziale czasu, rozmiar przesłanego komunikatu.

Model ewolucji sieci botnet w sieci Internet zdefiniowano następująco [4, 12, 13]:

$$Diff(t) = \left\langle CNet(t), \{MDM\}_{l\in\{1,\ldots,NMDM\}}, Gen(v,t) \right\rangle,$$

gdzie:

$CNet(t)$ – model opisujący topologię i charakterystyki ilościowe sieci Internet (lub jej fragmentu będącego przedmiotem zainteresowania ze względu na cel modelowania), będący elementem składowym modelu *CyberSpace(t)*;

MDM_l – probabilistyczna maszyna stanowa opisująca zjawisko rozprzestrzeniania się złościlwego oprogramowania (ang. *Malware Diffusion Model*) odpowiedzialnego za ewolucję *l*-tego rodzaju botnetu, $l\in\{1,\ldots,NMDM\}$;

$Gen(v,t)$ – funkcja modelująca interakcje (przesyłanie komunikatów) pomiędzy wierzchołkami w sieci $CNet(t)$.

Przedstawiony wyżej szkielet modelu cyberprzestrzeni pozwala na ilościową analizę sieci botnet z wykorzystaniem charakterystyk i algorytmów z obszaru teorii grafu i sieci; w konsekwencji umożliwia opracowanie skutecznych metod wykrywania, przeciwdziałania i neutralizacji skutków cyberzagrożeń typu botnet.

6 Teoria sieci złożonych a topologia sieci botnet

Warto zwrócić uwagę, że dotychczasowe badania sieci botnet wykazują, iż mają one topologię tzw. sieci złożonych (ang. *Complex Networks*) [7]. W konsekwencji algorytmy, które zostały opracowane na potrzeby generacji sieci złożonych, mogą

zostać wykorzystane do badania własności sieci botnet. Spostrzeżenie to jest niezwykle cenne, gdyż pozwala na prowadzenie eksperymentów, które w innym przypadku byłyby niemożliwe lub trudne do przeprowadzenia ze względu na ograniczone możliwości pozyskania danych o aktywnych w danym momencie sieciach botnet.

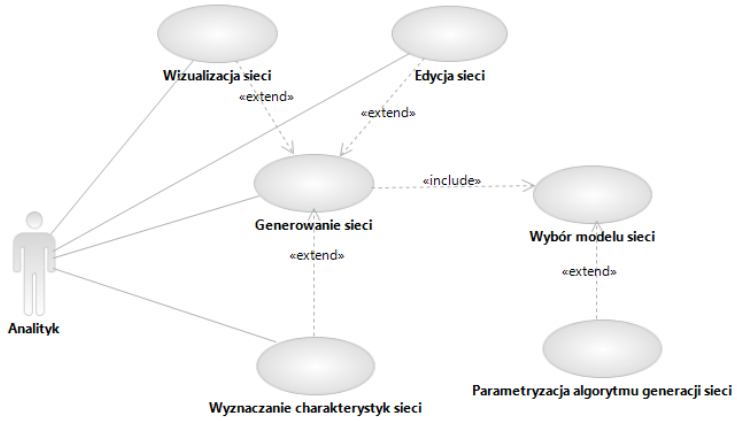
Powszechnie wykorzystwanymi algorytmami do generacji sieci złożonych są grafy losowe (ang. *Random Graphs*) [8, 9] oraz sieci bezskalowe (ang. *Scale Free*) [2, 3, 6]. W klasycznym już modelu *RG(n,p)* graf losowy jest generowany za pomocą procedury, która obejmuje dwa etapy. W pierwszym etapie ustala się liczbę *n* wierzchołków grafu, a następnie, w drugim etapie, każdą z C_n^2 par wierzchołków łączy się krawędzią z prawdopodobieństwem *p*. Powstała w ten sposób sieć ma jednorodną naturę wierzchołków, tj. nie istnieją wierzchołki wyróżnione (o wysokim stopniu w porównaniu z średnią wartością stopnia wierzchołka w grafie), które wpływają istotnie na funkcjonowanie sieci jako całości. W konsekwencji sieci o tej strukturze w praktyce są trudne do niszczenia, czyli rozspajania na wiele składowych spójności. Klasyczny model grafów losowych pozwala jedynie na generowanie sieci statycznych, co utrudnia analizę sieci ewoluujących, których przykładem są sieci botnet.

