прикладные науки

MDPI

Статья

Метод классификации доменных имен DGA на основе долговременной памяти с механизмом внимания

Янчен Цяо [1], Бинь Чжан [1,*], Вэйчжэ Чжан [1,2], Арун Кумар Сангайя [3] и Хуалун Ву [1]

[1] Исследовательский центр безопасности киберпространства, лаборатория Пэн Чэн, Шэньчжэнь 518000, Китай;qiaoych@pcl.ac.cn (Y.Q.);weizhe.zhang@pcl.ac.cn (W.Z.);hualong.wu@pcl.ac.cn (H.W.)

[2] Школа компьютерных наук и технологий, Харбинский технологический институт, Харбин, 150000, Китай

[3] Школа компьютерных наук и инженерии, Университет VIT, Веллор, 632014, Индияsarunkumar@vit.ac.in

* Корреспонденцияbin.zhang@pcl.ac.cn

Получено: 15 сентября 2019 г.; Принято: 3 октября 2019 г.; Опубликовано: 9 октября 2019 г.

проверка ротора **обновления**

Аннотация: В настоящее время многие кибератаки используют алгоритм генерации доменов (DGA) для генерации случайных доменных имен, чтобы поддерживать связь с сервером связи и управления (C&C). Заблаговременное обнаружение доменных имен DGA может помочь обнаружить атаки и вовремя отреагировать на них. Однако в последние годы был принят и введен в действие Общий регламент по защите данных (GDPR), и метод классификации DGA на основе контекстной информации, такой как WHOIS (информация о зарегистрированных пользователях или правопреемниках доменного имени), больше не применяется. В то же время получение алгоритма DGA путем реверсирования образцов вредоносных программ сталкивается с проблемой отсутствия образцов вредоносных программ по различным причинам, например, безфайловых вредоносных программ. Мы предлагаем метод классификации доменных имен DGA, основанный на использовании долговременной памяти (LSTM) с механизмом внимания. Этот метод ориентирован на последовательность символов доменного имени и использует LSTM в сочетании с механизмом внимания для построения классификатора доменных имен DGA для достижения быстрой классификации доменных имен. Экспериментальные результаты показывают, что метод имеет хорошие результаты классификации.

Ключевые слова: безопасность; классификация DGA; механизм внимания; LSTM

1. Введение

В последние годы кибератаки переживают взрывной рост, что серьезно угрожает безопасности имущественных данных интернет-пользователей. Доменное имя - важный вид инфраструктуры для кибератак. Оно может использоваться для поддержания связи с клиентом для осуществления возврата данных и доставки команд. Чтобы обойти механизм черных списков и продлить время атаки, злоумышленник обычно использует DGA для генерации новых доменных имен. Алгоритм генерации доменов (DGA) - это алгоритм, используемый кибератаками для периодической генерации доменных имен. Доменное имя, сгенерированное DGA, обычно называется доменным именем DGA. Доменные имена DGA не являются полностью случайными. Разные атакующие организации используют различные алгоритмы генерации доменных имен для разных видов атак, поэтому клиент может установить связь с C&C-сервером, используя доменное имя, сгенерированное DGA. Поскольку доменное имя DGA генерируется периодически, черный и белый список не обязательно существует, что влияет на время реакции организации безопасности на атаку. Определение атаки вредоносного доменного имени позволяет эффективно определить цель атаки, используемые инструменты и вредоносное ПО и т. д., чтобы обеспечить быстрое и эффективное реагирование, значительно снижая ущерб от кибератак.

Для выявления атак с помощью обнаружения доменных имен DGA организации, занимающиеся безопасностью, обычно проводят обратный анализ образцов вредоносного ПО. Они выявляют код и алгоритмы, связанные с DGA, в образцах вредоносного ПО и быстро генерируют доменные имена DGA, которые будут использоваться в соответствующей кибератаке. В свою очередь,

формируется способность обнаруживать соответствующую кибератаку. В 2014 году компания FireEye проанализировала DGA-алгоритм вредоносной программы Srizbi. Согласно алгоритму DGA, сотни DGA-доменных имен Srizbi были отсеяны, что эффективно сдерживало распространение Srizbi. Анализ алгоритма DGA по образцам вредоносного ПО - очень сложная задача, требующая привлечения опытных реверс-аналитиков, которых сейчас особенно мало. В то же время злоумышленник использует различные средства, такие как упаковка, шифрование и запутывание, чтобы не быть перевербованным или увеличить сложность перевербовки, чтобы увеличить время анализа DGA, тем самым продлевая время выживания атаки. Кроме того, для новых типов атак, а также для бесфайловых вредоносных программ часто не удается получить образцы вредоносного ПО, что приводит к невозможности реверсировать алгоритм DGA, используемый атакой. Таким образом, получить алгоритм DGA из образцов вредоносного ПО обратным путем довольно сложно, и в реальности существует множество недостижимых факторов.

Исследователи безопасности могут легко получить WHOIS [1-6] информацию о доменных именах до вступления в силу GDPR. Затем по контекстной информации доменного имени можно определить, является ли оно доменным именем DGA, генерируется ли оно по одному и тому же алгоритму DGA и является ли злоумышленник одним и тем же. WHOIS-информация о доменном имени включает данные о владельце регистрации, адрес электронной почты, мобильный телефон, адрес и т. п. Чтобы избежать слежки, злоумышленник обычно использует поддельную информацию для регистрации. Однако для снижения затрат обычно регистрируют несколько доменных имен с одинаковой регистрационной информацией. Поэтому по регистрационной информации можно определить связь между этими доменными именами. Например, исследователь GReAT [7] обнаружил связь между ними через WHOIS-информацию о доменном имени и определил, что эти атаки были инициированы организацией Winnti attack. Однако для новых доменных имен, появляющихся в сети, сложно в реальном времени получить суждения от получения их WHOIS-информации до создания ассоциаций на основе WHOIS-информации. В то же время в 2018 году начал действовать и внедряться GDPR. После его введения невозможно получить регистрационную информацию о доменном имени у регистратора доменного имени. Платформы безопасности, такие как VirusTotal, больше не предоставляют WHOIS-информацию о доменном имени. Поэтому обнаружение и классификация DGA по контекстной информации, такой как WHOIS доменных имен, больше не представляется возможным.

Поэтому определить, является ли это доменное имя DGA и к какому типу DGA оно относится, можно только проанализировав строку доменного имени. Существует множество работ, основанных на текстовых характеристиках доменных имен. Ядав и другие [8] выполняют обнаружение DGA, используя характеристики распределения символов и 2-граммовый набор символов в доменном имени. Antonakakis et al. [9] используют длину доменного имени, частоту символов, случайность и другие характеристики для неуправляемой кластеризации доменных имен, чтобы достичь обнаружения DGA. Традиционный метод машинного обучения имеет проблемы, такие как извлечение признаков, опора на экспертов, что позволяет злоумышленникам легко обойти его. Для решения проблем, существующих в текущей классификации доменных имен DGA, в данной статье предлагается метод классификации доменных имен DGA, основанный на LSTM с механизмом внимания на основе исследования DGA и доменных имен DGA. Этот метод не требует ни реверсирования образца вредоносного ПО, ни использования контекстной информации, такой как WHOIS доменного имени, а использует только последовательность символов доменного имени. Для последовательности символов доменного имени алгоритм LSTM в сочетании с механизмом внимания используется для построения классификатора доменных имен DGA, чтобы достичь быстрой и точной классификации доменного имени.

