Типичный подход применения переноса обучения заключается в том, чтобы обучить базовую сеть, затем скопировать *n* первых слоев базовой сети и использовать их как *n* первых слоев целевой сети. Оставшиеся слои целевой сети инициализируются случайным образом и дотренировываются на целевой задаче. Так же в зависимости от ситуации можно либо «заморозить» *n* первых слоев в процессе дальнейшего дообучения, либо распространять ошибки обучения и на них. Такой процесс называется*fine-tuning*.

Список использованных источников:
1.VisualizingandUnderstandingConvolutionalNetworks[Электронныйресурс] : CornellUniversity - Электронные данные. – Режимдоступа: https://arxiv.org/abs/1311.2901
2. Deep Learning of Representations for Unsupervised and Transfer Learning [Электронныйресурс] : Bengio et al./Workshop on Unsupervised and Transfer Learning - Электронные данные. – Режимдоступа: http://proceedings.mlr.press/v27/bengio12a/bengio12a.pdf
3.DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition Networks [Электронныйресурс] : UC Donahue et al./Berkeley. – Электронныеданные. – Режимдоступа: http://proceedings.mlr.press/v32/donahue14.pdf

БЛОКЧЕЙН КАК ОСНОВА ЗАЩИЩЁННЫХ РАСПРЕДЕЛЕННЫХ БАЗ ДАННЫХ

УО Белорусский Государственный Университет Информатики и Радиоэлектроники
г. Минск, Республика Беларусь

Глоба А.А.
Ганжа В.А. – к.физ.-мат.н., доцент

В настоящее время в связи с ростом числа пользователей Интернет-сервисов растёт и потребность в принципиально новом способе хранения данных, защищённом от атак и легко масштабируемом по горизонтали. Отличным кандидатом на эту роль может стать технология блокчейн.

Блокчейн в общем виде – это выстроенная в соответствии с определенными правилами цепочка блоков, содержащих какую-либо информацию. Обычно копии цепочек хранятся распределённо на разных устройствах в пределах одной сети. Для записи нового блока, необходимо последовательное считывание информации о старых блоках, при этом новый блок добавляется, если большая часть узлов в сети подтвердит, что в нем не нарушаются какие-либо заранее определенные правила. Все данные в блокчейн накапливаются и формируют постоянно дополняемую базу данных. С этой базы данных невозможно ничего удалить или провести замену/подмену блока, доступно только чтение.

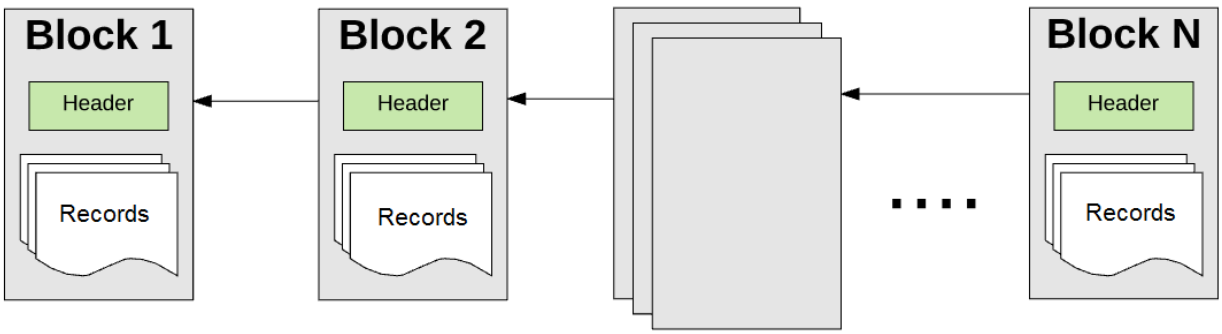

Рис. 1 – Схема блоков в БД

Главные особенности технологии блокчейн:

1. Распределенность данных: нет одного сервера, хранящего информацию. При традиционном подходе в случае какой-либо атаки на сервер или отказа оборудования по какой-либо причине теряется доступ к данным и к сервису. В случае технологии блокчейн данные распределены в виде локальных копий на разных узлах – система будет сохранять работоспособность до тех пор, пока хотя бы один узел остается в сети.
2. Невозможность изменения уже существующих данных: в случае с обычными БД всегда есть возможность преднамеренно или случайно изменить существующие записи. Сфальсифицировать новую запись можно только в том случае, если взять под контроль более 50% (если не все) узлы сети, что физически практически невозможно.

