СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Некоторые разновидности вредоносных программ используют алгоритмы генерирования доменных имен для определения адресов управляющих серверов. Подобные алгоритмы позволяют защитить вредоносные сервера от однократного отключения или добавления адресов в черные списки. Чаще всего данные алгоритмы используются в крупных ботнетах.

Ботнет - некоторая сеть, в том числе компьютерная, состоящая из устройств (ботов), со специально запущенным вредоносным программных обеспечением. Чаще всего боты инфицируются посредством вредоносного программного обеспечения, полученного из сети Интернет. Однако путём инфицирования может служить также локальная сеть или устройства ввода, например флэш накопители. Ботнеты являются наиболее распространенным средством кибер атак. Они управляются его создателем при помощи специальных управляющих командных серверов (Command and Control Servers). Большинство из них используются для монетизации различными способами, такими как: распределенные атаки отказ в обслуживании (DDoS атаки), продажа Drive By Download, атаки на клиентов дистанционного банковского обслуживания, для спама и проведения фишинговых атак. Эти и другие угрозы ботсетей выделены в статье [10].

Для удержания контроля над ботами и их управления Ботнеты используют множество способов. Это может быть одноранговые сети, почтовые протоколы, социальные сети или анонимные сети, такие как TOR или i2p. Однако самым распространенным на данный момент является Алгоритмы Генерации Доменных Имен (Domain Generation Algotithms).

Они позволяют удерживать контроль над управляющими серверами. В основном подобные алгоритмы используются в крупных ботсетях. Например, одним из первых случаев был компьютерный червь Conficker в 2008 году. На сегодняшний день подобных вредоносных программ насчитываются десятки, каждая из которых представляет серьезную угрозу. Помимо этого, алгоритмы

совершенствуются, их обнаружение становится сложнее. Например, осенью 2014 года была обнаружена новая версия ботнета Matsnu, в которой для генерации доменов используются существительные и глаголы из встроенного списка. Подробнее историю развития алгоритмов генерации доменных имен рассмотрена в статье [6].

Целью данной работы является разработка модели на основе методов машинного обучения для распознавания и классификации вредоносных доменных имен, полученных при помощи анализа алгоритмов генерации доменных имен, а также получение количественной оценки качества классификации на сформированной выборке доменных имен.

Проблемы автоматического анализа алгоритмов DGA и пути их решения можно найти в статье [1]. Идея использования методов машинного обучения освещена в работе [7]. Так, ряд известных компаний, занимающихся информационной безопасностью (Damballa, OpenDns, Click Security и др.), применяют подобные решения для анализа и фильтрации сетевой активности вредоносных программ. Например, Click Security в своей работе [3] предлагают использовать решающие деревья для бинарной классификации на принадлежность доменов к вредоносным. Для этого ими предложен способ выделения признаков из домена. Стоит отметить работу [4], которая рассматривает возможность классификации, используя метод опорных векторов (Support Vector Machine) и выделения из доменов признаков n-gram - подстрока, состоящая из последовательных n символов исходной строки. Схожий подход описывается и в работе [5]. Однако, в отличие от работы [4], имеет большую практическую направленность и предлагает использование алгоритма C4.5 (алгоритм для построения деревьев решений). Подход, основанный на анализе морфем в статье [8], является неактуальным, так как последние исследования [6] показывают, что алгоритмы генерации доменных имен совершенствуются с целью обхода существующих способов обнаружения.Поэтому рассмотрение алгоритмов машинного обучения для предотвращения современных угроз является актуальной проблемой информационной безопасности.

1 АЛГОРИТМЫ ГЕНЕРАЦИИ ДОМЕННЫХ ИМЕН

Алгоритмы Генерации Доменных Имен (DGA) представляют собой алгоритмы, используемые вредоносным программным обеспечением (malware) для генерации большого количества псевдослучайных доменных имен, которые позволят им установить соединение с управляющим командным центром. Тем самым они обеспечивают мощный слой защиты инфраструктуры для вредоносных программ. С первого взгляда концепция создания большого количества доменных имен для установки связи кажется не сложной, но методы, используемые для создания произвольных строк часто скрываются за разными слоями обфускации. Это делается для усложнения процесса обратной разработки и получения модели функционирования того или иного семейства алгоритмов. Рассмотрим общие принципы работы таких алгоритмов.

