СИСТЕМА ЗА РАЗПОЗНАВАНЕ НА БОТНЕТ АТАКИ, БАЗИРАНА НА АНОМАЛИИ

Юлия Александрова Алексиева[1], Христо Георгиев Вълчанов[2]
[1]*Технически Университет – Варна, e-mail: iulia93@abv.bg*
[2]*Технически Университет – Варна, e-mail: hristo@tu-varna.bg*

ANOMALIES BASED BOTNET DETECTION SYSTEM

Yulia Aleksandrova Aleksieva[1], Hristo Georgiev Valchanov[2]
[1] *Technical University of Varna, e-mail: iulia93@abv.bg*
[2] *Technical University of Varna, e-mail: hristo@tu-varna.bg*

Abstract
One of the tools for implementing of Denial of Service attacks is Botnet. Detecting such attacks is important for the security of computer networks. The intrusion detection systems based on anomalies detect Botnets, monitoring a number of network traffic anomalies such as high latency, high traffic volumes, traffic over unusual ports, etc., which could mean malicious bots in the network. This paper presents some aspects of the implementation of a host-based Botnet attack detection system. The system uses a technique to detect behavior anomalies, based on variation of genetic algorithm.The experimental results are shown.

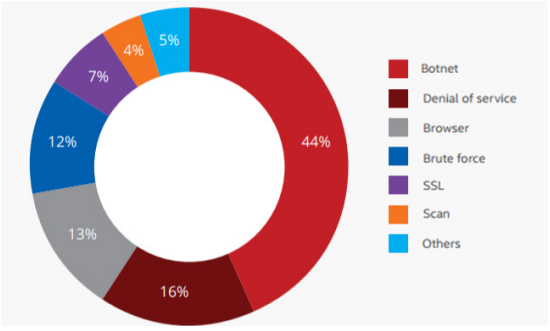
Keywords: Network Botnet Attacks, DoS Attacks, Intrusion Detecting System, Genetic Algorithms.

ВЪВЕДЕНИЕ

Атаките с отказ на услуга (Denial of service DoS) са едни от най разпространените в последните години в Интернет [1]. Те целят да доведат атакуваната система до невъзможност да предоставя услуга. Атаката с отказ на услуга е различна по цел, форма и ефективност спрямо повечето атаки срещани в мрежата и компютрите. Реализациите на подобен тип атаки биват осъществени посредством лавинообразно изпращане на голям брой пакети (flood) от един или много източници едновременно. Целта е да бъдат претоварени крайните устройства за които са предназначени пакетите или да предизвикат насищане на преносните канали към атакуваната система. Най-често този тип атаки се реализират с помощта на подмяна (spoofing) на адресите на източните на атаките. Поради разнообразният си характер, този тип атаки трудно могат да бъдат параметризирани, съответно разпознаването и предприемането на действие за защита срещу тях не е лесна задача.

Най-използваният начин за реализиране на DoS атаки е Ботнет [2]. Ботнет представлява група от ботове (съвкупност от заразени компютри), изпълняващи зловреден софтуер, които се контролират от един хакер – общоизвестен като *главен бот* (botmaster), чрез централизирана инфраструктура (C&C) като център на атаката. Според McAfee Labs [3] за последното тримесечие на 2017г. най-голям дял от мрежовите атаки в световен мащаб има Ботнет (фиг. 1).

Една от главните цели за съвременната мрежова сигурност е да бъдат създадени адекватни техники за откриването и евентуалното прекратяване на Ботнет заплахите. В настоящия доклад е представен подход за реализиране на хост-базирана система за откриване на Ботнет атаки. Системата открива аномалии в поведението на мрежата на базата на вариация на генетичен алгоритъм.



***Фиг. 1**. Дял на мрежовите атаки за 2017г.*

РАЗПОЗНАВАНЕ НА БОТНЕТ АТАКИ

Към момента съществуват разнородни средства за разпознаване на проникване в системата [4]. Съществуващите системи използват основно откриване базирано на аномалии или правила. При засичането на аномалиите основната част от системата е да се профилира нормалното поведение на системата. Когато нормалното поведение е установено, системата може да бъде използвана за откриване на аномалии на база профила изграден за нормалното поведение. Профилите са разработени чрез наблюдение на характеристиките на типична активност през определен период от време. Основен проблем при генерирането на профили е, че може да е доста трудно в някои случаи те да бъдат направени точни, поради сложността на компютърната дейност. Предимството е възможността за разпознаване на нови типове атаки.

