2021 IEEE 5-я Международная конференция по криптографии, безопасности и конфиденциальности

Обнаружение доменов DGA с помощью глубокого обучения

Халех Шахзад
Аналитики по кибербезопасности TELUS
Торонто, Канада
haleh.shahzad@telus.com

Абдул Рахман Саттар
Аналитика кибербезопасности TELUS
Торонто, Канада
Абдул_Rahman.Sattar@telus.com

Джанахан Скандараниям
Аналитики кибербезопасности TELUS
Торонто, Канада
Janahan.Skandaraniyam@telus.com

***Аннотация* - Алгоритмы генерации *доменов* (DGA) используются злоумышленниками для генерации большого количества псевдослучайных доменных имен для подключения к вредоносным командно-контрольным серверам (*C&C*). Эти доменные имена используются для обхода средств обнаружения и снижения уровня безопасности на основе доменов. Обратная разработка образцов вредоносного ПО для обнаружения алгоритма DGA и семян для создания списка доменов - одна из техник, используемых для обнаружения доменов DGA. Впоследствии эти домены предварительно регистрируются и закрываются или публикуются в черных списках устройств безопасности для снижения вредоносной активности. Этот метод требует много времени и может быть легко обойден злоумышленниками и авторами вредоносных программ. Статистический анализ также используется для выявления доменов DGA за определенный промежуток времени, однако многие из этих методов требуют контекстной информации, которую не так просто или невозможно получить. Существующие исследования также продемонстрировали использование традиционных методов машинного обучения для обнаружения доменов DGA. Наша цель состояла в том, чтобы обнаружить домены DGA на основе каждого домена, используя только доменное имя и не имея никакой дополнительной информации. данной работе представлен классификатор DGA, использующий архитектуру на основе рекуррентной нейронной сети (RNN) для обнаружения доменов DGA без необходимости использования контекстной информации или созданных вручную признаков. Мы сравнили производительность различных архитектур на основе RNN, оценив их на наборе данных из 2 миллионов доменных имен. Результаты показали незначительную разницу в показателях производительности между архитектурами RNN.**

***Индексные термины* - *кибербезопасность*, DGA, глубокое обучение, LSTM, Bi-LSTM, GRU**

I. Введение

Злоумышленники могут управлять группой зараженных вредоносным ПО машин, называемых ботнетами [1], удаленно через командно-контрольные (*C&C*) серверы. Вредоносные программы обычно используют алгоритмы генерации доменов (DGA), чтобы оставаться неуловимыми, предотвратить уничтожение *C&C* и продлить срок действия вредоносного ПО. DGA используются для генерации псевдослучайных доменных имен, а C&C регистрирует подмножество этих доменных имен и использует их для связи с вредоносным ПО. Сгенерированная последовательность доменных имен является псевдослучайной из-за начального семени, используемого DGA для их генерации. И вредоносная программа на зараженной машине, и *C&C* используют один и тот же DGA и семя для генерации последовательности доменных имен DGA. Если вредоносная программа пытается установить связь с доменным именем, зарегистрированным *C&C*, связь и, как следствие, вредоносная активность становятся успешными. *C&C* часто перебирают доменные имена DGA, и в итоге вредоносная программа на зараженной машине подключается к C&C-серверу. Этот процесс показан на рис. 1. Для создания DGA могут использоваться различные техники, которые приводят к сложности

