УДК: 004.056.57

Александр И. Мухамедов
Национальный исследовательский центр «Курчатовский институт»,
пл. Академика Курчатова, 1, Москва, 123182, Россия
e-mail: mukhamedov.sa@gmail.com, https://orcid.org/0000-0003-1414-1675

ИССЛЕДОВАНИЕ СОБСТВЕННЫХ СЕТЕВЫХ ПРОТОКОЛОВ ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

DOI: http://dx.doi.org/10.26583/bit.2023.2.05

Аннотация. Подавляющее большинство вредоносных программ осуществляет сетевое взаимодействие со своими компонентами, при котором может происходить скачивание необходимых модулей, передача украденной информации и т.д. Часто такое взаимодействие базируется на самых распространенных протоколах – HTTP и HTTPS, но при этом имеется достаточно много фактов реализации создателями вредоносного ПО собственных протоколов. Целью данной статьи является обзор подходов к реализации собственных протоколов, существующих в среде разработчиков вредоносного ПО, и методов обнаружения фактов использования подобных протоколов в сетевом трафике, выявление перспектив использования тех или иных подходов на практике в настоящее время. Анализируются возможные причины, которые побуждают создателей вредоносных программ поступать таким образом. Проводится обзор различных вариантов подобных протоколов, от примитивных до весьма сложных, приводятся конкретные примеры. В статье приведены способы выявления подобной активности в сетевом трафике: сигнатурный анализ, методы на основе статистики и машинного обучения, интеллектуальный анализ. Рассмотрены основные достоинства и недостатки данных методов. Результатом анализа данных методов является вывод, согласно которому наиболее перспективным подходом в рамках данной проблемы является применение нейронных сетей, в частности, наибольшей популярностью сейчас пользуются решения на основе архитектур CNN и LSTM, однако имеются и альтернативные подходы, которые сейчас не показывают выдающихся результатов, однако могут стать хорошими вариантами для использования в ближайшем будущем.
Ключевые слова: вредоносное ПО, сетевые протоколы, шифрование, обнаружение, нейронные сети.
Для цитирования: МУХАМЕДОВ, Александр И. ИССЛЕДОВАНИЕ СОБСТВЕННЫХ СЕТЕВЫХ ПРОТОКОЛОВ ВРЕДОНОСНОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ. Безопасность информационных технологий, [S.l.], т. 30, № 2, с. 80–88, 2023. ISSN 2074-7136. URL: https://bit.spels.ru/index.php/bit/article/view/149. DOI: http://dx.doi.org/10.26583/bit.2023.2.05.

Alexander I. Mukhamedov
National Research Centre "Kurchatov Institute",
Akademika Kurchatov Square, 1, Moscow, 123182, Russia
e-mail: mukhamedov.sa@gmail.com, https://orcid.org/0000-0003-1414-1675

Approaches to detection of malware using custom network protocols

DOI: http://dx.doi.org/10.26583/bit.2023.2.05

Abstract. The majority of malware communicate with their components over the network. The purpose of this interaction can be downloading the necessary modules, transferring stolen information, etc. Often such interaction is based on the most common protocols - HTTP and HTTPS, but there are quite a lot of facts that malware creators implement their own protocols. The purpose of this study is to analyze approaches to implementing custom protocols that exist among malware developers and methods for its detecting in network traffic. Possible reasons that encourage malware creators to do this are described. A brief overview of the variants of such protocols, from primitive to very complex, is given, including examples. The paper also describes some approaches for detection such activity in network traffic: signature analysis, methods based on statistics and machine learning, neural networks. The main

advantages and disadvantages of these methods are considered. The result of the analysis of these methods is the conclusion that the most promising approach to this problem is the use of neural networks, but when implementing neural networks in practice, certain difficulties may arise associated with their training and further support, which most likely will need to be overcome on their own.
Keywords: malware, network protocols, encryption, detection, neural networks.
For citation: MUKHAMEDOV, Alexander I. Approaches to detection of malware using custom network protocols. IT Security (Russia), [S.l.], v. 30, no. 2, p. 80–88, 2023. ISSN 2074-7136. URL: https://bit.spels.ru/index.php/bit/article/view/1495.DOI: http://dx.doi.org/10.26583/bit.2023.2.05.

