ОБНАРУЖЕНИЕ DNS-ТУННЕЛЕЙ С ПОМОЩЬЮ FEEDFORWARD НЕЙРОННОЙ СЕТИ

Белорусский государственный университет информатики и радиоэлектроники
г. Минск, Республика Беларусь.

Бубнов Я.В.
Иванов Н.Н. - канд. физ-мат. наук., доцент

Методика DNS-туннелирования эксплуатируется при разработке вредоносных программ с целью сокрытия их сетевой активности. Современные механизмы обеспечения защиты сетевой инфраструктуры не предоставляют необходимых инструментов для решения данной проблемы. В статье предлагается способ обнаружения DNS-туннелирования с помощью feedforward нейронной сети.

DNS-туннель (далее туннель) представляет собой способ предоставления сетевых сервисов с помощью инкапсуляции протоколов различных уровней модели OSI. Гибкость протокола DNS позволяет использовать разнообразные техники для создания туннелей. Самый распространенный способ подразумевает добавление передаваемой информации через субдомены делегированных зон [1].

В течение предыдущих лет создано большое количество вредоносных программ, таких как Morto [2], Feederbot [3], а также программ для кражи информации из платежных систем, как BernhardPOS и FrameworkPOS [4]. Упомянутые программы используют технику DNS-туннелирования для передачи данных из скомпрометированной системы.

В статье [5] Надлер, Аминов, Шабтай описывают два способа обнаружения туннелей: с помощью Isolation Forest, а также с помощью One-Class SVM. Исследование содержит позитивные результаты использования описанных техник, в частности авторы добились 99-процентной чувствительности разработанной модели. Однако авторы работы решают задачу бинарной классификации, то есть выявляют вредоносный трафик среди регулярного DNS трафика.

В данной статье описывается способ классификации сетевого трафика для обнаружения туннелей, а также программ, с помощью которых созданы туннели. Таким образом задача - классифицировать DNS трафик по типам программ для создания туннелей.

Исходные данные для решения задачи собраны путем создания туннелей с помощью наиболее распространенных программ, находящихся в открытом доступе: dns2tcp, dnscapy, iodine а также tuns. Исходя из этого можно выделить 6 классов: 4 вида туннелирования, регулярный трафик и весь остальной трафик.

Экспериментальная установка состоит из клиентской и серверной станций с установленным между ними туннелем. На серверной станции запущен SSH сервер, а клиентская станция копирует случайным образом сгенерированный файл размером 2MiB по инкапсулированному протоколу SSH. В таблице 1 указано количество выборок (DNS пакетов) для каждого вида программы. Помимо вредоносного трафика, на установке собран регулярный трафик путем разрешения списка доменных имен, представленных в репозитории OpenDNS [6].

Таблица 1. - Исходные данные задачи

Тип трафика	Размер выборки
Dns2tcp туннель	9070
Dnscapy туннель	14424
Iodine туннель	12228
Tuns туннель	12543
Регулярный трафик	17280

Для каждой секции DNS пакета из полученной выборки выделены атрибуты, указанные в таблице 2. Атрибуты состоят из атрибутов ресурсных записей (RR) каждой из секций: запроса, ответа, дополнительного ответа, а также из авторитативной секции.

Таблица 2. - Выбранные атрибуты DNS пакетов

Атрибут	Описание
RR count	Количество пакетов в секции сообщения.
RR name length	Длина имени ресурсной записи.
RR name shannon entropy	Информационная энтропия Шеннона имени ресурсной записи.
RR type	Тип ресурсной записи.
RR data length	Длина значения ресурсной записи.
RR data shannon entopy	Информационная энтропия Шеннона значения ресурсной записи.

Полученная совокупность атрибутов представлена на рисунке 1а с помощью пространственного уплотнения t-SNE [7].

Для решения задачи используется трехслойная нейронная сеть со следующей конфигурацией:

$$f(X) = ReLU(z^{(1)}) \circ D_{0.1}(z^{(2)}) \circ SoftMax(z^{(3)}),$$

где *ReLU* - функция активации первого слоя нейронной сети $z^{(1)}$, состоящая из 128 нейронов; $D_{0.1}$ - *dropout* слой для предотвращения переобучения сети [8] $z^{(1)}$;*SoftMax* - функция активации выходного слоя сети $z^{(3)}$количеством 128 нейронов. В качестве оптимизатора исследуемой функции классификации используется функция Нестерова-Адама [9], а в качестве функции потерь - перекрестная энтропия.

Обучение нейронной сети осуществляется поэтапно выборками фиксированного размера, где каждая

выборка случайным образом переупорядочивается с целью минимизации переобучения сети.

Результаты классификации с помощью описанной нейронной сети представлены в таблице 3. Как видно, точность распознавания трафика описанной модели составляет 83%, значение чувствительности составляет 81%, а специфичности - 84%.