1.1 Общие принципы работы

Общий принцип работы представлен на рис 1.1. В общем случае вредоносному файлу необходим какой-либо параметр для инициализации Генератора Псевдослучайных Чисел (ГСПЧ). В качестве этого параметра может выступать любой параметр, который будет известен вредоносному файлу и владельцу ботнета. В нашем случае - это значение текущей даты и времени. Вредоносный файл, используя протокол HTTP посылает запрос на сайт cnn.com. В ответ на этот запрос cnn.com возвращает в заголовках HTTP ответа текущие время и дату в формате GMT. Владелец ботнета таким же способом получает текущее время и дату в формате GMT. Далее, это значение, попадает в сам алгоритма генерации доменных имен, инициализируя ГСПЧ, который может иметь вид например Линейного конгруэнтного генератора или Регистра сдвига с обратной связью. Поэтому, используя одинаковые вектора инициализации, вредоносный файл и владелец ботнета получают идентичные таблицы доменных имен. После этого владельцу ботнета достаточно зарегистрировать лишь один домен, для того, чтобы вредоносный файл, рекурсивно посылая запросы к DNS серверу

получил IP адрес управляющего сервера для дальнейшей установки с ним соединения и получения, выполнения команд.

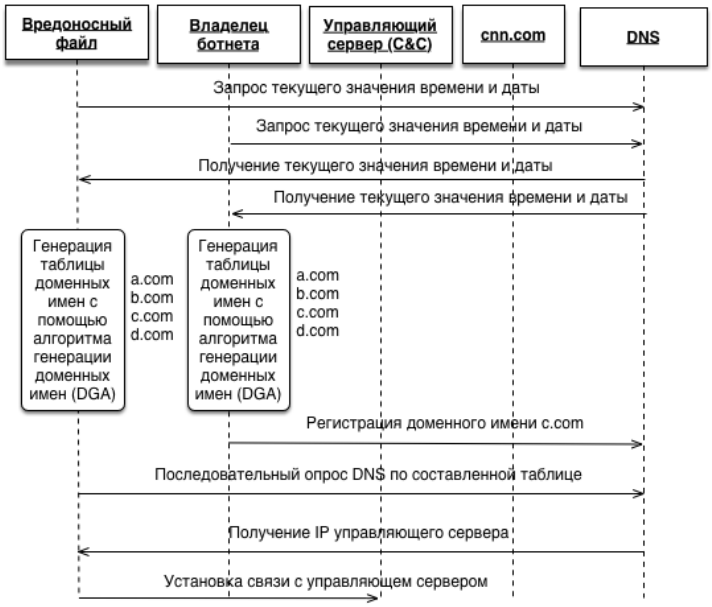

Рис. 1.1 – Общий принцип работы

1.2 Рассмотренные алгоритмы

В ходе выполнения данной работы были проанализированы 8 разновидностей алгоритмов генерации доменных имен, а именно: Conficker, Cryptolocker, Ramdo, PushDo, Zeus, Tinba, Rovnix, Matsnu. Для каждого из них, путём обратной разработки были составлены модели работы алгоритмов генерации доменных имен и реализованы на языках программирования высокого уровня. Далее в работе представлены описание и особенности работы каждого из этих алгоритмов.

Conficker - вирус, впервые появившийся в 2008 году, использующий для заражения машин популярную уязвимость MS08-067. Одним из первых применил технику DGA. Процесс генерации доменного имени можно описать пятью шагами, как показано на рис 1.2 и его архитектура идентична принципу, описанному в части 1.1.