Введение

Сетевого взаимодействия вредоносному программному обеспечению (ВПО) избежать практически невозможно. Попав на целевой компьютер, ВПО не знает ни характеристики ресурса, ни геолокацию, у него может отсутствовать большинство модулей, необходимых для работы, и т.д. От того насколько удачно сетевой обмен ВПО будет скрыт от систем обнаружения атак зависит его выживаемость, и, в конечном итоге, заработки злоумышленников, поэтому во многих случаях оно или маскирует сетевой обмен под стандартную активность веб-браузеров, или пытается обеспечить передачу зашифрованной информации с использованием как существующих решений, так и самостоятельно разрабатываемых протоколов.

Из стандартных решений, в первую очередь, следует выделить протокол защиты транспортного уровня TLS. Существует много готовых библиотек, на основе которых можно реализовать протокол TLS, объем TLS трафика в сетевых каналах очень велик. В качестве причин реализации злоумышленниками самодельных протоколов шифрования можно выделить следующие:

- Сложность реализации корректного с точки зрения требований (RFC) и принятых практик TLS взаимодействия. Отклонения от «правильной» реализации могут быть выявлены средствами обнаружения.
- Большой объем кода, который должен быть скачан на целевой компьютер для работы TLS модуля.
- Необходимость генерации сертификата сервера и его подписи известным Удостоверяющим центром (CA). Самоподписанные сертификаты легко выявляются и служат хорошим индикатором подозрительного взаимодействия на предмет наличия в них каких-либо подозрительных элементов или аномалий.

Однако, какими бы ни были причины, остаётся факт – на практике встречаются вредоносные программы, которые для коммуникации используют собственные протоколы передачи данных, и при разработке систем обнаружения вторжений необходимо это учитывать.

1. Варианты собственных протоколов

По степени сложности самостоятельно написанные протоколы различаются довольно сильно. В простых случаях используются несложные варианты сокрытия, вроде приведения байтов к текстовому виду. Примером может послужить вредоносное ПО для удалённого доступа DarkComet [1], которое передаёт данные, предварительно кодируя последовательность байтов данных в строку ASCII-символов (например, последовательность байт 0x12AC0034 в данном случае будет представлена в виде ASCII-строки «12AC0034»). Также встречаются образцы, которые перед передачей информации кодируют её с использованием алгоритма Base64. Поскольку данные, изменённые таким образом, всё ещё, по сути, находятся в открытом виде, так как механизм подобного изменения обычно общеизвестен и, что более важно, не требует для расшифровки какой-

либо дополнительной информации, то анализ таких случаев является тривиальным, а потому не представляет особого интереса.

Существуют более интересные примеры, в которых разработчики ВПО уже больше внимания уделяют непосредственно защите данных, хотя и всё ещё отдают предпочтение простым алгоритмам. В таких случаях разработчики ВПО обычно придумывают схемы с использованием дополнительной информации, которая, по крайней мере поначалу, известна только человеку, который это ПО использует. Обычно это операции связаны с применением гаммирования, вычитаний определённых байт и других похожих вещей, которые являются вполне посильными как с точки зрения вычислительной сложности, так и с точки зрения программной реализации, и в то же время обеспечивают определённый уровень защиты, и для того, чтобы определить, какая информация содержится в передаваемых данных, уже придётся приложить усилия.

В практических приложениях часто появляется алгоритм RC4. К примеру, он используется при работе программ Warzone [2], Redleaves [3] и многих других. Данный алгоритм основан на применении к данным операции гаммирования. Он достаточно прост в реализации и поэтому данный алгоритм и используется так часто. Само по себе гаммирование может обеспечивать хорошую защиту, однако в случае некорректного использования, например, при использовании одной и той же гаммы в течение долгого времени, данные, зашифрованные таким способом, можно попытаться полностью и частично восстановить.

Самыми сложными с точки зрения обнаружения вредоносных программ являются ситуации, когда разработчики ВПО уделяют проблеме сокрытия передаваемой информации больше внимания. Они используют доказавшие свою надёжность алгоритмы, и в таком случае, если всё сделано правильно, данные, передаваемые по сети, будут выглядеть просто как случайный набор байтов, что и должно происходить в случае правильно реализованного шифрования. Например, так поступает бэкдор Carbanak [4], который начинал с использования гаммирования с фиксированной гаммой, а пришёл к использованию RSA для обмена ключами и AES для шифрования данных. Такую же схему для общения с управляющей машиной использует Pay2Key [5].