Известни са редица средства за разпознаване на атаки, базирани на този подход.

NIDES/STAT (Next-generation real-time Intrusion Detection Expert System Statistical component) е експертна система за откриване на атаки в реално време със статистическа компонента [5]. Системата използва статистически модели за описване на нормалното поведение на системата. Извършва се наблюдение над хоста и ако се открие поведение, което значително се различава от определеното като нормално, то се отбелязва като евентуално наличие на атака. Създава се профил за всеки обект, който отразява поведението на потребител. Основният проблем с тази система е, че въвежда възможността на атакуващия постепенно да тренира профила, за да може системата постепенно да приема тези действия като нормално поведение.

Аналогична система е Haystack [6]. Тя също използва статистически модел за профилиране на поведението на системата. За разлика обаче от NIDES/STAT, нейният алгоритъм има етап, при който се определят прилики спрямо познати атаки. Предимството е, че има информация за повече атаки и съответно по-бързо и по-ефикасно те могат да бъдат откривани. Недостатък е сложността на алгоритъма на оценка на поведението, системата е значително по-бавна в анализа и алармирането при наличие на атака.

Системата Ossec е безплатна и е с отворен код, като предоставя графичен интерфейс за управление [7]. Има мощен корелационен и аналитичен компонент, извършва анализи на журнали, проверява интегритета на файловете и наблюдава регистрите. Предимство е работата в реално време. Недостатък е малкото знания за познати атаки, водещо до висок коефициент на грешно алармиране за открито проникване.

Алтернативно средство е Tripwire [8]. Това е също система с отворен код която проверява интегритета на данните. Системата наблюдава определени файлове и дава сигнал, когато открие някаква промяна. Тъй като се извършва постоянен мониторинг, се изискват значителни ресурси на машината. Друг недостатък е, че алармирането за открито нарушение се извършва чак когато в системата вече има наличие на повредени файлове.

Системата Nessus представлява комерсиален скенер за уязвимости [9]. Основната й функционалност е откриване на опит за отдалечен неоторизиран достъп до машината. Предимството на това средство е извършването на мониторинг над мрежовия трафик и анализиране на пакетите, давайки възможност за откриване на DoS атаки срещу TCP/IP стека. Недостатък е, че продуктът не може да се използва за персонална защита.

ПРЕДЛАГАН ПОДХОД

Предлаганият в доклада подход за анализ на обработваните пакети използва специфична вариация на генетичен алгоритъм [10,11]. Тази вариация се базира на генетичния оператор селекция, като се оценяват

всички индивиди в поредните поколения на база на аналитично определена фитнес функция. Предлаганият подход е реализиран в хост-базирана система за разпознаване на Ботнет атака.

Всеки жив организъм има собствен индивидуален план, закодиран в гените си. Тези гени са свързани помежду си и сформират хромозоми, а хромозомите сформират организъм, наречен фенотип. В случая на използвания генетичен алгоритъм, като фенотип се третират получените пакети. На тяхна основа се формират хромозоми, изследват се и така се засича промени във фенотипа и мутация. Алгоритъмът извлича от получения пакет (фенотипа) неговите гени и комбинира тези гени във верификационна хромозома. Когато се извършва засичане на spoofing атака, алгоритъмът изследва хедърите на всеки пакет и извлича следните атрибути, наричани по-нататък гени:

- Source IP
- Source Mac Address
- DestinationMAC Address
- Initial Time to Live
- Hop Count
- Protocol Number
- Packet ID

Всички тези гени представляват променливи, които могат да бивaт променени, в случай, че нарушител се опитва да приложи spoofing атака. Алгоритъмът преобразува стойностите на тези гени в техния двоичен еквивалент и ги конкатенира, за да се получи хромозомата на пакета. На фиг.2 е показано формирането на примерна хромозома.

Трябва да се отбележи, че последният бит на хромозома представлява валидността на идентификатора на пакета, а не самият идентификатор. Ако идентификаторът на пакета е по-голям от този на предишния пакет, изпратен от същия източник, то валидността се приема за 1, в противен случай е 0. Идентификаторът на пакета е нарастваща стойност и следователно, последователно изпращани валидни пакети от един и същ източник, трябва да имат последователни нарастващи идентификатори. В противен случай се приема, че има външна намеса, осъществена при изпращането на пакета, превръщайки го в евентуален опит за атака.