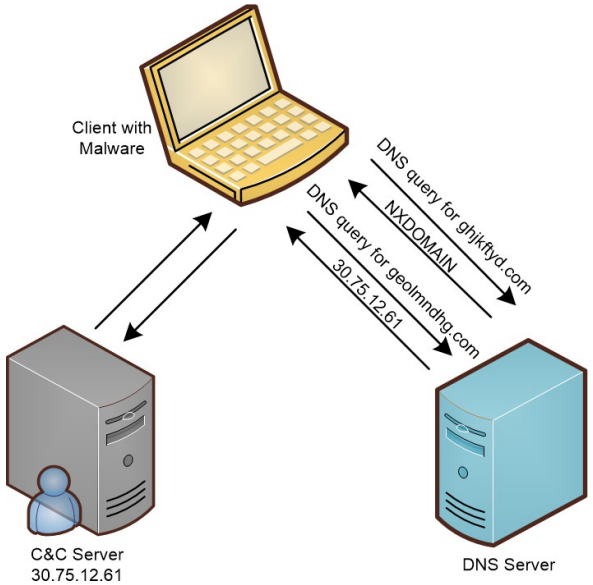

Рис. 1: Типичный пример вредоносного ПО, использующего DGA для подключения к *C&C*

Диапазон генерируемых доменов - от простых, единообразно сгенерированных доменных имен до тех, которые пытаются следовать реальным распределениям доменных имен [2].

В идеале мы должны обнаруживать домены DGA на основе каждого домена с помощью классификатора DGA до начала любой коммуникации C&C. Этот классификатор может находиться в сети и прослушивать DNS-запросы для выявления DGA-доменов. При обнаружении DGA-домена классификатор может уведомить другие автоматизированные инструменты или администраторов сети для дополнительной проверки обнаружения, а все дальнейшие коммуникации с потенциальным DGA-доменом будут заблокированы. Подобное обнаружение в режиме реального времени является значительно более сложной задачей, и зачастую производительность существующих методов слишком мала для применения в реальных условиях.

В данной статье сравнивается эффективность беспризнаковой техники с использованием различных архитектур RNN для классификации DGA. Сравниваются следующие архитектуры RNN: сети с длинной кратковременной памятью (LSTM), бидирективные LSTM (BiLSTM) и GRU. Этот классификатор DGA не имеет признаков и работает с необработанными доменными именами (например, google, yahoo). Если обнаруживается новое семейство доменов DGA, то

2021 IEEE 5th International Conference on Cryptography, Security and Privacy (CSP)| 978-1-7281-8621-4/21/$31.00 ©2021 IEEE| DOI: 10.1109/CSP51677.2021.9357591

классификатор можно переобучать без необходимости ручного извлечения признаков. Наш классификатор работает в основном как "черный ящик", что делает его очень сложным для злоумышленников с целью обхода обнаружения доменов DGA. Этот классификатор DGA также может быть использован в многоклассовой классификации для определения уникального семейства, к которому принадлежит домен DGA, при полном отсутствии контекстной информации.

II. Связанные работы

Одним из подходов к обнаружению DGA-доменов является обратная разработка двоичных файлов вредоносного ПО, о чем говорится в [3]. Как только алгоритм DGA и семя известны, *можно* узнать последовательность доменных имен DGA, к которым будет пытаться подключиться вредоносная программа, и, следовательно, те, которые будут зарегистрированы ее ЦУПом, и предпринять соответствующие действия для смягчения последствий, подрывной деятельности и сбора разведданных. Сетевые администраторы могут вносить генерируемые DGA домены в черный список на устройствах безопасности, чтобы блокировать соединения с потенциальными *C&C* и предотвращать вредоносную сетевую активность. Синхолинг - это техника, которая может быть использована для перенаправления трафика с предполагаемого места назначения на другой сервер. Этот сервер может выступать в качестве самозваного C&C, регистрируя доменные имена DGA. После создания успешной "воронки" активность вредоносного ПО можно контролировать, отслеживать и перехватывать до такой степени, что вредоносное ПО становится бессильным. Например, для продолжения кампании, в которой участвуют ботнеты, использующие DGA, необходимо развернуть новые ботнеты с новыми семенами [2].

В статье [4] обсуждалась эффективность подхода к созданию черных списков, который основан на добавлении доменов, сгенерированных DGA, в черный список, используемый устройствами контроля безопасности для блокирования соединений с *C&C*. Как статическое создание черных списков, так и занесение в черные списки эффективны только в том случае, если известны DGA и семена, используемые кампанией [2]. Кроме того, количество доменных имен, которые могут быть сгенерированы DGA, достаточно велико, поэтому составление черных списков и создание sinkhol всех их, когда на самом деле может использоваться только часть, является неэффективным и утомительным процессом. Кроме того, для случаев, когда семя меняется динамически, этот процесс становится еще более сложным.