2. Методы обнаружения

Поскольку на практике встречаются варианты защиты информации разной степени сложности, то и подходы к их анализу тоже могут быть разными.

Исторически появившееся раньше остальных, а потому наиболее простое решение для классификации трафика, – это использование сигнатур (характерных признаков атаки). Например, можно составить определённые сигнатуры по отношению к трояну remcosRAT [6] и боту QuakBot [7]. Для реализации подобного анализа существует множество программных продуктов, в том числе и свободно распространяемых. К ним относятся такие средства обнаружения вторжений как Bro (Zeek) [8], Snort [9], Surricata.

Очевидным плюсом такого подхода является его потенциально высокая точность, поскольку сигнатуры, по которым будет впоследствии осуществляться анализ, указывают на конкретные специфичные места в данных, и поэтому с большой вероятностью будут работать корректно. Минусом же можно назвать то, что не всегда можно выделить какую-нибудь специфичную часть в данных, и это обстоятельство накладывает серьёзные ограничения на применение сигнатур. Также из-за возникновения нового вредоносного ПО или изменения старого появляется необходимость составлять и добавлять в базы новые сигнатуры и исключать уже не работающие старые, поэтому при использовании

данного подхода необходимо выделять дополнительные ресурсы на поддержание баз сигнатур в актуальном состоянии.

Статистические методы основаны на построении некоторой модели данных, основанной на ряде статистических параметров, и на значениях этих параметров и строится вывод о принадлежности конкретного соединения, или группы соединений, к какому-либо классу. Пример приведен в [10], где предложен подход к объединению сетевых потоков в наборы и извлечение из этих наборов статистических признаков, которые потом можно использовать при обнаружении похожих соединений в другом трафике. Также статистические методы часто используют алгоритмы машинного обучения, такие как логистическая регрессия, SVM (метод опорных векторов), Random Forest (метод случайного леса) и другие. Например, в [11] рассматривается возможность применения метода k-ближайших соседей и метода случайного леса применительно к задаче обнаружения трафика вредоносного ПО на основе статистических признаков, а в [12] для той же цели применяется метод логистической регрессии.

Основным достоинством методов, основанных на данном подходе, является большая универсальность – они не опираются на содержание передаваемых данных, а потому могут применяться и в случае анализа зашифрованного трафика. Также такие методы являются довольно быстрыми и в то же время не особо требовательными к машинным ресурсам, и поэтому могут использоваться в системах, где требуется проводить анализ в режиме реального времени. Недостатком таких методов является необходимость сбора большого количества информации для построения корректно работающей модели. Точность таких систем может значительно снижаться со временем из-за изменения сетевого поведения, поэтому периодически необходимо проводить переобучение, что может быть затруднительно на практике.

Несмотря на успехи других способов, в настоящее время самым многообещающим выглядит использование нейронных сетей и глубокого обучения. В последние годы появилось множество работ, которые показывают возможности такого рода анализаторов. В [13] представлен интересный подход, при котором анализ сетевого трафика осуществляется при помощи одномерной свёрточной нейронной сети (CNN – convolutional neural network), которая в процессе обучения сама выделяет признаки, по которым она потом будет относить конкретное соединение к тому или иному классу. В [14] предлагается подход к классификации вредоносных программ, основанный на глубоком обучении, в котором в качестве модели выступают сети долгой краткосрочной памяти (LSTM). В [15] рассматриваются возможности применения алгоритмов роевого интеллекта к задаче выявления сетевых атак, генетический алгоритм, алгоритм серых волков и алгоритм светлячков. В результате наибольшую точность показал алгоритм серых волков, однако большую устойчивость при изменении параметров нейронной сети показал алгоритм светлячков, из чего авторы делают вывод, что он является более универсальным. В [16] предлагается подход, основанный на представлении соединения в виде временного ряда, и поиск закономерностей в подобных рядах с помощью нейронных сетей. После проведения экспериментов с использованием разных архитектур нейронных сетей, авторы выделяют вариант LSTM как наиболее подходящий для решения данной задачи. В [17] применением нейронных сетей прямого распространения (Feed Forward Neural Network, FFNN) удалось добиться точности в определении протоколов на уровне 90% и выше и точности определения вредоносных программ на уровне 87% и выше. В [18] рассматривается подход, при котором трафик сначала разделяется на легитимный и подозрительный с помощью метода независимых компонент, и уже после этого вредоносный трафик подаётся на вход классификатору на основе алгоритма случайных