Model sieci *Scale Free* uwzględnia właśnie fakt, że sieci rzeczywiste nie są konstrukcjami statycznymi, lecz mają charakter ewolucyjny. Sieci rzeczywiste „rosną" przez dodawanie kolejnych węzłów, przy czym nowe węzły przyłączane są z większym prawdopodobieństwem do tych węzłów, które mają większy stopień. Tego typu zachowanie określane jest jako „dołączenia preferencyjne" (ang. *preferential attachment*), co polega na przyłączaniu węzłów do istniejącej sieci według określonej hierarchii. Modyfikacji podstawowego algorytmu generacji sieci *Scale Free* jest niezwykle dużo i ciągle powstają nowe, co świadczy oczywiście o olbrzymim zainteresowaniu tym obszarem badań nad sieciami złożonymi. Modyfikacje polegają głównie na zmianie liniowej reguły preferencyjnych dołączeń na różne (często bardzo skomplikowane) reguły nieliniowe. Innym pomysłem jest uwzględnianie w liniowej regule preferencyjnych dołączeń tzw. początkowej „atrakcyjności" węzłów lub efektu „starzenia się" węzłów i możliwości ich dezaktywacji (braku możliwości przyłączania się do nich nowych węzłów). Dodatkowo wprowadza się również modyfikacje samego algorytmu ewolucji sieci. I tak na przykład w kolejnych krokach ewolucji możemy mieć do czynienia nie tylko z dodaniem nowego węzła wraz z nowymi krawędziami, ale również z dodaniem jedynie samych krawędzi do już istniejących węzłów czy z przepięciem wybranych krawędzi. W przypadku adaptacji alagorytmu generacji sieci *Scale Free* do modelowania sieci botnet uwzględnia się liczne parametry (np. położenie geograficzne, średni czas „życia" *zombie*, zanim zostanie wykryty, czasowa dezaktywacja spowodowana na przykład wyłączeniem zainfekowanego urządzenia na noc). Badania nad sieciami *Scale Free* dowodzą, iż sieć taka jest odporna na ataki losowe. Inaczej jest z atakami celowanymi na tzw. huby, czyli wierzchołki o wysokim stopniu. Taki atak może istotnie wpłynąć na integralność sieci i jej funkcjonowanie jako całości.

7 Środowisko eksperymentalne do badania sieci botnet

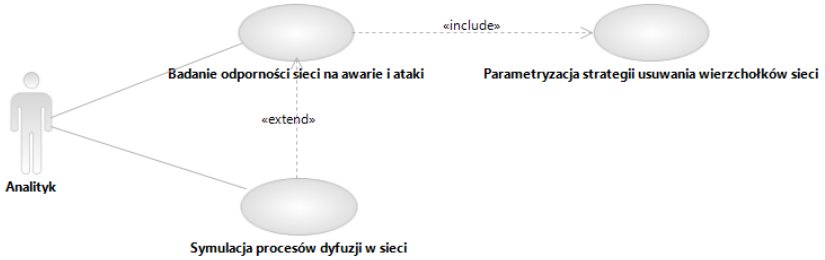
Platformą programową dla opracowanego środowiska eksperymentalnego jest Framework Gephi – interaktywna platforma do wizualizacji i eksploracji grafów i sieci udostępniana na licencji CDDL1.0 i GNU GPLv.3, o modularnej budowie,