В связи с недостатками предыдущих подходов для устранения были предложены подходы машинного обучения с супервизором и без него. В рамках методов машинного обучения были предложены ретроактивные системы обнаружения и системы обнаружения в реальном времени. В работах [5] [6] [7] первая категория хорошо изучена с помощью таких средств, как кластеризация, однако эти подходы имеют недостатки. Эти подходы основаны на анализе партий данных о трафике, что предполагает временную задержку перед обнаружением, а значит, обнаружение произойдет после того, как вредоносная программа свяжется с C&C настолько, что сможет нанести ущерб. Эти методы также используют контекстную информацию, такую как заголовки HTTP, ответы NXDomain и исторические данные о разрешении DNS, чтобы повысить производительность, но сбор этой информации требует больших затрат и иногда не представляется возможным.

Извлеченные вручную признаки, такие как длина строки, соотношение гласных и цифр, распределение вероятностей символов, используются в модели машинного обучения во многих подходах реального времени [8] [9]. Ручное извлечение признаков - трудоемкий процесс. и эти особенности легко обходятся противниками. Злоумышленники могут создать новый DGA, который учитывает набор характеристик классификатора, чтобы обойти обнаружение. Противник, создающий такой DGA, должен иметь доступ к доброкачественной исходной информации, чтобы имитировать особенности доброкачественных доменов при генерации доменов DGA. Злоумышленники также могут использовать реализацию классификатора для тестирования своих DGA-доменов, чтобы убедиться, что они могут эффективно снизить точность классификатора [10]. Если злоумышленник создаст новое семейство DGA с помощью такого процесса, то для борьбы с ним потребуется определить новые характеристики.

В работе [11] для создания 16-битного представления домена использовался автоэнкодер, а затем имя домена классифицировалось с помощью SVM. В [12] для обнаружения доменов DGA предложена двухуровневая архитектура с использованием машинного обучения. Сначала домены классифицируются на классы DGA и не-DGA с помощью алгоритма машинного обучения. Были протестированы семь различных классификаторов машинного обучения, и было заявлено, что классификатор на основе дерева решений имеет наилучшую производительность. Класс доменов DGA затем передается в алгоритм кластеризации DBSCAN для определения семейств, к которым принадлежат домены DGA. Очевидно, что использование любого алгоритма машинного обучения требует эффективного процесса разработки признаков, а сами признаки могут нуждаться в обновлении.

В [13] предложена модель обнаружения сетевых вторжений на основе глубоких сетей убеждений (DBN), которая обучается в три этапа с использованием атрибутивных признаков, представленных в наборе данных KDD CUP 1999. В [14] представлена контекстная глубокая нейронная LSTM, которая учитывает как контекстные признаки, так и метаданные для обнаружения ботов. В работе [2] предложен метод, использующий LSTM на уровне символов с неглубокой архитектурой для предсказания доменных имен DGA. Архитектура состоит из слоя встраивания, одного слоя LSTM и одного слоя логистической регрессии. Для обучения использовались синтетически сгенерированные маркированные данные в виде образцов доменов DGA и не-DGA. В работе [15] исследовалась производительность архитектур CNN и LSTM для обнаружения доменов DGA. Обе модели работали на уровне символов, при этом в качестве входных данных использовалась только строка с названием домена без какой-либо другой контекстной информации. Но в данном случае модели были настроены только на целевой коэффициент ложных срабатываний с использованием неглубокой архитектуры. Они также фильтровали реальный DNS-трафик, чтобы получить образцы доменов DGA и не DGA для обучения.