Вклад данной работы включает в себя два аспекта:

1. Мы используем только последовательность символов доменного имени для классификации доменных имен DGA, чтобы доказать, что последовательность символов содержит признаки DGA.
2. Мы объединили LSTM с механизмами внимания и применили их к классификации доменных имен DGA, чтобы доказать, что веса символов в доменных именах DGA различны.

Остальная часть данной работы организована следующим образом. В разделе 2 мы кратко представим смежные работы по обнаружению и классификации доменов DGA. В разделе 3 мы кратко описываем методы, используемые в нашем подходе. В разделе 4 мы описываем архитектуру нашего подхода. В разделе 5 представлены экспериментальные результаты, а в разделе 6 - резюме всей работы.

2. Связанные работы

Существует множество работ, в которых для обнаружения и классификации DGA используются динамические характеристики доменных имен, включая возможность их преобразования в IP-адреса, географическое распределение IP-адресов и т. д. В 2010 году Антонакокис и другие [10] создали динамическую систему оценки доменных имен, которая использует три типа признаков, включая сетевые (такие как исторический номер IP-адреса доменного имени, географическое распределение, AS-домен и т. д.), доменные (такие как длина доменного имени, распределение символов и т. д.) и доказательные (включая то, связано ли оно с известным семейством вредоносных программ, разрешается ли оно на вредоносный IP и т. д.). При тестировании в реальных условиях точность системы достигла 96,8 %. В 2010 году Ядав и другие [11] разработали методику обнаружения доменных потоков в трафике DNS путем поиска закономерностей, присущих доменным именам, которые были сгенерированы алгоритмически, в отличие от тех, которые были сгенерированы человеком. Более того, они применили эту методику к трассировке пакетов, собранных у интернет-провайдера первого уровня, и показали, что могут автоматически обнаруживать доменные потоки, используемые ботнетом Conficker, с минимальным количеством ложных срабатываний. В 2011 году Бильге и другие [4] предложили технологию обнаружения вредоносных доменных имен, основанную на методе пассивного анализа доменных имен. Они извлекли из трафика 15 типов признаков, включая время жизни доменного имени, схожесть периодов, количество обращений, количество разобранных IP-адресов, наличие общих IP-адресов с другими доменными именами, соотношение цифровых символов и длину самой длинной значимой подстроки и т. д. Наконец, классификатор был построен с помощью алгоритма дерева решений J48. После проверки на реальных условиях точность обнаружения этого метода достигает 98 %. В 2012 году Антонакокис и другие [9] представили новую технику обнаружения случайно сгенерированных доменов без реверсирования. Их идея заключалась в том, что большинство сгенерированных DGA (случайных) доменов, которые запрашивает бот, приводят к ответам в виде несуществующих доменов (NXDomain), и что боты из одного ботнета (с одним и тем же алгоритмом DGA) будут генерировать одинаковый трафик NXDomain. Проведя многомесячный этап оценки, они показали, что система может достичь очень высокой точности обнаружения. В 2013 году Кришнан и другие [12] предложили метод обнаружения атак с помощью последовательной проверки гипотез. Они полагают, что хосты, зараженные вредоносным ПО, будут демонстрировать поведение сканирования доменных имен. Большинство просканированных доменных имен не могли разрешить IP-адрес. Такое поведение было ненормальным. Сначала обнаруживалось такое аномальное поведение, затем анализировалось доменное имя, запрашиваемое терминалом, и с помощью фильтра Zipf определялось вредоносное доменное имя.

Поскольку динамический анализ требует больших вычислительных ресурсов и занимает много времени, во многих работах для обнаружения и классификации использовались только символьные и последовательные признаки доменного имени. В 2012 году Ядав и другие [8] предложили метод обнаружения доменных имен DGA. На эту работу их вдохновило наблюдение, что разница в распределении символов между обычным доменным именем и доменным именем DGA довольно велика. Они использовали символы в доменном имени и характеристики распределения 2-граммового набора символов, а затем применили алгоритм edit distance и Jaccard distance [13]. Фактический коэффициент обнаружения метода составил 83,87 %. В 2012 году Антонакокис и другие [9] предложили метод определения доменных имен DGA по неразрешенным доменным именам. Сначала они использовали характеристики длины доменного имени, частоты символов, случайности и другие характеристики для неуправляемой кластеризации доменных имен. Затем они использовали модель классификации на основе Маркова для определения атаки, стоящей за доменным именем, и отфильтровали активное доменное имя, которое являлось доменным именем C&C. В 2014 году Билге и другие [14] систему под названием EXPOSURE для обнаружения доменов DGA в режиме реального времени, применяя 15 уникальных признаков, сгруппированных в четыре категории. Они провели контролируемый эксперимент с большим набором реальных данных, состоящим из миллиардов DNS-запросов. Результаты показали, что система хорошо работает на практике и позволяет автоматически определять широкую категорию вредоносных доменов и связи между ними. В 2014 году Скьявони и другие [15] создали систему отслеживания и разведки доменных ботнетов DGA. Сначала использовались признаки на основе символов и IP-адресов для идентификации DGA и не-DGA доменных имен, а затем DGA доменные имена были сгруппированы для определения ботнета, к которому они принадлежат; они протестировали более 1 миллиона доменных имен, и точность обнаружения составила 94,8 %. В 2016 году Вудбридж и другие [16] представили классификатор DGA, который

использует сети долговременной кратковременной памяти (LSTM) для предсказания DGA в реальном времени без необходимости использования контекстной информации или созданных вручную признаков. Результаты экспериментов показали, что метод значительно превосходит все современные методики. В 2017 году Ю и другие [17] предложили метод обнаружения доменных имен DGA, основанный на глубоком обучении. Для построения модели классификации использовались алгоритмы CNN и LSTM. Показатели точности составили 72,89 % и 74,05 % соответственно.

3. Теоретическая основа

3.1. Рекуррентная нейронная сеть (PHC)

Рекуррентная нейронная сеть (PHC) [18], обладающая свойствами обработки исторических данных и моделирования памяти, является важной ветвью глубокого обучения. С биологической точки зрения, PHC - это простая имитация кольцевой связи биологической нейронной системы, которая подходит для задач с временными рядами, таких как распознавание рукописного шрифта, распознавание речи и обработка естественного языка. Оригинальная RNN состоит из входного вектора, x; состояния скрытого слоя, s; выходного вектора, h; весового параметра, U, информации о входной последовательности; весового параметра, W, состояния скрытого слоя; весового параметра, V, информации о выходной последовательности; и тому подобного.

