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.

Ботнеты - 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 p2p 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.

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 (SVM) и выделения из доменов 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. 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. Рассмотрим общие принципы работы таких алгоритмов.

1.1 Общие принципы работы

Общий принцип работы представлен на рис 1.1. В общем случае вредосному файлы необходим какой-либо параметр для инициализации Генератора Пседослучайных 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, инициалиируя ГСПЧ, который может иметь вид Линейного конгруэтного генератора. 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.

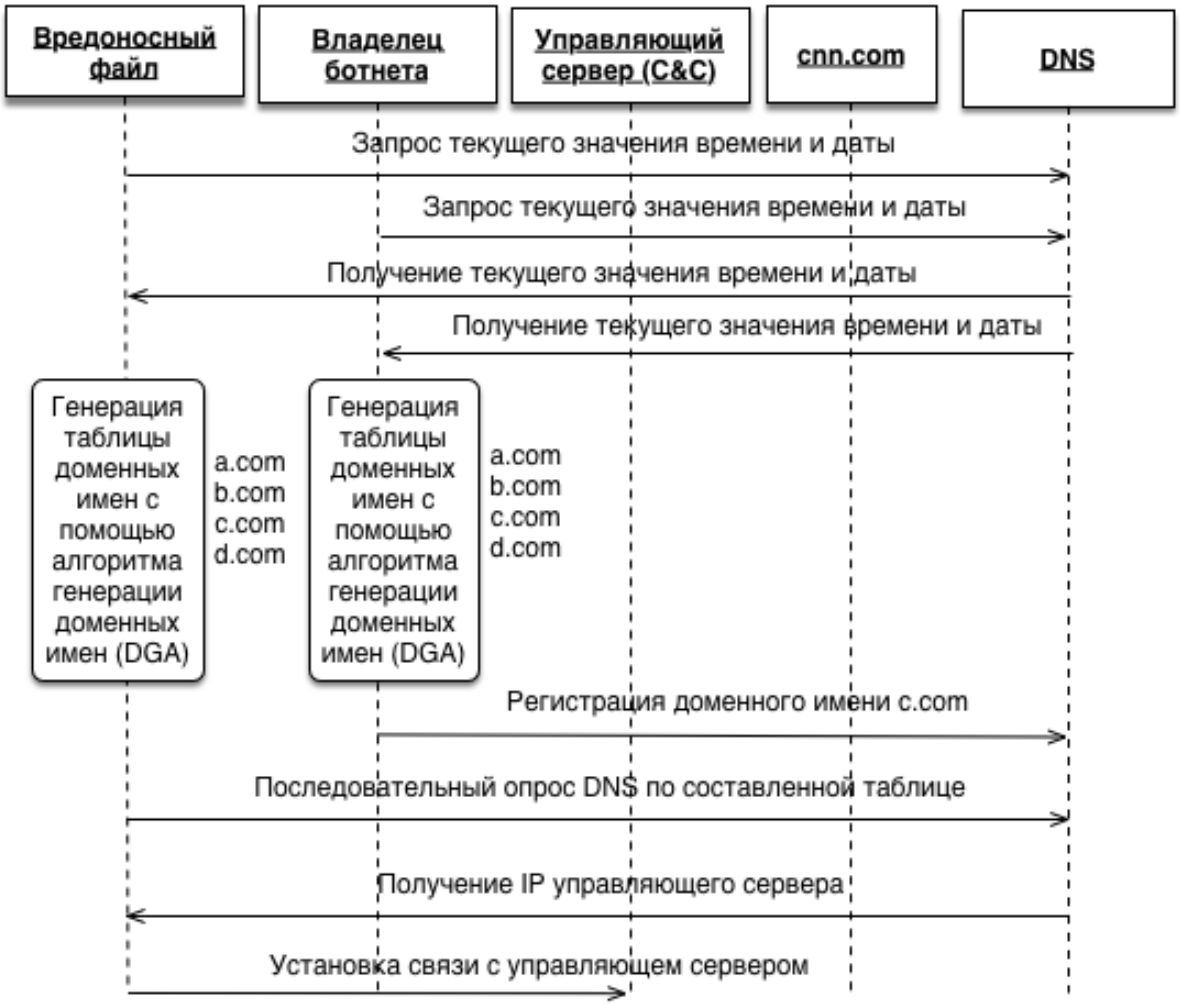

Рис. 1.1 – Общий принцип работы

1.2 Расмотренные алгоритмы

В ходе работы были проанализированы 8 разновидностей алгоритмов генерации доменных имен, а именно:

- Conficker
- Cryptolocker
- Zeus
- PushDo
- Rovnix
- Tinba

– Matsnu
– Ramdo

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 предствлены особенности работы каждого из этих алгоритмов.

Conficker - вирус, впервые появивший в 2088 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 соответствует самому простому случаю, описанному в главе 1.1.

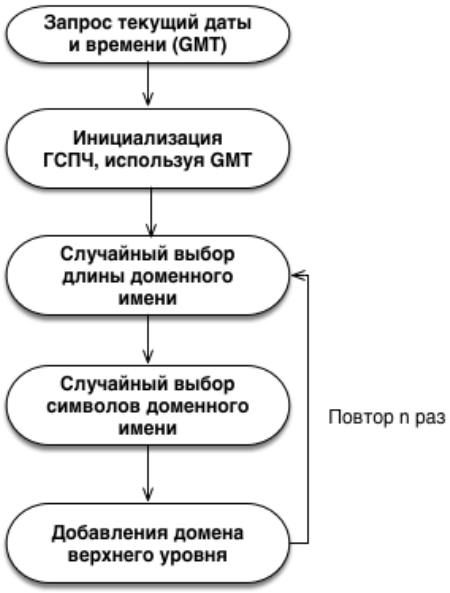

Рис. 1.2 – Принцип работы conficker DGA

Cryptolocker - имя вредоносных программ вида троян-вымогатель (ransomware). Целью данного вируса являются системы Microsoft Windows. Cryptolocker польностью 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 использует 4 значения - ключ, день, месяц, год. Ключ может быть константой и расчитываться 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);
date &= 0xFFFFFFFF;
month = ((month<<2 & 0xFFFFFFFF)>>25 & 0xFFFFFFFF) ^ ((month>>3 & 0
   xFFFFFFFF)<<7 & 0xFFFFFFFF) ^ (month>>25 & 0xFFFFFFFF);
month &= 0xFFFFFFFF;
year = ((year<<3 & 0xFFFFFFFF)>>11 & 0xFFFFFFFF) ^ ((year>>4 & 0
   xFFFFFFFF)<<21 & 0xFFFFFFFF) ^ (year>>11 & 0xFFFFFFFF);
year &= 0xFFFFFFFF;
```

Каждый символ рассчитывается по формуле

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

и его длина составляет

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

В конце добавляется домен верхнего уровня, который последовательно выбирается из массива

```
["com", "net", "biz", "ru", "org", "co.uk", "info"];
```

Zeus PushDo Rovnix Tinba Matsnu Ramdo

2 СУЩЕСТВУЮЩИЕ ПОДХОДЫ К КЛАССИФИКАЦИИ

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 Реккурентные 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