Source MAC - 02:00:4c:4f:4f:50 : (10)(0)(1001100)(1001111)(1001111)(1010000)
Destination Mac - 01:00:5e:7f:ff:fa : (1)(0)(1011110)(1111111)(11111111)(11111010)
TTL - 31 : (11111)
Hop Count - 1 : (1)
Packet ID - 288 : (1)

1001001100100111110011110100001010111101111111111111111111010111111

***Фиг. 2**. Формиране на хромозома от пакет*

Фитнес функцията на алгоритъма представлява аналитично изчисление между две хромозоми, получени от един и същ адрес. Двете хромозоми биват двоично сравнени бит по бит, изчислявайки се математически фитнес нивото на получения пакет. Когато системата получи пакет за първи път от даден IP адрес, гените му формират хромозома и тя се запазва със 100% фитнес ниво (ниво на годност). Тази хромозома се използва като шаблон за последваща верификация. Хромозомите на следващите пакети, получени от същия източник, се сравняват с шаблона, прилагайки фитнес функцията и аналогично се изчислява фитнес нивото на пакет. Ако фитнес нивото на новия приет пакет е по-малко от това на пакета, получен по-рано, то се счита, че е налице външна намеса.

Минималното допустимо фитнес ниво, което е настроено в системата, е 65% хромозомен фитнес. Този процент е получен след анализ и наблюдение на поведението на системата в нормална среда и в атакувана среда, като числото е избрано с цел по-ниски фалшиви резултати. Предвидено е потребителят да може да настрои нивото така, че да отговаря на конкретната мрежова среда.

След като хромозомата е генерирана, трябва да се изчисли нейното фитнес ниво. Нека, например, хромозомата на получен пакет от източник за първи път, която е генерирана от алгоритъма, е 1010101010. След това, ако се получи друг пакет от източник със същия IP адрес, се очаква неговата хромозома да е почти същата, ако ли не – то поне близка до предишната. Това се дължи на факта, че MAC адресите на източника и дестинацията трябва да са еднакви, а TTL и HopCount - приблизително еднакви. Също така, идентификаторът на пакета се очаква да е по-голям от идентификатора на предишния пакет. Ако всичко това е нали-

це, очаква се вторият получен пакет да има почти същата хромозома. Например, нека хромозомата на втория получен пакет е 1100101001.

	1 0 1 0 1 0 1 0 1 0
	1 1 0 0 1 0 1 0 0 1
Резултат:	C H H C C C C C H H

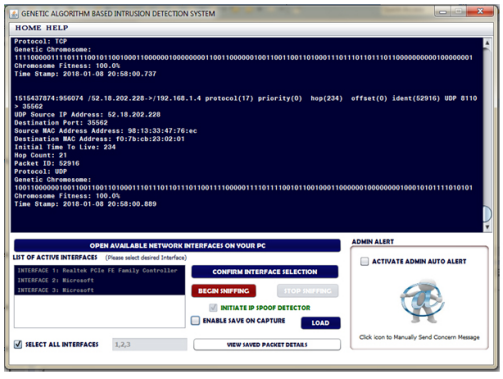
***Фиг. 3**. Сравнение на две хромозоми на последователни пакети*

Може да се направи следното сравнение на две хромозоми на последователни пакети от един и същ източник, показано на фиг. 3 (обозначенията са: *C = Съвпадение, H = Няма съвпадение*). От сравнението се виждат 4 разлики в хромозомите. В този случай вторият пакет притежава фитнес ниво 60%, което означава, че е бил значително променен и системата ще генерира аларма.

ЕКСПЕРИМЕНТАЛНИ ИЗСЛЕДВАНИЯ И РЕЗУЛТАТИ

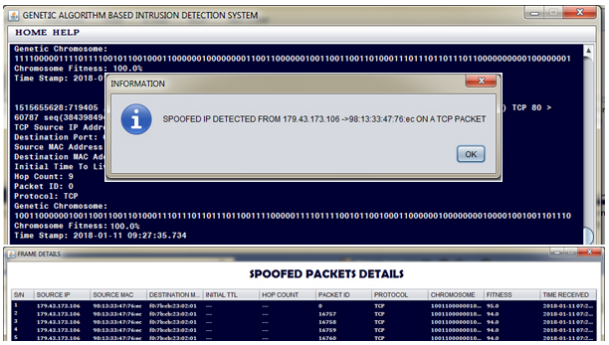
Тестването на представената система е извършено в локална мрежа в която са атакуващата машина и хост-базираната система. В нормална среда хост-базираната система получава пакети с фитнес ниво в диапазона 95-100%, породено основно от промените в идентификаторите на пакета (фиг. 4).