S_t вычисляется на основе состояния скрытого слоя s_{t-1} в предыдущий момент времени и входного сигнала x_t в текущий момент времени. Пусть функция активации состояния скрытого слоя равна f , тогда текущее состояние скрытого слоя, s_t, вычисляется как

$$s_t = f(Ws_{t-1}, Ux_t) \quad (1)$$

Если предположить, что функция активации на выходе равна g, то выход рассчитывается как

$$h_t = g(Vs_t) \quad (2)$$

Из формулы видно, что состояние скрытого слоя s_t RNN имеет функцию памяти для последовательности, и информация о последовательности может быть сохранена состоянием скрытого слоя. Доменное имя DGA - это последовательность символов, которая автоматически строится с помощью алгоритмов. Доменное имя DGA может быть смоделировано с помощью RNN для обнаружения или классификации.

3.2. Длительная кратковременная память (LSTM)

Однако, будучи ограниченной структурой, оригинальной RNN трудно выучить достоверные данные в долгосрочной зависимой последовательности данных. Входные данные, которые находятся далеко от текущего момента, не могут способствовать обновлению параметров модели текущего времени, так называемая проблема исчезновения градиента. Длина доменного имени DGA обычно очень велика. Например, доменное имя коммутатора Wannacry из 41 символа, и на практике часто встречаются доменные имена DGA длиной более 70. Наиболее популярным решением проблемы исчезновения градиента в PHC является использование структуры LSTM [19] вместо сигмоидальной функции активации в оригинальной PHC.

LSTM - это специальная искусственная архитектура RNN [19], используемая в области глубокого обучения. LSTM оказалась более эффективной, чем традиционные модели RNN, при решении проблем долгосрочных зависимостей. LSTM и RNN схожи по временным характеристикам, но способ вычисления состояния нейронов скрытого слоя отличается. Каждый блок памяти LSTM включает в себя четыре основных элемента: входные ворота, ворота забывания, выходные ворота и самозацикливающиеся связанные блоки. Таким образом, выходное значение контролируется между 0 и 1, отвечая за описание того, сколько пройдено. В момент времени t x_t представляет собой вход; i_t - значение активации входных ворот; i_t - значение активации входных ворот; f_t - значение активации забывающих ворот; o_t - значение активации выходных ворот; h_t и h_{t-1} - выходы ячейки памяти в моменты времени t и $t-1$, соответственно; C_t и C_{t-1} - состояния ячейки памяти в моменты времени t и $t-1$, соответственно; C' - состояние-кандидат ячейки памяти. W_i, U_i, W_c, U_c, W_f, U_f, W_o, U_o и т. д. - веса соответствующих ворот в основном блоке памяти. b_i, b_c, b_f, b_o и т. д. - смещения соответствующих ворот в блоке памяти. σ на рисунке - активационная функция.

Состояние ячейки памяти в момент времени *t* выглядит следующим образом,

$$C_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \times tan(W_c x_t + U_c h_{t-1} + b_c) + \sigma(W_f x_t + U_f h_{t-1} + b_f) \times C_{t-1} \quad (3)$$

Выходной сигнал блока памяти t - это

$$H_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \times tan(C_t) \quad (4)$$

Такой механизм работы блока памяти LSTM позволяет долго хранить и получать доступ к информации о последовательности, тем самым уменьшая проблему исчезновения градиента. Он подходит для построения моделей обнаружения и классификации доменных имен DGA. Вудбридж и другие [16] использовали модель обнаружения и классификации доменных имен DGA, построенную с помощью LSTM, и получили хорошие результаты.

3.3. Механизм внимания

Проанализировав алгоритм DGA, мы обнаружили, что злоумышленник может контролировать доменное имя, сгенерированное DGA; то есть доменное имя, сгенерированное в одном цикле, не может быть продублировано зарегистрированным доменным именем другого человека, но может поразить злоумышленника регистрацией доменного имени, обычно добавляя некоторые ограничения. Например, в доменном имени Banjori преобразуются только первые четыре буквы, а часть после доменного имени остается неизменной. Поэтому, пока последняя часть обнаруживается, доменное имя в принципе можно рассматривать как DGA и не обязательно обращать внимание на все доменное имя. Поэтому в данной статье представлен механизм внимания, который в модели классификации уделяет разное внимание разным частям входного доменного имени, эффективно эффект классификации доменных имен DGA.

В последние годы механизм внимания широко используется в различных типах задач глубокого обучения, таких как машинный перевод, распознавание изображений и речи. По сравнению с простыми конволюционными нейронными сетями (CNN) и рекуррентными нейронными сетями (RNN) были достигнуты лучшие результаты. Использование самовнимания в модели BERT [20] значительно усиливает эффект машинного перевода. Когда человек смотрит на картинку, он быстро сканирует глобальное изображение, чтобы найти целевую область, на которой нужно сосредоточиться, то есть фокус внимания, а затем вкладывает больше ресурсов внимания в эту область, чтобы получить детали цели, подавляя при этом другую бесполезную информацию. Механизм внимания в глубоком обучении по сути похож на механизм селективного зрительного внимания человека. Цель - выбрать из множества информации ту, которая более важна для выполнения текущих задач. Существует ряд популярных механизмов внимания, таких как внимание на основе содержания [21], внимание на основе (глобального/локального) местоположения [22], самофокусировка [23] и т. д. В данной работе рассматривается глобальное внимание [22] и используется более простой механизм внимания для снижения вычислительной производительности.

Наиболее важным аспектом механизма глобального внимания является расчет его положения Вес, если предположить, что $H = [h_1, h_2, \dots, h_m]^T$, где $h_i = [h_i^1, h_i^2, \dots, h_i^n]$, то H - матрица из $m \times n$:

$$H = \begin{bmatrix} h_1^1 & \cdots & h_1^n \\ \vdots & \ddots & \vdots \\ h_m^1 & \cdots & h_m^n \end{bmatrix} \quad (5)$$

Нам нужно вычислить позиционный вес для h_i; сначала транспонируем H, тогда H^T будет матрицей из $n \times m$:

$$H^T = \begin{bmatrix} h_1^1 & \cdots & h_m^1 \\ \vdots & \ddots & \vdots \\ h_1^n & \cdots & h_m^n \end{bmatrix} \quad (6)$$

Затем, используя функцию softmax для каждой строки H^T, получаем следующий результат,

$$S^T = \begin{bmatrix} \frac{h_1^1}{\sum_j^m h_j^1} & \cdots & \frac{h_1^1}{\sum_j^m h_j^1} \\ \vdots & \ddots & \vdots \\ \frac{h_1^n}{\sum_j^m h_j^n} & \cdots & \frac{h_1^n}{\sum_j^m h_j^n} \end{bmatrix} = \begin{bmatrix} a_1^1 & \cdots & \\ \vdots & \ddots & \vdots \\ a_1^n & \cdots & a_m^n \end{bmatrix} \quad (7)$$

Затем транспонируйте ST, чтобы получить требуемую весовую матрицу S и матрицу внимания:

$$S = \begin{bmatrix} a_1^1 & \cdots & \\ \vdots & \ddots & \vdots \\ a_m^1 & \cdots & \end{bmatrix} \quad (8)$$

Наконец, H поэлементно умножается на S, чтобы получить механизм внимания:

$$H \times S = \begin{bmatrix} h_1 a_1 & \cdots & h_n a_n \\ \vdots & \ddots & \vdots \\ h_1 a_1 & \cdots & h_n a_n \end{bmatrix} \quad (9)$$

Механизм внимания может быть размещен перед слоем LSTM или за ним. В данной работе показано, что механизм внимания лучше работает за слоем LSTM. Поэтому в данной работе механизм внимания размещается за слоем LSTM.