Существуют и отрицательные черты БД на основе блокчейн:

1. Низкая скорость добавления новой записи в цепочку: механизм устеновления консенсуса предполагает участие всех (или большинства) узлов в сети для установления соответствия нового блока правилам системы, что вызывает определенную задержку;
2. Необходимость хранения локальных копий БД: в некоторых случаях БД может разрастаться до

больших размеров. Например, цепочка Биткоина уже занимает размер в 100Gb, что является внушительным числом для среднестатистического ПК.

При этом нет никаких ограничений на формат хранимых данных. К примеру, Ethereum позволяет хранить в блокчейне не только транзакции, но и полноценные Тьюринг-полные программы, называемые смарт-контрактами, которые позволяют очень тонко настроить блокчейн на прикладную задачу, например, распределенный DNS-сервер.

Использованныеисточники:
1. Melanie Swan, Blockchain: Blueprint for a New Economy.
2. Jacob William, Blockchain: The Simple Guide To Everything You Need To Know.

ИЗВЛЕЧЕНИЕ ХАРАКТЕРНЫХ ПРИЗНАКОВ В ИСПОЛНЯЕМЫХ ФАЙЛАХ ДЛЯ ОБУЧЕНИЯ НЕЙРОННЫХ СЕТЕЙ ДЛЯ ОБНАРУЖЕНИЯ ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

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

Гороховик Я.В

Воронов А.А. – к.т.н., доцент

Вредоносное программное обеспечение в общем случае представляется бинарными исполняемыми файлами, которые, как правило, регистрируются в системе, распространяются по ней, копируя себя, и оказывают на систему вредоносное воздействие. Современные антивирусные системы определяют вредоносное ПО, обладая знаниями о различных шаблонах поведения вирусов, однако определение новых, ранее не выявленных угроз, представляет определённую сложность для них. Огромное количество эвристических методов, используемых антивирусными решениями, потребляют значительное количество памяти и других ресурсов процессора. Эта нагрузка может быть преодолена путём обучения искусственных нейронных сетей, обученных на характерных признаках вредоносного ПО, содержащихся в самих исполняемых файлах. Исполняемые файлы формата PortableExecutable (PE) содержат множество полей, которые могут использоваться для предсказания поведения программы. Подход к сбору этих признаков из PE-файлов описан в данной работе.

В настоящее время одной из главных проблем информационной безопасности является выявление нового вредоносного ПО и новых угроз. Известные вирусы не представляют особой угрозы, так как могут быть легко детектированы сигнатурным анализом. Однако, выявление новых угроз требует намного более сложных эвристических подходов. Имеются следующие методы выявления таких угроз:

1. нахождение сходств между семействами вирусов. Как правило, данный метод основывается на различных алгоритмах машинного обучения, таких как Байесовская сеть или генетических алгоритмы;
2. реализация алгоритмов, эмулирующих методы принятия решений аналитика-человека – фактически, создание экспертной системы;
3. анализ файла в «песочнице». Для этого необходимо реализовать перехваты важных функций режимов пользователя и ядра. Этот метод предполагает изучения реального поведения файла в виртуальной среде[1].

Однако, каждый из перечисленных методов имеет свои ограничения:

1. первый метод может стоить значительного количества ошибок первого рода и занимать ресурсы ЭВМ, что приемлемо для тестовой среды, но не для домашнего компьютера;
2. второй метод наиболее удачен, так как предполагает изучение действительного поведения, как бы его проводил аналитик, однако, данный метод сложен и требует много памяти и ресурсов ЦП;
3. третий метод в значительной степени зависит от качества реализации соответствующего эмулятора ЦП и перехватов системных вызовов в драйверах и библиотеках. Он эффективен, однако стоит много ресурсов и денег.

В данной работе предлагается подход к извлечению характерных особенностей из исполняемых файлов для создания и обучения системы, которая принимает большое количество признаков PE-файла для определения его легитимности.

Предлагаемый метод включает сбор характерных признаков исполняемого файла следующими способами:

1. просмотр таблицы импорта для нахождения системных вызовов, наиболее часто используемых вредоносным ПО (AdjustTokenPrivileges, CreateRemoteThread, GetProcAddress, VirtualProtectEx, WriteProcessMemory и т.д.);
2. просмотр структуры PE в целом для нахождения наиболее важных полей.

Для получения особенностей таблицы импорта исполняемого файла создаётся список наиболее часто используемых вредоносным ПО функций в алфавитном порядке. Если в таблице импорта встречается такая функция, она кодируется единицей, если нет, то нулём. Полученная строка из единиц и нулей принимается за