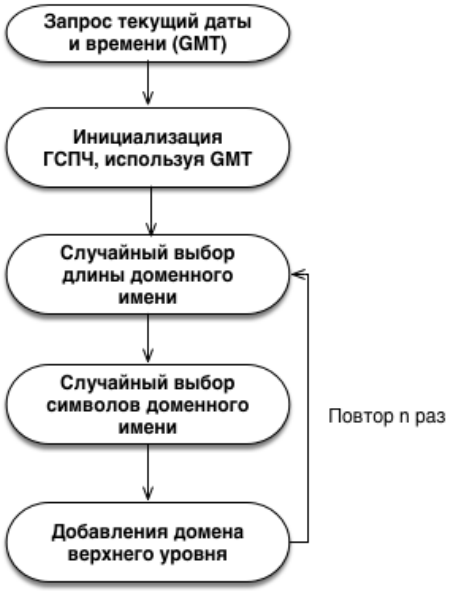

Рис. 1.2 – Принцип работы conficker DGA

Cryptolocker - имя вредоносных программ вида троян-вымогатель (ransomware). Целью данного вируса являются системы Microsoft Windows. Cryptolocker полностью шифрует содержимое файловой системы жертвы и требует заплатить выкуп для получения ключа дешифрования. DGA, который использует cryptolocker относительно прост, однако использует множество приемов, которые осложняют процесс его обратной разработки. Его алгоритм использует для инициализации 4 значения - ключ, день, месяц, год. Ключ может быть константой или рассчитываться по формуле:

```
key = (((key * 0x10624DD3) >> 6) * 0xFFFFFC18)+ key)
```

В данной работе за значение ключа взята константа 0x41. Дата, месяц, год инициализируются соответственно формулам:

```
date = ((date<<13 & 0xFFFFFFFF)>>19 & 0xFFFFFFFF) ^ ((date>>1 & 0
   xFFFFFFFF)<<13 & 0xFFFFFFFF) ^ (date>>19 & 0xFFFFFFFF);
month = ((month<<2 & 0xFFFFFFFF)>>25 & 0xFFFFFFFF) ^ ((month>>3 & 0
   xFFFFFFFF)<<7 & 0xFFFFFFFF) ^ (month>>25 & 0xFFFFFFFF);
year = ((year<<3 & 0xFFFFFFFF)>>11 & 0xFFFFFFFF) ^ ((year>>4 & 0
   xFFFFFFFF)<<21 & 0xFFFFFFFF) ^ (year>>11 & 0xFFFFFFFF);
```

Каждый символ рассчитывается по формуле:

```
chr(ord('a') + (year ^ month ^ date) % 25);
```

А длина составляет доменного имени составляет:

```
date>>3 ^ year>>8 ^ year>>11 & 3 + 12
```

В конце генерации доменного имени добавляется домен верхнего уровня, который последовательно выбирается из массива зашитых значений.

PushDo - второй по величине ботнет, впервые появившийся в 2007 году. Для установки связи с командным сервером использует два механизма. Первый - зашитые доменные имена, и второй - DGA. Второй механизм имеет четыре составляющие, а именно - системное время, вектор инициализации, функцию хеширования MD5 и сам генератор доменных имен. Каждый день, при генерации первого доменного имени DGA вектору инициализации присваивается значение системного времени, которое получается путем вызова WINAPI функции GetLocalTime, операций битового сдвига и XOR - функция шифрования. После этого вектор инициализации попадает на вход функции хеширования MD5, вывод которой в свою очередь подает в генератор доменного имени и вектор инициализации генерации следующего доменного имени. Однако вектор инициализации следующего доменного имени берет не весь вывод функции MD5, а лишь первые её 4 байта. Схема работы этого алгоритма генерации доменных имен представлена на рис 1.3.