III. Архитектура модели

В данной работе мы сравнили производительность классификатора DGA, основанного на следующих архитектурах RNN: однонаправленная LSTM-сеть, двунаправленная LSTM (Bi-LSTM) и GRU. Проблема исчезающих градиентов является ключевым мотивом для применения архитектуры Long Short-Term Memory (LSTM) [16], [17], [18], которая состоит из ячеек LSTM с набором ворот для управления потоком информации. Конструкция ячеек LSTM позволяет LSTM изучать долгосрочные зависимости. Вход в однонаправленной LSTM осуществляется из прошлого, поэтому сохраняется только информация из прошлого, в то время как в двунаправленной LSTM вход двусторонний: один из прошлого в будущее, а другой - из будущего в прошлое.

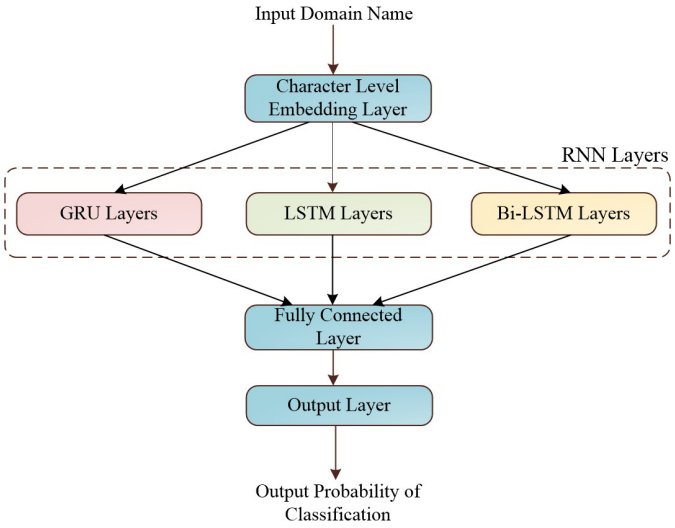

Рис. 2: Архитектура системы

Поэтому информация из будущего может быть сохранена, т.е. исходные данные будут подаваться из конца в начало. GRU очень похож на LSTM, но в нем есть только два гейта: гейт сброса и гейт обновления. Затвор обновления действует аналогично затворам забывания и входа в LSTM. Он решает, какую информацию отбросить, а какую добавить. Ворота сброса используются для принятия решения о том, сколько прошлой информации нужно забыть. При использовании названий доменов в качестве входной последовательности архитектуры на основе RNN узнают важные комбинации букв, которые отличают домены DGA от не-DGA.

На вход нашей модели подается только домен второго уровня (SLD), который находится непосредственно под доменом верхнего уровня (TLD), т.е. на вход подается "google" для "google.com". Модель работает с

В качестве входных данных используются последовательности символов переменной длины, состоящие из слоя встраивания на уровне символов, слоев RNN (LSTM, Bi-
LSTM или GRU слои), полностью связанный слой и, наконец, выходной слой с одним узлом и сигмоидальной активацией; архитектура представлена на рис. 2. Каждый уникальный символ во входном доменном имени сопоставляется с собственным векторным представлением
размерность d в слое встраивания, где d можно настраивать. В качестве входных символов используются нередуцированные строчные буквенно-цифровые символы,
точка, тире и символ подчеркивания. Входными данными для слоя встраивания являются индексы символов входных доменных имен.

Мы использовали библиотеку Torchtext [19] для токенизации входного доменного имени на отдельные символы и присвоения им индексов. Мы добавляем нули в конец строки доменного имени, чтобы все входные данные имели фиксированную длину. Затем мы вставляем входные данные
символы с помощью слоя встраивания, а встраивание изучается в процессе обучения. Слои RNN изучают шаблоны
символы из встроенных векторов, чтобы отличить доменные имена DGA от не-DGA доменных имен. Отсеивание применяется после слоев RNN и перед слоем с полным подключением, а также между слоями RNN для предотвращения избыточной подгонки.

IV. Экспериментальная установка

В следующем разделе мы описываем детали нашей экспериментальной установки для оценки нашего классификатора DGA в бинарном эксперименте (DGA против не-DGA) с использованием общедоступных наборов данных доброкачественных и DGA доменных имен. Мы обучили нашу модель, используя Python 3, Pytorch [20] и станцию NVIDIA DGX (GPU), которая обеспечила вычислительную мощность [21].