zrealizowana w architekturze MVC (ang. *Model-View-Controller*) z wykorzystaniem wzorca IoC (ang. *Inversion of Control*). Gephi rozwija się poprzez dodawanie nowych rozszerzeń (ang. *plugins*) do istniejącego środowiska. Na uwagę zasługuje fakt, niejako wymuszenia przez twórców Gephi, implementacji rozszerzeń zgodnie z najlepszymi praktykami programowania obiektowego, które sprowadza się często do tzw. zasady SOLID (ang. *Single responsibility*, *Open-closed*, *Liskov substitution*, *Interface segregation*, *Dependency inversion*). Ciekawym zabiegiem jest również rozróżnienie między API (ang. *Application Programming Interface*) samego Frameworka Gephi a API oferowanego przez dołączane do niego rozszerzenia, tzw. SPI (ang. *Service Provider Interface*). API Gephi jest tworzone przez autorów platformy (lub pod ich nadzorem) i z założenia jest rzadko zmieniane. SPI jest natomiast zbiorem interfejsów do usług zaimplementowanych w postaci rozszerzeń i z tego powodu za ich poprawne działanie nie odpowiadają architekci Frameworka Gephi. Takie podejście jest ukłonem w kierunku potrzeby dzisiejszych czasów, związanej z koniecznością szybkiego wytwarzania oprogramowania w oparciu o istniejące komponenty. Z drugiej jednak strony przyjęte rozwiązanie zapewnia zachowanie wysokiego poziomu szeroko pojętej jakości oprogramowania, przy jednoczesnym zapewnieniu, że zbudowane rozszerzenia będą mogły być wykorzystywane przez liczną już społeczność użytkowników Gephi. Środowisko eksperymentalne zostało zbudowane jako zbiór autorskich rozszerzeń środowiska Gephi, a jego funkcjonalność przedstawiona została z wykorzystaniem diagramów przypadków użycia (ang. *Use Case Diagram*) [13].



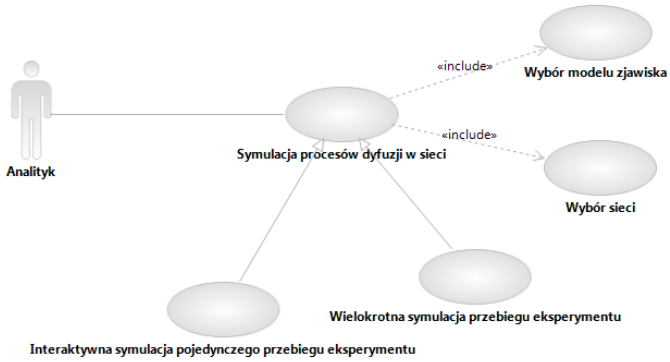
Rys. 2. Diagram przypadków użycia dla funkcjonalności „Generowanie sieci botnet”
Źródło: Opracowanie własne

Fig. 2. A use case diagram for functionality: ”Generating botnet network”
Source: Own elaboration



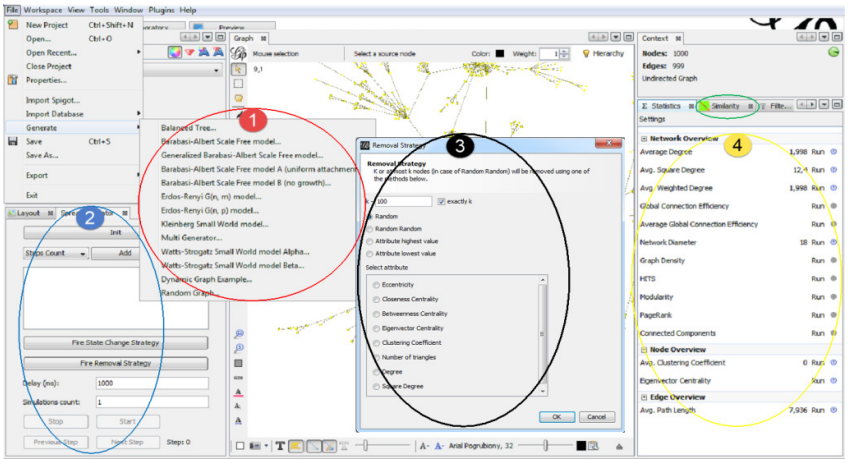
Rys. 3. Diagram przypadków użycia dla funkcjonalności „Badanie odporności sieci botnet na awarie i ataki”
Źródło: Opracowanie własne