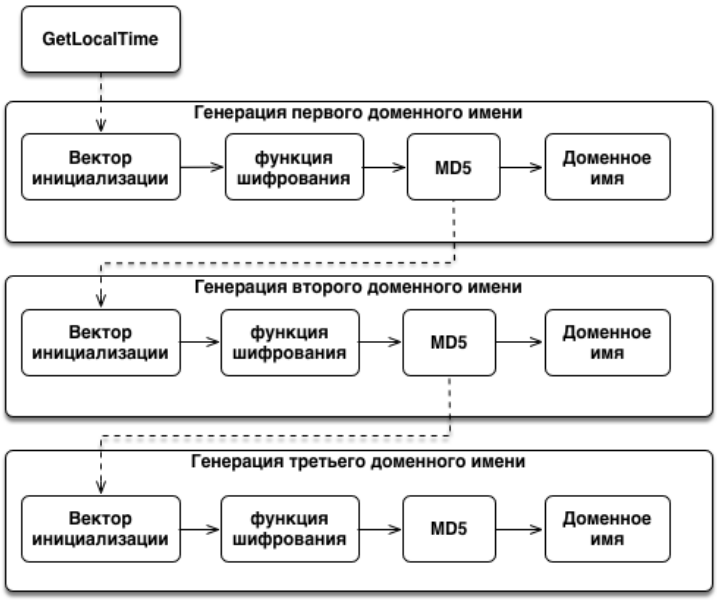

Рис. 1.3 – Принцип работы PushDo DGA

Отдельно стоит рассмотреть генератор доменного имени из вывода функции MD5. Каждый экземпляр такого генератора имеет четыре уникальных строки. Это позволяет обновлять генератор с каждой новой версий ботнета или иметь множество непересекающихся ботнетов одного и того же типа. Сначала генератор вычисляет длину доменного имени. Она рассчитывается делением первых четырех байт на 4 и лежит в пределах от 9 до 12. Далее специальный цикл, используя остаток от деления переводит вывод функции MD5 в читаемое доменное имя и добавляет к нему домен верхнего уровня. В итоге, всего, каждый день PushDo DGA генерирует 30 доменных домен, зависящих не только от системного времени, но и от зашитых значений специальных строк генератора.

Zeus - семейство вредоносных программ, ворующих аутентифицирующие данные, которое впервые появилось в 2007 году. Первые два варианта этого ботнета базировались на зашитых централизованных адресах управляющих серверов. Эти сервера постоянно отслеживались и отключались при помощи антивирусных компаний и центров реагирования на инциденты информационной безопасности. Именно поэтому, в 2011 году было обнаружено появление новой версии этого ботнета, который использует одноранговые сети и DGA для защиты своей инфраструктуры. В первую очередь защита построена на одноранговых сетях, однако если все зараженные машины не отвечают, то используется DGA. Схема его реализация схожа с алгоритмом, используемом в ботнете PushDo. Алгоритм также использует системное время, алгоритм хеширования MD5 и генератор доменного имени. Однако в отличии от PushDo системное время является вектором инициализации для всех 1000 доменных имен генерируемых каждую неделю, а номер доменного имени является параметром salt (соль) в функции хеширования MD5. Генератор доменного имени можно представить в виде следующего псевдокода.

```
hash = MD5(time+salt)
name = ""
for (j = 0; j < len(hash); j++) {
    c1 = (hash[j] & 0x1F) + ''a;
    c2 = (hash[j] / 8) + ''a;
    if(c1 != c2 && c1 <= ''z) name += c1;
```

```
    if(c1 != c2 && c2 <= ''z) name += c2;
}
```

Семейство Ramdo, впервые обнаруженное в декабре 2013 года кардинально отличается от других ботнетов. Ramdo используют множество приемов антиотладки, а также использует технологию double flux (рис **??**) для установки соединения с управляющими серверами. Несмотря на сложность обратной разработки данный ботнет имеет достаточно простой и небольшой DGA, который основывается на зашитой константе и итераторе, основанного на регистре сдвига и булевой операции исключающее или. Для формализации модели DGA данного семейства в работе использован исходный код из работы [15]. Кроме этого идентичную модель DGA использует и Tinba, более известный как Tiny Banker. Основное отличие лишь в том, что для генерации следующего доменного имени Tinba использует также зашитый первоначальный адрес управляющего сервера.