4. Методология

4.1. Обзор

Как видно на рисунке 1, метод данной статьи состоит из обучающих и тестовых фраз. Как в обучающих, так и в тестовых фразах имена доменов из обучающего и тестового наборов данных должны быть предварительно обработаны. Для каждого доменного имени после извлечения строк DGA, добавления и встраивания они преобразуются в матрицу 54× 128. Затем, в обучающей фразе, матрицы обучающего набора данных подаются в сеть глубокого обучения для создания модели классификации. Наконец, матрицы тестового набора данных проверяются моделью классификации. Конкретное описание выглядит следующим образом.

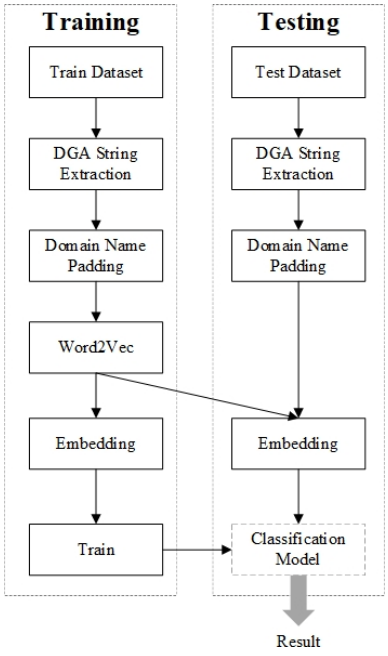

Рисунок 1. Обзор метода классификации Domain Generation Algorithm (DGA).

4.2. Извлечение строк DGA

Как правило, чтобы улучшить управляемость и избежать быстрой блокировки, большинство злоумышленников регистрируют доменное имя второго уровня. Например, "h7smcnrwlddsdn34fgv.info" - это доменное имя DGA, зарегистрированное для вредоносного кода Sality. Однако регистрация доменного имени второго уровня сопряжена с определенными затратами. Более того, даже если используется поддельная информация, может остаться отслеживаемая информация, IP, и есть вероятность, что источник будет отслежен. В то же время некоторые атаки используют динамические службы доменных имен для генерации доменных имен третьего уровня, чтобы сэкономить расходы на атаку, например "blackshadespro.no-ip.org". Доменное имя второго уровня и доменное имя третьего уровня в службе динамических доменных имен не обязательно полностью разделены. Два типа доменных имен могут использоваться одновременно в одной атаке или в разных атаках, проводимых одной и той же организацией. В данном документе для обнаружения и классификации доменных имен используется строка, сгенерированная DGA, например "h7smcnrwlddsdn34fgv" в "h7smcnrwlddsdn34fgv.info", "blackshadespro" в "blackshadespro.no-ip.org" и так далее. Поэтому для извлечения строки доменного имени DGA из доменного имени мы следуем следующим рекомендациям.

1. Если это доменное имя второго уровня, извлекается часть доменного имени второго уровня,
2. Если это доменное имя третьего уровня, сначала определите, является ли доменное имя второго уровня доменным именем динамической службы доменных имен, например "no-ip.com", "afraid.org", "duckdns.com", "dnsdynamic.org", "dyndns.net", "dynu.com" и т.д., если да, то извлекается часть доменного имени третьего уровня.
3. Если доменное имя второго уровня в доменном имени третьего уровня не является доменным именем поставщика услуг динамических доменных имен, извлекается самая длинная строка.
4. В противном случае извлекаем самую длинную строку.

4.3. Подложка для доменного имени

Длина доменного имени не является фиксированной. В некоторых работах [9,10] длина доменного имени используется как одна из характеристик для обнаружения и классификации доменных имен DGA. В таблице 1 приведены длины 11 распространенных доменных имен DGA. Из таблицы видно, что длина различных доменных имен DGA обычно отличается, однако метод, предложенный в данной работе, требует фиксированной длины в качестве входных данных.

Таблица 1. Одиннадцать типов длины доменного имени DGA.

DGA	Длина	Пример
банджори	10	pdtmstring
corebot	23	a0c4e8sr70oluhsf3t1h1va
cryptolocker	8	rifxkpdx
dircrypt	10	xzdiobjady
кракен	8	iuhqhbmq
локи	17	qqeuxqbetndnsclkm
pykspa	9	Фольмецика
qakbot	20	gutkdzfamdgsjbhpuoyb
рамдо	8	kuekesqm
рамнит	9	byqdmekgd
simda	23	ювелирыдатывыжолывофох

Обычно в качестве фиксированной выбирается самая длинная длина в наборе данных, а затем короткое доменное имя добавляется. Однако, учитывая масштабируемость модели, в данной работе проводится статистика длине доменного имени DGA, опубликованного на сайте Bambenek Consulting [24]. Результат показан на рис. 2; большинство доменных имен DGA сосредоточены в интервале 10-20, а самое длинное доменное имя DGA - 44, что охватывает больше возможностей и гарантирует эффективность метода.

Добавив 10, мы используем 54 в качестве входной длины; все доменные имена дополняются до 54. В данной статье для завершения доменного имени используется символ "*", который не допускается в доменном имени.

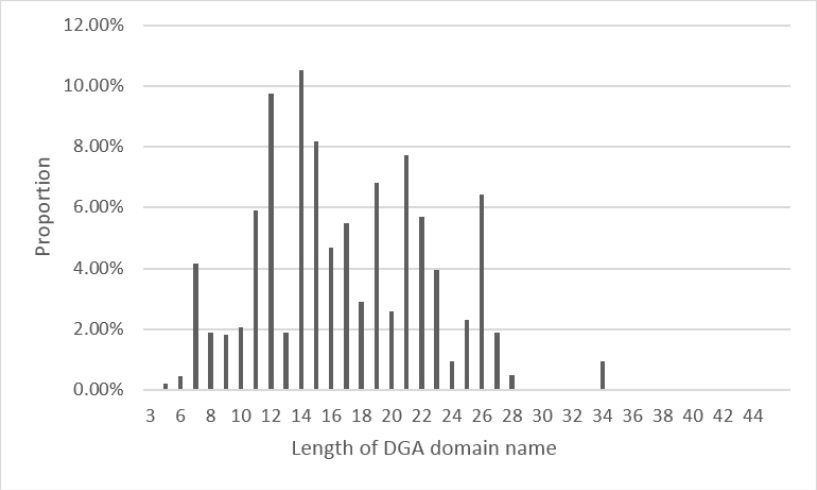

Рисунок 2. Распределение длины доменных имен DGA.