Fig. 3. A use case diagram for functionality: ”Testing botnet network resilience to failures and attacks”
Source: Own elaboration



Rys. 4. Diagram przypadków użycia dla funkcjonalności „Symulacja ewolucji sieci botnet” zrealizowanej jako szczególny przypadek symulacji procesu dyfuzji w sieci
Źródło: Opracowanie własne

Fig. 4. A use case diagram for functionality: ”Simulation of botnet network evolution” - a special case of diffusion process simulation in the network
Source: Own elaboration



Rys. 5. Okno główne środowiska eksperymentalnego
Źródło: Opracowanie własne

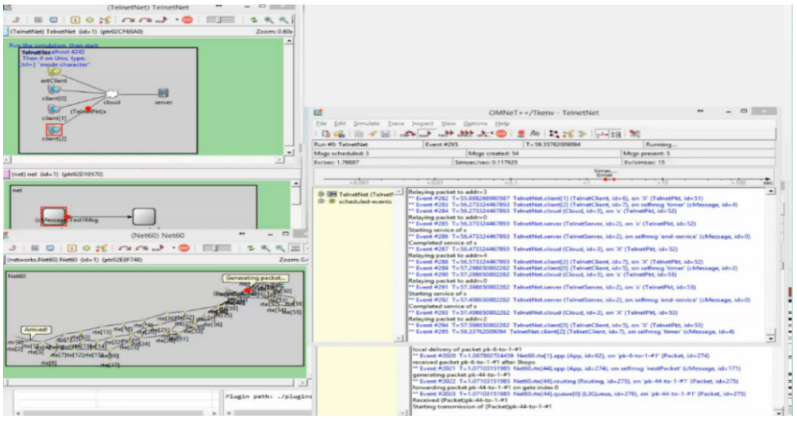
Fig. 5. A main window of experimental environment software
Source: Own elaboration

Rysunek 5 prezentuje główne okno środowiska eksperymentalnego z zaznaczonymi elementami interfejsu odpowiadającymi funkcjonalnością przedstawionym w postaci przypadków użycia. Kolorem czerwonym (obszar nr 1) zaznaczone jest podmenu z listą zaimplementowanych generatorów sieci. Zakładkę do parametryzacji symulatora procesów dyfuzji w sieci zaznaczono kolorem niebieskim (obszar nr 2). Kolorem czarnym (obszar nr 3) zaznaczono okno pozwalające na badanie odporności sieci na awarie i ataki. Warto zwrócić również uwagę na zakładkę prezentującą statystyki grafów (zaznaczoną kolorem żółtym – obszar nr 4) dostępne na platformie *Gephi*. Algorytmy te stanowią integralną część platformy i są sukcesywnie dopisywane i udoskonalane przez społeczność programistów *Gephi*, w tym przez autorów pracy. Reasumując, środowisko zbudowane w oparciu o *Gephi* umożliwia:

- analizę charakterystyk zidentyfikowanej/hipotetycznej sieci botnet;
- ocenę odporności sieci botnet na zdarzenia przypadkowe i świadome działania mające na celu zwalczanie/przejęcie sieci botnet;
- opis i analizę, w tym symulację, procesu rozprzestrzeniania się złośliwego oprogramowania i ewolucję sieci botnet.