При тестовете за детектиране на Ботнет атака, ботът изпълнява TCP SYN Flood атака, като изпраща пакети, докато не бъде спрян. Функционалността му е да прикрие IP адреса на машината, на която е стартиран, като при стартирането му се задава друг адрес на източника. За конкретния тест е избран IP адрес 222.222.222.222. Като команда се подава и адреса на атакуваната машина – тази, на която е стартирана хост-базираната система за откриване на атаки (адрес 192.168.1.3).



***Фиг. 4**. Информация за преминаващия трафик*

От машината, на която се изпълнява бота, първо са изпратени валидни пакети, след това е стартирана атаката. Системата получава валидните пакети, съхранява ги в базата данни и когато атаката се стартира, при получаване на пакет се формира хромозом. В базата данни се търсят пакети от същия MAC адрес. При откриване на такива се сравняват двете хромозоми, но тъй като IP адресът е бил подменен, се получава голяма разлика, която надвишава фитнес нивото (в случая 65%). В резултат се генерира аларма (фиг.5), като може да се получи детайлна информация за прехванатите пакети.



***Фиг. 5**. Детектиране на Ботнет атака*

Резултатите от тестването показват, че алгоритъмът функционира коректно, но конкретната мрежа трябва да бъде внимателно анализирана, за да се избере подходящо фитнес ниво. В атакувана среда хост-базираната система открива веднага атаката и сигнализира адекватно.

Спрямо резултатите може да се заключи, че ниво под 65% не е допустимо да се допуска, както и ниво над 85%, тъй като това води до фалшиви позитивни резултати за откриване на атака.

ЗАКЛЮЧЕНИЕ

В настоящия доклад е представен подход за реализиране на хост-базирана система за откриване на Ботнет атаки. Системата използва техника за откриване на аномалии на базата на вариация на генетичен алгоритъм, като се анализира трафика, преминаващ през мрежовите интерфейси на хоста. Алгоритъмът анализира всеки получен пакет индивидуално, като определя дали е имало външна намеса, дължаща се на про-

ведена spoofing атака. Експерименталните резултати показват, че предлаганият подход има реална приложимост.

Цел на бъдеща работа е разширение на функционалността на системата, предоставяща още възможни техники за откриване на аномалии, като добавяне на анализ на интегритета на данните. Предвижда се и интегриране на сигнатурно-базирана техника, която да осигури бързо и ефикасно откриване на вече познати атаки.

ЛИТЕРАТУРА

[1] Shankdhar P. DoS Attacks and free DoS attacking tools, INFOSEC Institute, 2013.

[2] Elisan C. Malware, Rootkits& Botnets A beginner's guide, McGraw-Hill Education – Europe, ISBN 9780071792066, 2012, pp.55-82.

[3] McAfee Labs. https://www.mcafee.com (01.09.2018).

[4] Chouhan P., V. Richhariya. Anomaly Detection in Network using Genetic Algorithm and Support Vector Machine, International Journal of Computer Science and Information Technologies, Vol. 6 (5) , 2015, pp. 4429-4433.

[5] NIDES. http://www.csl.sri.com/projects/nides/ (01.09.2018).

[6] Al-Sakib K. The State of the Art in Intrusion Prevention and Detection. Auerbach Publications, 2014. ISBN 9781482203523

[7] OSSEC. https://www.ossec.net/ (01.09.2018).

[8] TripWire. https://www.tripwire.com (01.09.2018).

[9] Nessus Professional. https://www.tenable.com.

[10] Hashemi V., Z. Muda, W. Yassin. Improving Intrusion Detection Using Genetic Algorithm, ANSI - Information Technology Journal 12(11), 2013, pp. 2167-2173.

[11] Maniya P., V. Musande. Rules Based Intrusion Detection System Using Genetic Algorithm, International Journal of Computer Science and Network, Volume 5, Issue 3, June 2016, pp.554-55.