Бурное развитие методов идентификации и фильтрации вредоносных доменных имен привело к развитию новых видов DGA. Такими примерами являются алгоритмы генерации доменных имен ботнетов Rovnix и Matsnu. Их алгоритмы проектировались специально для обхода систем идентификации, построенных на количественной оценки энтропии доменного имени или лингвистических моделях, например n-gramm. Rovnix первым из ботнетов предложил использовать специально заготовленный текст для генерации доменного имени. В качестве такого текста Rovnix использует Декларацию о независимости США. На основе системного времени Rovnix выбирает слова из данного текста и конкатенирует их, для достижения определенной длины. Ботнет Matsnu имеет схожий подход, вредоносная программа использует два зашитых словаря, один из которых содержит существительные, а второй глаголы. Тем самым, основываясь на системном времени алгоритм генерации доменного имени выбирает существительное, а затем выбирает глаголы, для достижения необходимой длины доменного имени. Стоит отметить, что подходы этих двух ботнетов действительно с легкостью обходят системы идентификации, основанные лишь на энтропийной оценке, поэтому для идентификации этих классов

вредоносных доменных имен требуются более сложные модели, описанные в главах 4.1 и 4.2.

2 СУЩЕСТВУЮЩИЕ ПОДХОДЫ К КЛАССИФИКАЦИИ

В данной главе рассматриваются теоретические основы существующих подходов классификации вредоносных доменных имен. Их реализация и результаты их работы представлены в части 4.1.

2.1 Naive Bayes

2.2 Logistic Regression

2.3 Random Forest

2.4 Extra Tree Forest

2.5 Voting Classification

3 НЕЙРОННЫЕ СЕТИ

3.1 Рекуррентные нейронные сети

3.2 LSTM

3.3 Biderectial LSTM

3.4 Механизм внимания

4 ЭКСПЕРИМЕНТ

4.1 Существующие подходы

4.2 Результаты LSTM

4.3 Сравнительный анализ

ЗАКЛЮЧЕНИЕ

Выводы, значение полученных результатов Рекомендации по применению

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. T. Barabosch, A. Wichmann, F. Leder, and E. Gerhards-Padilla Automatic extraction of domain name generation algorithms from current malware.// STO-MP-IST-111 Information Assurance and Cyber Defence, 2012.

2. P. Barthakur, M. Dahal, and M. K. Ghose. An efficient machine learning based classification scheme for detecting distributed command & control traffic of p2p botnets.// 5(10):9, 2013.

3. Click Security. Exercise to detect algorithmically generated domain names.// 2014.

4. N. Davuth and S.-R. Kim. Classification of malicious domain names using support vector machine and bigram method.// 7(1):51–58, January 2013.

5. J. Jacobs. Building a dga classifier. [Электронный ресурс] // URL: http://datadrivensecurity.info/blog/posts/2014/Oct/dga-part3/, October 2014 (дата обращения:)

6. Raff. generation Dgas: A evolution. [Электронный ресурс] // URL: http://www.seculert.com/blog/2014/11/dgas-a-domain-generation-evolution, November 2014. (дата обращения:)

7. M. Stevanovic and J. Pedersen. Machine learning for identifying botnet network traffic //, 2013.

8. Z. Wei-wei, G. Jian, and L. Qian. Detecting machine generated domain names based on morpheme features.// pages 408–411, october 2013.

9. 9. S. Yadav, A. K. K. Reddy, A. L. N. Reddy, and S. Ranjan. Detecting algorithmically generated domain-flux attacks with dns traffic analysis.// 20(5):1663–1677, Oct. 2012.

10. Н. О. Гончаров. Современные угрозы ботсетей.// 10, октябрь 2014.

11. Machine Learning Text feature extraction (tf-idf) [Электронный ресурс] // URL: http://blog.christianperone.com/?p=1589 (дата обращения:)

12. Understanding LSTM Networks [Электронный ресурс] // URL: http://colah.github.io/posts/2015-08-Understanding-LSTMs/ (дата обращения:)

13. Chunting Zhou, Chonglin Sun, Zhiyuan Liu, Francis C.M. Lau A C-LSTM Neural Network for Text Classification // 30 Nov 2015

14. Fidelis Threat Advisory 1016// Pushdo It To Me One More Time// 16 Apr 2015