4.4. Встраивание

После заполнения каждого доменного имени форма будет иметь вид $d = [a_1, a_2, \text{- - - - }, a_{54}]$, где подстрочный индекс a указывает на местоположение. Используя символы как слова, каждое доменное имя можно представить как предложение, состоящее из символов. Далее, используя модель CBOW Word2Vec, на полном обучающем множестве вычисляется вектор слов всех символов в доменном имени. В данной работе размерность вектора слов установлена на 128, учитывая, что количество элементов в ASCII равно 128. Вектор слов каждого
символ может быть выражен как $W_a = [x_1^a, x_2^a, \text{- - - - }, x_{128}^a]$, где a представляет символ в домене
имя. Затем векторы слов упорядочиваются в порядке следования символов в доменном имени, так что каждое доменное имя преобразуется в матрицу 54× 128, как показано ниже,

$$M^d = \begin{bmatrix} W^{a_1} \\ W^{a_2} \\ \cdot \\ W^{a54} \end{bmatrix} = \begin{bmatrix} x_1^{a1} & \text{- - -} & x_{128}^{a1} \\ & \cdot \cdot \cdot & \\ x_1^{a54} & \text{- - -} & x_{128}^{a54} \end{bmatrix} \quad (10)$$

4.5. Структура сети глубокого обучения

В данной работе мы используем LSTM в сочетании с механизмом внимания, как показано на рисунке 3; функции каждого слоя следующие,

1. ВХОД: Входной слой, доменное имя преобразуется в матрицу размерностью 54× 128 после выравнивания длины и встраивания, таким образом, размерность входа составляет 54× 128.
2. LSTM: слой LSTM, выход последовательности и выход 54× 128 вектор признаков.
3. ATTENTION: Механизм внимания, согласно разделу 3.3, и вывод 54× 128 вектора признаков.
4. FC: Полностью связанный слой, который растягивает вектор признаков, выведенный ATTENTION. Каждый пиксель представляет собой единицу. Выходной признак составляет 6912 единиц при полностью подключенного слоя, а вероятность DROPOUT установлена на 0,5.
5. ВЫХОД: Выходной слой, этот слой полностью связан со слоем FC; выходная длина - это требуемое количество классификаций, которое представляет, к какой классификации относятся извлеченные признаки; функция классификации - Softmax.

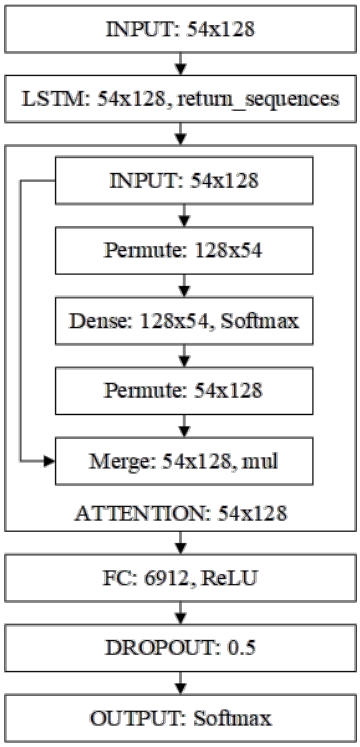

Рисунок 3. Дизайн структуры классификационной сети на основе механизма внимания и LSTM.

Параметры в структуре сети следующие.

- Классификатор: На основе характеристик доменов DGA мы используем классификатор Softmax, чтобы определить, к какому типу относится домен. Суть функции Softmax заключается в отображении K-мерного произвольного вещественного вектора в другой K-мерный вещественный вектор, где значение каждого элемента вектора находится в интервале [0, 1], как показано в формуле (11), где v_j - *j-й* элемент вектора, а Softmax-значение элемента равно $f\,tmax(v_j)$,

$$поэтому\ f\,tmax_j(v) = \frac{v_j}{\sum_{k=1}^{K} vk} \tag{11}$$

- Функция потерь: Когда модель обучена, потери рассчитываются в соответствии с функцией потерь, а затем обратное распространение (BP) используется для параметров. В данной работе в качестве функции потерь модели используется функция потерь категориальной перекрестной энтропии.

$$L(Y, \hat{Y}) = -\sum_i y_i \times \log \hat{y}_i \tag{12}$$

- Функция активации: Формула функции активации ReLU выглядит следующим образом. Эта функция может удовлетворить разреженность в бионике. Она активирует блоки, когда входное значение превышает определенное число, и может быстро сходиться в алгоритме стохастического градиентного спуска. Градиент функции равен 0 или постоянен, что может облегчить проблему исчезновения градиента, тем самым улучшая точность и скорость обучения нейронной сети. Поэтому в данной работе ReLU используется в качестве функции активации в двух конволюционных и двух полносвязных слоях.

$$relu(x) = \max(0, x) \tag{13}$$

В то же время, чтобы предотвратить перебор, в структуру сети добавляется отсеивающий слой. Отсеивающий слой предотвращает чрезмерную подгонку, предотвращая синергию определенных признаков. При каждом обучении,

единицы удаляются случайным образом, что позволяет одной единице выглядеть независимой от другой, предотвращая взаимовлияние признаков и уменьшая передачу ошибочной информации.

5. Экспериментальная оценка

В этом разделе мы сначала опишем набор данных, используемый в экспериментах. Далее мы представляем эксперимент, доказывающий определенные возможности нашего метода. Наконец, мы сравниваем наш метод с предыдущими работами по многим параметрам.

5.1. Набор данных DGA

В качестве DGA-доменов были собраны данные OSINT DGA от Bambenek Consulting [24]. Затем мы отфильтровали классы с более чем 5000 для обучения, и в итоге было собрано 765 091 DGA-домен. В то же время первый миллион доменных имен сайта Alexa [25] был собран как обычные домены. Таким образом, был сформирован набор данных, включающий обычные домены и домены DGA, общим числом 1 675 404, как показано в таблице 2.

Таблица 2. Детали набора данных.

DGA	Сумма
банджори	439,223
Пост	66,000
тинба	65,603
рамнит	47,510
некур	32,768
qakbot	20,000
мурофет	14,260
pykspa	14,215
ranbyus	13,960
simda	13,681
shiotob/urlzone/bebloh	12,521
dyre	7998
Cryptolocker	6000
nymaim	6000
локи	5352
Алекс	910,313

5.2. Экспериментальные результаты

В данной работе для создания наборов для оценки использовалась случайная выборка. Сначала набор образцов был случайным образом разделен в среднем на 10 частей, одна из которых была взята в качестве тестового набора. Один из оставшихся девяти образцов был использован в качестве проверочного набора, а остальные восемь - в качестве обучающего набора. Соотношение итоговых обучающего, проверочного и тестового наборов составляет 8:1:1. После обучения тестовый набор использовался для проверки модели классификации. Результаты эксперимента приведены в таблице 3.