A. Показатели производительности

Наш классификатор DGA, использующий LSTM, BiLSTM и GRU, был оценен на популярных наборах данных, которые представлены в разделе IV-B. Производительность модели может быть оценена по определенным метрикам. Матрица путаницы приведена в таблице I, и на ее основе можно рассчитать различные показатели эффективности, такие как *точность*. Количество экземпляров, правильно или неточно предсказанных моделью, может быть представлено в виде матрицы смешения. Матрица путаницы обычно представлена четырьмя значениями, такими как TP, TN, FP и FN.

ТАБЛИЦА I: Матрица путаницы, представленная четырьмя параметрами

		Прогнозируемый	
		Не-DGA	DGA
Фактический	Не-DGA	TN	FP
	DGA	FN	TP

Ниже приведены показатели эффективности, основанные на матрице путаницы. *Точность* - это отношение точно классифицированных DGA и доброкачественных доменных имен ко всему набору данных. *Точность* определяется следующим образом:

$$Точност = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

FPR, False Positive Rate, определяется как доля доброкачественных доменов, ошибочно предсказанных как домены DGA, ко всем доброкачественным доменам:

$$FP = \frac{FP}{FP + TN} \quad (2)$$

TPR, True Positive Rate, также называемый чувствительностью или отзывом, определяется как доля доменов DGA, которые правильно идентифицированы, по отношению ко всем доменам DGA. Он дает вероятность того, что классификатор может правильно предсказать положительные экземпляры (домены DGA).

$$TP = \frac{TP}{TP + FN} \quad (3)$$

Компромисс между TPR и FPR измеряется с помощью ROC-кривой. AUC показывает площадь под ROC-кривой, и чем ближе AUC к 1, тем лучше работает наш классификатор. Для оценки нашей модели используются такие метрики, как *точность*, TPR, FPR и AUC.

B. Данные

Для образцов, не относящихся к DGA, мы использовали домены Alexa Top 1M [22] и список популярности Cisco umbrella [23]. Для DGA-доменов использовался фид OSINT DGA от Bambenek consulting, содержащий различное количество образцов из тридцати семейств DGA [24]. 44 семейства DGA-доменов, представленных в Netlab 360 [25], также использовались в качестве образцов DGA-доменов в наших экспериментах. Источники данных, которые мы использовали, обобщены в таблице II. Количество образцов в исходном наборе данных указано во втором столбце таблицы II. В наших экспериментах мы использовали в качестве входных данных только SLD, т. е. "google" с сайта "google.com". В результате, например, записи "google.com" и "google.ca" в исходном наборе данных в нашем случае оказались дублированными. После дедупликации количество уникальных образцов в каждом наборе данных для наших экспериментов показано в третьем столбце таблицы II.

ТАБЛИЦА II: Описание источников данных

Источники данных	Количество образцов до дедупликации	Количество образцов после дедупликации
Алекса	1,000,000	905,832
Cisco	1,000,000	492,840
OSINT	833,222	787,442
Netlab 360	72,101	69,325

V. Результаты

Для реализации модели был использован Pytorch, предпочтительный отраслевой исследовательский ли- бр для глубокого обучения. Производительность модели оценивалась с помощью набора данных, который был взят из источников данных, представленных в разделе IV-B. Набор данных содержал легитимные образцы из набора данных Alexa [22] и списка популярности Cisco umbrella [23], а также образцы домена DGA из набора данных OSINT [24] и Netlab 360 [25], в общей сложности 2 миллиона с лишним образцов. Мы использовали процентное соотношение 0,8 и 0,1,
0.1 для обучающих, валидационных и тестовых данных, соответственно, в нашем эксперименте.