Таблица 3. - Оценки качества классификации

Метрика	Значение
Accuracy	0,835156
Recall	0,818359
Precision	0,841027

Для оценки качества классификации построены ROC-кривые, представленные на рисунке 1б, для каждого из рассматриваемого класса, а также усредненное значение по всей совокупности классов. Собранные оценки характеризует высокую степень распознавания различных механизмов DNS туннелирования предложенной методикой.

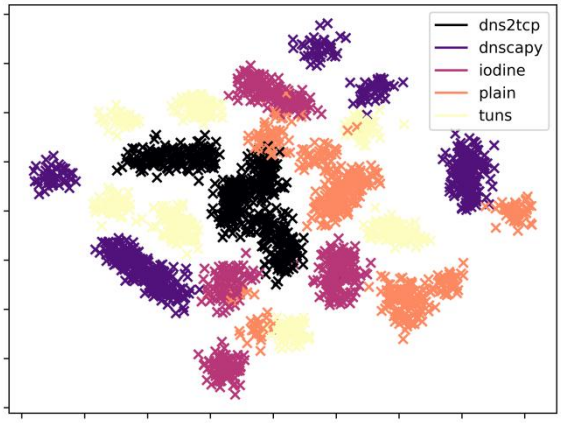

Рис. 1а - Исходные данные в двухмерном t-SNE представлении.

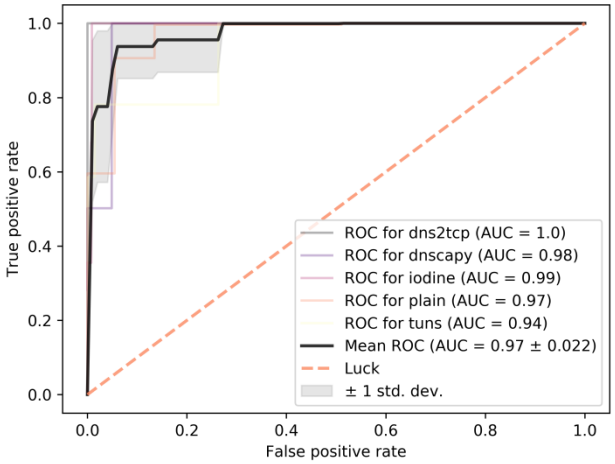

Рис. 1б – ROC-кривые для каждого класса, а также их среднее значение.

Большую точность в распознавании программы туннелирования dns2tcp можно объяснить тем, что это единственная программа, которая использует исключительно ресурсные записи типа TXT. Причем, в отличие от туннеля dnscapy, который периодически меняет тип записей с TXT на CNAME, программа dns2tcp не полностью утилизирует доступное пространство пакета: его длина в среднем меньше длины пакетов других программ.

В работе представлены результаты использования feedforward нейронной сети для решения задачи классификации DNS трафика с целью выявления туннелирования. Результаты подтверждают эффективность использования предложенной методики, дальнейшая работа будет направлена на улучшение качества распознавания, а также на возможность распознавания новых методик туннелирования.

Список использованных источников:

1. Farham, G. Detecting DNS Tunneling / G. Farham, A. Atlasis. - Boston : SANS Institute, 2013.

2. Zhong, X. Stealthy Malware Traffic - Not as Innocent as It Looks / X. Zhong, Y. Fu, L. Yu, R. Brooks, K. Venayagamoorthy - Clemson : Clemson University Press, 2017.

3. Deitrich, C. On Botnets that use DNS for Command and Control / C. Deitrich, C. Rossow, F. Freiling, H. Bos, M. van Steen, N. Pohlman - Gelsenkirchen : Institute for Internet Security, 2011.

4. Valenzuela, I. Game Changer: Identifying and Defending Against Data Exfiltration Attempts / I. Valenzuela. - Boston : Sans Institute, 2015.

5. Nadler, A. Detection of Malicious and Low Throughput Data Exfiltration Over the DNS protocol / A. Nadler, A. Aminov, A. Shabtai. - Negev : Ben Gurion University of the Negev, 2017.

6. OpenDNS Top Domains List [Электронный ресурс]. - Режим доступа: http://github.com/opendns/public-domain-lists. - Дата доступа: 03.04.2018.

7. Van der Maaten, L. Visualizing Data using t-SNE / L. van der Maaten, G. Hinton. - Camgridge : Journal of Machine Learning Research, 2008.

8. Srivastana, N. Dropout: A Simple Way to Prevent Neural Networks from Overfitting / N. Srivastana, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov. - Cambridge : Journal of Machine Learning Reasearch, 2014.

9. Dozat, T. Incorporating Nesterov Momentum into Adam / T. Dozat. - Stanford University Press, 2017.