Из таблицы 3 видно, что средний показатель точности составляет 95,05%, средний показатель запоминания - 95,14%, а средний показатель F_1 - 95,48%. Точность составляет 95,14 %, что является очень высоким показателем. Матрица путаницы для многоклассового классификатора LSTM с механизмом внимания показана на рисунке 4. Блоки на рисунке представляют собой доли доменов, принадлежащих к семействам DGA по вертикальной оси, классифицированных как семейства DGA по горизонтальной оси, где 0 изображен белым цветом, а 1 - черным. Как видно на рисунке 4, большое количество DGA Cryptolocker, DGA Locky и DGA Necurs классифицируются как Ramnit. Мы проанализировали эти DGA и обнаружили, что они очень похожи. Реми Коэн [26] отметил, что Ramnit приобрел части кода Zeus и стал банковским трояном после утечки исходного кода Zeus. Лимор Кессем [27] предположил, что операторы Necurs в ранние годы были связаны с самым центром элиты Zeus, а Том Спринг [28] рассказал нам, что программа Locky ransomware ожила благодаря ботнету Necurs. Можно предположить, что между Cryptolocker и Locky существует определенная связь,

Locky, Necurs и Ramnit; именно поэтому некоторые DGA Cryptolocker, Locky и Necurs классифицируются как Ramnit.

Таблица 3. Экспериментальные результаты нашего метода.

Тип домена	Точность	Отзыв	F_1 Оценка	Поддержка
nymaim	0.3988	0.1115	0.1743	601
ranbyus	0.4672	0.8455	0.6018	1346
мурофет	0.7641	0.7207	0.7418	1443
pykspa	0.8972	0.7207	0.7994	1393
локи	0.0000	0.0000	0.0000	574
шиотоб	0.9751	0.9251	0.9494	1268
банджори	0.9998	1.0000	0.9999	43,808
некур	0.6651	0.1722	0.2735	3241
Cryptolocker	0.1000	0.0018	0.0035	562
simda	0.9264	0.9669	0.9462	1418
dyre	1.0000	1.0000	1.0000	797
Пост	0.9994	0.9998	0.9996	6644
тинба	0.9259	0.9920	0.9578	6498
qakbot	0.7862	0.5013	0.6122	1973
рамнит	0.4688	0.7525	0.5777	4856
Алекс	0.9898	0.9956	0.9927	91,119
среднее/общее количество	0.9505	0.9514	0.9458	167,541

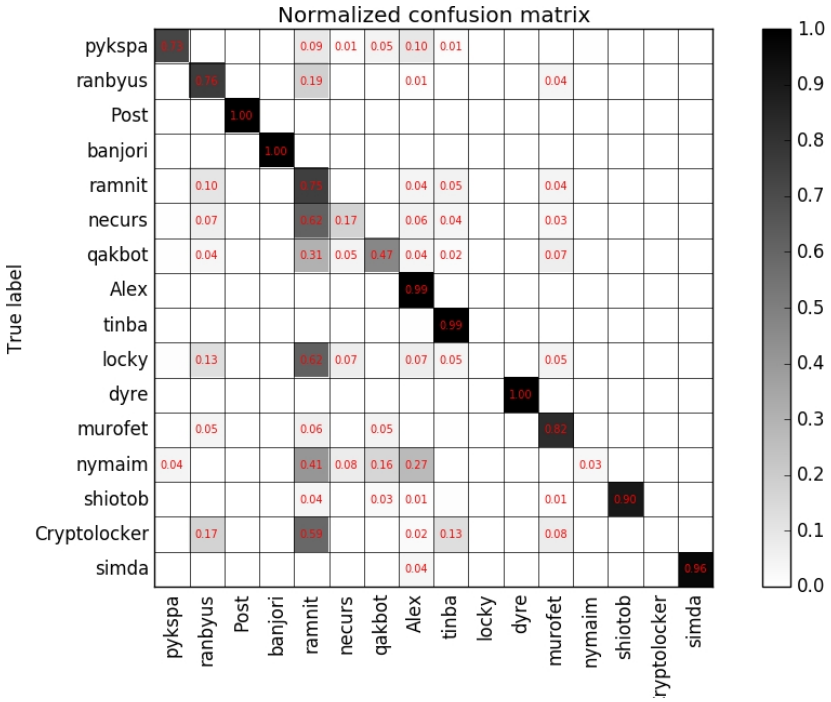

Рисунок 4. Матрица запутанности для модели долговременной кратковременной памяти (LSTM) с механизмом внимания.

5.3. Сравнение и обсуждение работ

Ядав и другие [8] предложили метод обнаружения доменных имен DGA. На эту работу их вдохновило наблюдение, что разница в распределении символов между обычным доменным именем и доменным именем DGA довольно велика. Они использовали символы в доменном имени и характеристики распределения 2-граммового набора символов, а затем применили алгоритм edit distance и Jaccard distance [13]. Хотя Вудбридж и другие [16] в 2016 году предложили метод обнаружения и классификации DGA с помощью алгоритма LSTM, они обнаруживали и классифицировали только строки доменных имен, и в ходе экспериментов было подтверждено, что этот метод обладает очень хорошими эффектами обнаружения и классификации. По сравнению с

В данном методе отсутствует механизм внимания. Мы сравнили наш метод с методами Ядава и др [8] и Вудбриджа и др [16], и результаты приведены в таблице 4:

Таблица 4. Результаты экспериментов Вудбриджа и других [16].

Тип домена	Ядав и другие [8]			Вудбридж и др [16]			Наш метод			Поддержка
	Точность	Recall	F_1 Оценка	Точность	Recall	F_1 Оценка	Точность	Recall	F_1 Оценка	
Алекс	0.9612	0.9863	0.9736	0.9865	0.9970	0.9917	0.9956	0.9956	0.9927	91,119
банджори	0.9741	0.9889	0.9814	0.9998	1.0000	0.9999	1.0000	1.0000	0.9999	43,808
Cryptolocker	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0018	0.0018	0.0035	562
dyre	0.9820	1.0000	0.9909	0.9975	1.0000	0.9987	1.0000	1.0000	1.0000	797
локи	0.0294	0.0038	0.0067	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	574
мурофет	0.5139	0.4861	0.4996	0.7441	0.7477	0.7459	0.7207	0.7207	0.7418	1443
некур	0.2609	0.0253	0.0461	0.6225	0.1725	0.2701	0.1722	0.1722	0.2735	3241
nymaim	0.1080	0.0409	0.0593	0.3621	0.1048	0.1626	0.1115	0.1115	0.1743	601
Пост	0.9897	0.9913	0.9905	0.9995	1.0000	0.9998	0.9998	0.9998	0.9996	6644
pykspa	0.7253	0.5690	0.6377	0.9407	0.6827	0.7912	0.7207	0.7207	0.7994	1393
qakbot	0.7160	0.3483	0.4686	0.7970	0.4774	0.5971	0.5013	0.5013	0.6122	1973
рамнит	0.3541	0.5580	0.4333	0.4711	0.7360	0.5745	0.7525	0.7525	0.5777	4856
ranbyus	0.0000	0.0000	0.0000	0.4599	0.8522	0.5974	0.8455	0.8455	0.6018	1346
шиотоб	0.9349	0.6584	0.7727	0.9873	0.9211	0.9531	0.9251	0.9251	0.9494	1268
simda	0.8203	0.6839	0.7459	0.9688	0.8533	0.9074	0.9669	0.9669	0.9462	1418
тинба	0.6709	0.8588	0.7533	0.9264	0.9912	0.9577	0.9920	0.9920	0.9578	6498
среднее/общее количество	0.8960	0.9127	0.9001	0.9482	0.9504	0.9445	0.9505	0.9514	0.9458	167,541

Из таблицы 4 видно, что наш метод имеет гораздо более высокие показатели точности, запоминания и F_1 по сравнению с работой Ядава и других [8]. сравнению с работой Woodbridge et al. [16] улучшение метода не очевидно.