Был использован встроенный слой встраивания Pytorch, и модель обучалась на партии размером 16, используя архитектуру, описанную в разделе III. После слоя встраивания (первый слой) слой RNN состоит из LSTM, BiLSTM или GRU (несколько ячеек с активацией Tanh по умолчанию), полностью связанного слоя (100 узлов) и, наконец, выходного слоя с одним узлом и сигмоидальной функцией активации. Если вероятность на выходе больше 0,5, доменное имя идентифицируется как DGA, в противном случае - как не-DGA. Мы получили лучшие результаты, используя алгоритм оптимизации RMSprop по сравнению с оптимизатором Адама, а скорость обучения была установлена на уровне 10^{-5}. Мы также применили отсев в 30 % на слое RNN.

Модель обучалась в течение 15 эпох с 3 слоями и 400
клетки. *Точность* 87 % и *TPR* 81 % на тестовом наборе, содержащем 10 % образцов набора данных, были получены для модели на основе LSTM. Архитектура Bi-LSTM достигла *точности* немного меньше, чем модель на основе LSTM. модель, в то время как архитектура GRU не имеет относительно более высокой производительности, хотя и сокращает время обучения за счет более простой архитектуры. ROC-кривая тестового набора для архитектуры на основе LSTM показана на рис. 3. Для архитектуры на основе LSTM достигнуто значение AUC около 0,98, что свидетельствует о хорошей работе классификаторов на более чем 200 000 тестовых образцов.

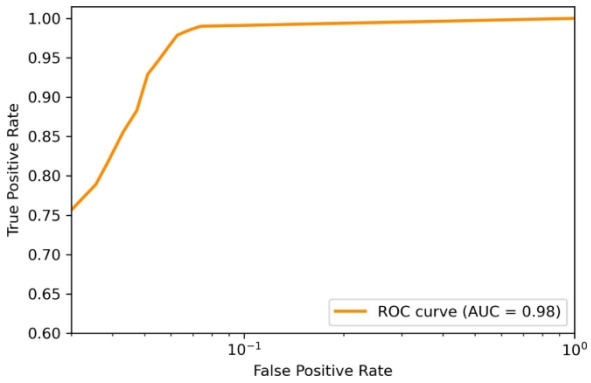

Рис. 3: ROC-кривая для модели на основе LSTM

VI. Заключение

В данной работе представлен подход, использующий архитектуры на основе RNN для обнаружения доменов, генерируемых DGA. Было проведено сравнение производительности следующих архитектур RNN: LSTM, Bi-LSTM и GRU. Этот подход использует только исходные имена доменов в качестве входных данных и поэтому не требует ручного создания признаков, что является громоздким для обслуживания. Классификаторы DGA на основе RNN используют в качестве входных данных только доменные имена из DNS-запросов и поэтому могут быть развернуты в ИТ-среде с минимальными сложностями. Они также обладают высокой масштабируемостью и адаптивностью к изменениям в реальном трафике. Оценка на общедоступных наборах данных показала, что классификаторы DGA на основе RNN показали результаты. Информация о ДВУ также очень ценна для обнаружения DGA и будет включена в нашу будущую работу. Более эффективное обнаружение DGA по словарю с помощью модели на основе RNN также станет предметом нашей дальнейшей работы.

Ссылки

[1] Е. Кук, Ф. Джаханиан и Д. Макферсон, "Зомби-раунд-ап: Понимание, обнаружение и уничтожение бот-сетей", *SRUTI*, том 5, стр. 6-6, 2005.

[2] Дж. Вудбридж, Х. С. Андерсон, А. Ахуджа, и D. Грант, "Предсказание алгоритмов генерации доменов с помощью сетей долговременной кратковременной памяти", *препринт arXiv:1611.00791*, 2016.

[3] В. Стоун-Гросс, М. Кова, Б. Гилберт, Р. Кеммерер, С. Крюгель и Г. Винья, "Анализ захвата ботнета", *IEEE Security & Privacy*, vol. 9, no. 1, pp. 64-72, 2010.

[4] M. Ku¨hrer, C. Rossow, and T. Holz, "Paint it black: Оценка эффективности черных списков вредоносного ПО", *Международный семинар по последним достижениям в области обнаружения вторжений*, с. 1-21, Springer, 2014.