Dopełnieniem zaprezentowanego środowiska eksperymentalnego jest środowisko symulacji sieci teleinformatycznych wykorzystywane do modelowania i analizy, w tym symulacji, skutków wybranych ataków na rzeczywisty/hipotetyczny cel w sieci Internet. Przy wyborze środowiska symulacyjnego kluczowym kryterium była jego skalowalność i rozszerzalność. Ponadto symulator powinien umożliwić badanie skuteczności metod/mechanizmów ochrony różnych metod/mechanizmów ataków w oparciu o zamodelowaną infrastrukturę. Istnieje wiele środowisk pozwalających na modelowanie rzeczywistych sieci teleinformatycznych (np. OMNeT++, CNet, NS-2, PRIME SSF, Möbius i inne). W środowiskach naukowych, w zakresie analizy skutków

ataków na infrastrukturę teleinformatyczną, często wykorzystywany jest pakiet OMNeT++. Ze względu na istnienie w tym narzędziu wszystkich protokołów i warstw ISO OSI można wiernie odwzorować cel ataku, jak i sam atak. Pakiet OMNeT++ jest dostępny publicznie na licencji APL, posiada budowę modułową, silnik symulacji z dyskretnym modelem zdarzeniowym oraz otwartą architekturę (implementacja w C++). Dużym ułatwieniem są rozbudowane narzędzia programisty i dobra dokumentacja od projektu, przez implementację po uruchomienie i zbieranie wyników. Programista ma możliwość skorzystania z bogatego zbioru gotowych bibliotek.



Rys. 6. Okno główne środowiska OMNeT++
Źródło: Opracowanie własne

Fig. 6. A main window of OMNeT++
Source: Own elaboration

8 Podsumowanie

Zagrożenia wynikające z działalności sieci botnet są niezwykle istotne w kontekście utrzymania bezpieczeństwa w cyberprzestrzeni. Dodatkowo duża dynamika zmian w sposobie funkcjonowania botnetów oraz metodach ataku, których są źródłem, zwiększa potrzebę ich szczegółowej analizy. Nie wystarcza obserwacja i analiza ruchu sieciowego oraz reakcje na występujące anomalie. Rośnie potrzeba posiadania umiejętności przewidywania ataków oraz uruchamiania metod ochrony dostosowujących się do dużej dynamiki zmian w formach aktaków. Zbudowany szkielet modelu matematycznego cyberprzestrzeni, uwzględniający fakt występujących w niej zagrożeń, stanowił podstawę do budowy środowiska eksperymentalnego i koncepcji wykorzystania środowiska symulacyjnego OMNeT++. Kolejnym krokiem będzie rozszerzenie modelu matematycznego oraz zbudowanych narzędzi programowych, które stanowią swoiste laboratorium badawcze do analizy sieci botet oraz opracowania metod ich skutecznego zwalczania.

Literatura

1. Antkiewicz R., Dyk M., Kasprzyk R., Najgebauer A., Pierzchała D., Tarapata Z., Maj M.: Koncepcja rozwoju zdolności w obszarze cyberbezpieczeństwa infrastruktur krytycznych państwa, w raporcie Instytutu Kościuszki na temat

Bezpieczeństwo infrastruktury krytycznej wymiar teleinformatyczny, ISBN 978-83-63712-15-0, str. 93-102, Warszawa 2014