Поэтому для сравнения метода Вудбриджа и др. [16] с нашим методом мы провели эксперимент с 10-кратной перекрестной валидацией. Набор образцов был разделен в среднем на 10 частей, одна из которых каждый бралась в качестве тестового набора. Одна из оставшихся девяти выборок использовалась в качестве проверочной, а остальные восемь - в качестве обучающей. Соотношение итоговых обучающего, проверочного и тестового множеств составляет 8:1:1. Каждый эксперимент проводился в течение 20 эпох обучения. После обучения тестовый набор использовался для проверки модели классификации. Результаты 10 экспериментов показаны на рисунке 5. Как видно из рисунка 5, точность нашего метода превосходит точность метода Вудбриджа [16] в каждом эксперименте. Это доказывает, что веса различных символов в разных позициях в доменном имени DGA различны, и для классификации DGA необходим механизм внимания.

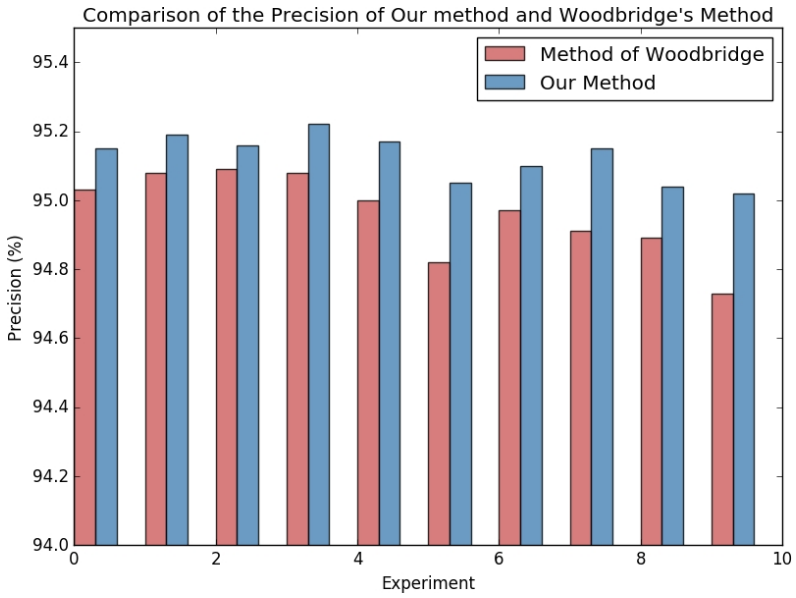

Рисунок 5. Сравнение точности нашего метода и метода Вудбриджа.

6. Выводы

В настоящее время существует множество проблем в классификации доменных имен DGA. Например, трудно получить алгоритм DGA путем реверсирования образцов вредоносного ПО, а информацию WHOIS, вызванную внедрением GDPR, уже невозможно легко получить. Основываясь на исследовании алгоритма DGA и доменного имени DGA, мы предлагаем метод классификации доменных имен DGA, основанный на LSTM с механизмом внимания. Этот метод больше не реверсирует образец вредоносного ПО, не использует контекстную информацию, такую как WHOIS доменного имени, а использует только последовательность символов доменного имени. Для получения последовательности символов доменного имени каждое доменное имя преобразуется в матрицу фиксированной размерности путем вставки и встраивания. Затем с помощью алгоритма LSTM с механизмом внимания строится модель классификации доменных имен DGA, позволяющая быстро и точно классифицировать доменные имена. Экспериментальные результаты показывают, что сочетание механизма внимания с LSTM позволяет эффективно классифицировать доменные имена DGA. Таким образом, доменное имя DGA ассоциируется с сетевой атакой, а время реакции на инцидент с атакой сокращается. Метод учитывает веса различных символов в разных позициях в доменном имени DGA и имеет более высокую точность классификации, чем простой алгоритм LSTM.

Авторский вклад: Концептуализация, Y.Q. и B.Z.; методология, Y.Q.; программное обеспечение, Y.Q.; валидация, Y.Q.; формальный анализ, W.Z.; исследование, Y.Q. и H.W.; ресурсы, B.Z.; курирование данных, Y.Q. и H.W.; написание - подготовка первоначального проекта, Y.Q.; написание - рецензирование и редактирование, B.Z. и A.K.S.; визуализация, H.W.; руководство, B.Z. и W.Z.; администрирование проекта, W.Z. и B.Z.; получение финансирования, B.Z. и W.Z.

Финансирование: Работа выполнена при поддержке Ключевой программы исследований и разработок провинции Гуандун, 2019B010136001, и Проекта лаборатории Пэн Чэн провинции Гуандун, PCL2018KP004 и PCL2018KP005.

Благодарности: Мы с благодарностью принимаем полезные замечания и предложения рецензентов и редакторов.

Конфликты интересов: Авторы заявляют об отсутствии конфликта интересов.

Ссылки

1. McGrath, D.K.; Gupta, M. Behind Phishing: An Examination of Phisher Modi Operandi. В материалах 1-го семинара Usenix по крупномасштабным эксплойтам и возникающим угрозам, Сан-Франциско, Калифорния, США, 15 апреля 2008 г.; Ассоциация USENIX: Berkeley, CA, USA, 2008; pp. 4:1-4:8.
2. Ма, Дж.; Соул, Л.К.; Сэвидж, С.; Воелкер, Г.М. За пределами черных списков: Learning to Detect Malicious Web Sites from Suspicious URLs. В материалах 15-й Международной конференции ACM SIGKDD по обнаружению знаний и добыче данных, Париж, Франция, 28 июня - 1 июля 2009 г.; ACM: Нью-Йорк, штат Нью-Йорк, США, 2009; pp. 1245-1254. [CrossRef].
3. Felegyhazi, M.; Kreibich, C.; Paxson, V. On the Potential of Proactive Domain Blacklisting. В материалах 3-й конференции USENIX по крупномасштабным эксплойтам и возникающим угрозам: Botnets, Spyware, Worms, and More, San Jose, CA, USA, 17 April 2010; USENIX Association: Berkeley, CA, USA, 2010; p. 6.
4. Бильге, Л.; Кирда, Э.; Крюгель, К.; Балдуцци, М. EXPOSURE: Поиск вредоносных доменов с помощью пассивного анализа DNS. В материалах 18-го симпозиума по безопасности сетей и распределенных систем, Сан-Диего, Калифорния, США, 6 февраля 2011 г.; Internet Society: Reston, VA, USA, 2011; pp.1-17.
5. Canali, D.; Cova, M.; Vigna, G.; Kruegel, C. Prophiler: Быстрый фильтр для крупномасштабного обнаружения вредоносных веб-страниц. В материалах 20-й Международной конференции по Всемирной паутине, Хайдарабад, Индия, 28 марта-1 апреля 2011 года; ACM: Нью-Йорк, штат Нью-Йорк, США, 2011; стр. 197-206. [CrossRef].
6. Zhang, J.; Saha, S.; Gu, G.; Lee, S.; Mellia, M. Systematic Mining of Associated Server Herds for Malware Campaign Discovery. В материалах 35-й Международной конференции 2015 IEEE по распределенным вычислительным системам, Колумбус, штат Огайо, США, 29 июня - 2 июля 2015 г.; стр. 630-641. [CrossRef].
7. ПОДРОБНЕЕ Виннти. Больше, чем просто игра. 2013. Доступно онлайн: https://securelist.com/analysis/internal-threats-reports/37029/winnti-more-than-just-a-game/ (дата обращения: 27 июня 2019 г.).
8. Ядав С., Редди А.К.К., Редди А.Л.Н., Ранджан С. Обнаружение алгоритмически сгенерированных атак на домен с помощью анализа DNS-трафика. *IEEE/ACM Trans. Netw.* **2012**, *20*, 1663-1677. [CrossRef].