[5] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Абу-Нимех, В. Ли и Д. Дагон, "От выброшенного трафика до ботов: обнаружение роста вредоносного ПО на основе dga", в *докладе на 21-м симпозиуме по безопасности {USENIX} ({USENIX} Security 12)*, стр. 491-506, 2012.

[6] S. Yadav, A. K. K. Reddy, A. N. Reddy и S. Ranjan, "Detecting algorithmically generated malicious domain names," in *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, pp. 48-61, 2010.

[7] S. Yadav, A. K. K. Reddy, A. N. Reddy, and S. Ranjan, "Detecting algorithmically generated domain-flux attacks with dns traffic analysis," *IEEE/Acm Transactions on Networking*, vol. 20, no. 5, pp. 1663-1677, 2012.

[8] S. Скьявони, Ф. Маджи, Л. Кавалларо и С. Занеро, "Phoenix: Dga-based botnet tracking and intelligence," in *International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment*, pp. 192- 211, Springer, 2014.

[9] S. Саад, И. Траоре, А. Горбани, Б. Сайед, Д. Чжао, W. Lu, J. Felix, and P. Hakimian, "Detecting p2p botnets through network behavior analysis and machine learn- ing," in *2011 Ninth annual international conference on privacy, security and trust*, pp. 174-180, IEEE, 2011.

[10] J. Спурен, Д. Превенерс, Л. Десмет, П. Янссен и В. Йосен, "Обнаружение алгоритмически сгенерированных доменных имен, используемых ботнетами: гонка вооружений", в *Трудах 34-го симпозиума ACM/SIGAPP по прикладным вычислениям*, стр. 1916-1923, 2019.

[11] B. Dahal и Y. Kim, "Autoencoded domains with mean activation for dga botnet detection," in *2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3)*, pp. 208-212, IEEE, 2019.

[12] Y. Li, K. Xiong, T. Chin, and C. Hu, "A machine learning framework for domain generation algorithm-based mal- ware detection," *IEEE Access*, vol. 7, pp. 32765-32782, 2019.

[13] N. Gao, L. Gao, Q. Gao, and H. Wang, "An intrusion detection model based on deep belief networks," in *2014 Second International Conference on Advanced Cloud and Big Data*, pp. 247-252, IEEE, 2014.

[14] S. Кудугунта и Э. Феррара, "Глубокие нейронные сети для обнаружения ботов", *Information Sciences*, vol. 467, pp. 312-322, 2018.

[15] B. Yu, D. L. Gray, J. Pan, M. De Cock, and A. C. Nascimento, "Inline dga detection with deep networks," in *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 683-692, IEEE, 2017.

[16] S. Хохрайтер и Й. Шмидхубер, "Длительная кратковременная память", *Нейронные вычисления*, том 9, № 8, стр. 1735-1780, 1997.

[17] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Непрерывное предсказание с помощью lstm", 1999.

[18] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Обучение точной синхронизации с помощью рекуррентных сетей lstm". *Журнал исследований в области машинного обучения*, том 3, нет. Aug, pp. 115-143, 2002.

[19] Torch Contributors , "Пакет Torchtext." https://pytorch. org/text/.

[20] Исследовательская лаборатория искусственного интеллекта Facebook, "Библиотека Pytorch". https:// pytorch.org/text/.

[21] Nvidia, "Nvidia dgx station." https://docs.nvidia.com/ dgx/dgx-station-user-guide/.

[22] Amazon, "Есть ли у alexa список самых рейтинговых сайтов". https://support.alexa.com/hc/en-us/articles/ 200449834.

[23] Cisco, "Cisco зонтик популярность список." http:// s3-us-west-1.amazonaws.com/umbrella-static/index.htm.

[24] Бамбенек Консалтинг, "Осинт бамбенек консалтинг фид." http://osint.bambenekconsulting.com/feeds/.

[25] Исследовательская лаборатория сетевой безопасности 360, "Проект Netlab dga". https://data.netlab.360.com/.