2. Barabási A.L., Albert R.: Emergency of Scaling in Random Networks, *Science*, 286, 509-512, 1999
3. Barabási A.L., Albert R.: Topology of Evolving Networks: Local Events and Universality, *Physical Review Letters*, Vol. 85, Nr 24, 5234-5237, 2000
4. Bartosiak C., Kasprzyk R., Najgebauer A.: The graph and network theory as a tool to model and simulate the dynamics of infectious diseases, *Bio-Algorithms and Med-Systems*, Vol. 9, Issue 1, 17-28, 2013
5. CERT.GOV.PL, *Raport o stanie bezpieczeństwa cyberprzestrzeni RP w roku 2014*, 2015
6. Chen Q., Chang H., Govindan R., Jamin S., Shenker S.J., Willinger W.: The origin of power laws in Internet topologies revisited, *Proceedings of the 21st Annual Join Conference of IEEE Computer and Communication Societies* 2002, IEEE Computer Society, 2002
7. Dagon D., Gu G., Zou C., Grizzard J., Dwivedi S., Lee W., Lipton R.: A Taxonomy of botnet structures – referat wygłoszony: *Computer Security Applications Conference*, 2007. ACSAC 2007
8. Erdös P., Rényi A.: On random graphs, *Publicationes Mathematicae* 6, 290-297, 1959
9. Erdös P., Rényi A.: On the evolution of random graphs, *Publications of the Mathematical Institute of the Hungarian Academy of Sciences* 5, 17-61, 1959
10. Godkin T.: *Statistical Assessment of Peer-to-Peer Botnet Features* (Master of Applied Science), University of Victoria, 2013
11. Grzelak M., Liedel K.: Bezpieczeństwo w cyberprzestrzeni. Zagrożenia i wyzwania dla Polski - zarys problem, *Bezpieczeństwo Narodowe* nr 22, II – 2012
12. Kasprzyk R.: Diffusion in Networks, *Journal of Telecommunications and Information Technology*, 2/2012, 99-106, 2012
13. Kasprzyk R.: *Modele ewolucji systemów złożonych i metody badania ich charakterystyk dla potrzeb komputerowej identyfikacji potencjalnych sytuacji kryzysowych*, praca doktorska, promotor A. Najgebauer, Wydział Cybernetyki Wojskowa Akademia Techniczna, Warszawa 2012
14. Kijewski A.: *Secure 2013 CERT Polska vs botnety*, 2013
15. Korzan G.: *Elementy Teorii grafów i sieci – metody i zastosowania*, WNT, Warszawa 1978
16. Namiestnikow J.: *Ekonomia botnetu*, Kaspersky Lab, 2009
17. Tarapata Z., Kasprzyk R.: Graph-based optimization method for information diffusion and attack durability in networks, *Lecture Notes in Computer Science*, 2010, Volume 6086/2010, p. 698-709, Springer Berlin/Heidelberg
18. Kamluk V.: *Biznes botnetowy*, Kaspersky Lab

Streszczenie

W pracy przedstawiono analizę cyberzagrożeń ze szczególnym naciskem położnym na cyberzagrożenia wynikające z aktywności sieci typu **botnet.** Sieci te są najbardziej powszechne i często postrzegane jako wyjątkowo istotne z punktu widzenia bazpieczeństwa państwa. Ich klasyfikacja oraz metody rozprzestrzeniania się są

podstawą do budowy szkieletu modelu cyberprzestrzeni uwzględniającego występowanie w niej cyberzagrożeń (w tym sieci typu botnet). Opracowany model jest podstawą budowy środowiska eksperymentalnego umożliwiającego analizę charakterystyk sieci botnet, badanie jej odporności na różne zdarzenia, symulację jej rozprzestrzeniania się oraz ewolucji. Zaproponowane zostały do tego celu platformy, których możliwości i cechy funkcjonalne są w stanie sprostać tym wymaganiom.

Słowa kluczowe: cyberzagrożenia, sieci botnet, sieci złożone

Modelling and simulation of Botnet-based cyber threats

Summary

The paper presents an analysis of cyberthreats, with particular emphasis on the threats resulting from **botnet** activity. Botnets are the most common types of threats and often perceived as crucial in terms of national security. Their classification and methods of spreading are the basis for creating cyberspace model including the presence of cyberthreats (including botnets). A well-designed cyberspace model enables to construct an experimental environment that allows for the analysis of botnet characteristics, testing its resistance to various events and simulation of the spread and evolution. For this purpose, dedicated platforms with capabilities and functional characteristics to meet these requirements have been proposed.

Keywords: cyberthreat, botnet networks, complex networks