9. Антонакокис М., Пердиши Р., Наджи Й., Василоглоу Н., Абу-Нимех С., Ли В., Дагон Д. От бросового трафика до ботов: Обнаружение роста вредоносного ПО на базе DGA. В материалах 21-й конференции USENIX по безопасности, Бельвью, штат Вашингтон, США, 8-10 августа 2012 г.; Ассоциация USENIX: Беркли, Калифорния, США, 2012; стр. 24.
10. Антонакокис М., Пердиши Р., Дагон Д., Ли В., Фамстер Н. Создание динамической системы репутации для DNS. В материалах 19-й конференции USENIX по безопасности, Вашингтон, округ Колумбия, США, 11-13 августа 2010 г.; USENIX Association: Berkeley, CA, USA, 2010; p. 18.
11. Ядав С., Редди А.К.К., Редди А.Н., Ранджан С. Обнаружение алгоритмически сгенерированных вредоносных доменных имен. В материалах 10-й конференции ACM SIGCOMM по измерениям в Интернете, Мельбурн, Австралия, 1-3 ноября 2010 г.; ACM: Нью-Йорк, штат Нью-Йорк, США, 2010; стр. 48-61. [CrossRef].
12. Кришнан, С.; Тейлор, Т.; Монроуз, Ф.; Макхью, Дж. Переступая порог: Обнаружение сетевых злоупотреблений с помощью последовательной проверки гипотез. В материалах 43-й ежегодной международной конференции IEEE/IFIP 2013 по зависимым системам и сетям (DSN), Будапешт, Венгрия, 24-27 июня 2013 г.; стр. 1-12. [CrossRef].
13. Jain, A.K.; Dubes, R.C. *Algorithms for Clustering Data*; Prentice-Hall, Inc: Upper Saddle River, NJ, USA, 1988.
14. Бильге, Л.; Сен, С.; Бальзаротти, Д.; Кирда, Е.; Крюгель, К. Разоблачение: Пассивная служба анализа DNS для обнаружения и сообщения о вредоносных доменах. *ACM Trans. Inf. Syst. Secur.* **2014**, *16*, 14:1-14:28. [CrossRef].
15. Скьявони, С.; Магги, Ф.; Кавалларо, Л.; Занеро, С. Феникс: Отслеживание и разведка ботнетов на основе DGA. *Обнаружение вторжений и вредоносного ПО, а также оценка уязвимости*; Дитрих, С., ред.; Springer International Publishing: Cham, Switzerland, 2014; pp. 192-211.
16. Вудбридж, Дж.; Андерсон, Х.С.; Ахуджа, А.; Грант, Д. Предсказание алгоритмов генерации доменов с помощью сетей кратковременной памяти . *arXiv* **2016**, arXiv:1611.00791.
17. Ю, Б.; Грей, Д.Л.; Пан, Дж.; Кок, М.Д.; Насименто, А.К.А. Обнаружение ДГА в режиме реального времени с помощью глубоких сетей. In Proceedings of the 2017 IEEE International Conference on Data Mining Workshops (ICDMW), New Orleans, LA, USA, 18-21 November 2017; pp. 683-692. [CrossRef].
18. Румельхарт Д.Е., Хинтон Г.Е., Уильямс Р.Дж. Обучение представлений путем обратного распространения ошибок. *Природа*
1986, *323*, 533-536. [CrossRef].
19. Хохрайтер, С.; Шмидхубер, Дж. Длительная кратковременная память. *Neural Comput.* **1997**, *9*, 1735-1780. [CrossRef] [PubMed].
20. Васвани, А.; Шазир, Н.; Пармар, Н.; Ушкорейт, Дж.; Джонс, Л.; Гомес, А.Н.; Кайзер, Л.; Полосухин, И. Внимание Is All You Need. *arXiv* **2017**, arXiv:1706.03762.
21. Грейвс, А.; Уэйн, Г.; Данихелка, И. Нейронные машины Тьюринга. *arXiv* **2014**, arXiv:1410.5401.
22. Луонг, М.Т.; Фам, Х.; Мэннинг, К.Д. Эффективные подходы к нейронному машинному переводу на основе внимания.
Препринт arXiv **2015**, arXiv:1508.04025.
23. Cheng, J.; Li, D.; Lapata, M. Long Short-Term Memory-Networks for Machine Reading. *arXiv* **2016**, arXiv:1601.06733.
24. Бамбенек, Дж. OSINT-каналы от Bambenek Consulting / Дж. Бамбенек // Информационные технологии. 2019. Доступно онлайн: http://osint.bambenekconsulting. com/feeds/ (дата обращения: 10 сентября 2019 г.).
25. Алекс. Исследование ключевых слов, конкурентный анализ и ранжирование сайтов| Alexa. 2019. Available online: https://www.alexa.com/ (accessed on 10 September 2019).
26. Коэн, Р. Банковские трояны: Справочное руководство по семейному древу вредоносных программ. 2019. Available online: https://www.f5.com/labs/articles/education/banking-trojans-a-reference-guide-to-the-malware-family-tree (accessed on 3 September 2019).
27. Кессем, Л. Ботнет Necurs: Ящик Пандоры вредоносного спама. 2017. Доступно онлайн: https: //securityintelligence.com/the-necurs-botnet-a-pandoras-box-of-malicious-spam/ (дата обращения: 3 сентября 2019 г.).
28. Spring, T. Locky Ransomware возвращается к жизни с помощью ботнета Necurs. 2017. Доступно онлайн: https://threatpost. com/locky-ransomware-roars-back-to-life-via-necurs-botnet/125156/ (дата обращения: 3 сентября 2019 г.).