лесов. Авторы данной работы демонстрируют, что предварительная обработка данных способна значительно повысить точность работы классификаторов. Похожая идея реализована в [19], где авторы применяют древовидную структуру классификаторов вида «от общего к частному», построенных на рекуррентных нейронных сетях, и таким образом им удаётся добиться значительного улучшения по сравнению с имеющимися на тот момент лучшими решениями. В [20] автор предлагает решение на основе глубокого многослойного перцептрона с модифицированным алгоритмом обучения, который позволяет использовать неоднородные выборки данных и учитывать при обучении априорную информацию о ценности отдельных предметов. В [21] в основу системы обнаружения вторжений положена искусственная иммунная система (ИИС). В табл. 1 приведено сравнение анализаторов.

Таблица 1. Сравнение различных подходов к интеллектуальному анализу трафика

Авторы	Алгоритм	Тип используемых признаков	Достигнутая точность
Wei Wang, Ming Zhu [13]	CNN	Полезная нагрузка	99.41%
Niu W., Zhou J. [14]	LSTM	Ассоциативные правила + временные признаки	96%
Chastikova V.A., Malykhina M.P. [15]	Роевой интеллект	Параметры сетевого трафика	99.3%
Полянская М.С. [16]	LSTM	Временные признаки	84%
Millar K., Smit D. [17]	FFNN	Полезная нагрузка	87%
Mekky H., Mohaisen A. [18]	PCA + RandomForest	N-граммы	95%
Ren X., Gu H. [19]	Tree-RNN	Полезная нагрузка	98.98%
Зуев В.Н. [20]	Многослойный перцептрон	Параметры сетевого трафика	89%
Бурлаков М.Е., Ивкин А.Н [21]	ИИС	Параметры сетевого трафика	74%

К достоинствам нейронных сетей, так же, как и в случае машинного обучения можно отнести возможность использования для анализа самого разного сетевого трафика, при этом в случае использования глубокого обучения даже пропадает необходимость в осознанном выборе признаков, использующихся в модели (но появляется задача выбора представления данных, которое будет использоваться при обучении и использовании модели). Однако проблемы в данном подходе тоже имеются. Во-первых, данные способы (особенно глубокое обучение) могут требовать больших вычислительных затрат, по крайней мере, на этапе обучения. В некоторых моделях могут быть десятки миллионов параметров, которые необходимо оптимизировать в процессе обучения нейронной сети. И хотя ведутся работы в области оптимизации (например, в [22] предлагается метод предобработки, который выбирает для обучающей выборки только потенциально важную информацию, и тем самым уменьшает требования к вычислительным мощностям) и создаются выборки для обучения [23–25], возможности использования таких технологий всё ещё ограничены. Также большое влияние на качество модели оказывает качество обучающей выборки, и с этим могут возникать трудности. Находящихся в открытом доступе наборов данных для тренировки мало, и они могут быть уже не актуальными. В

идеальном случае нужно проводить обучение с использование трафика, снятого из той компьютерной сети, в которой подразумевается использование модели, что может быть сложно. Немаловажной проблемой является актуализация модели, чтобы она могла справляться с детектированием новых вариантов. Полное переобучение модели каждый раз, когда появились какие-либо новые данные – весьма ресурсоёмкий процесс, а дообучение или обобщение уже существующей модели – отдельная непростая задача. Таким образом, использование нейронных сетей и глубокого обучения применительно к задаче классификации сетевого трафика и обнаружения таким образом вредоносной активности выглядит очень перспективно, но на текущем этапе имеется ряд технических задач, которые необходимо решить для того, чтобы можно было использовать данные методы более эффективно. Также в связи с имеющейся тенденцией к увеличению количества ВПО, использующего шифрование [26], возникают вопросы к целесообразности применения сетей, принимающих решение о принадлежности соединения к тому или иному трафику, анализируя данные из полезной нагрузки.

Заключение

Многообразие различных видов вредоносного программного обеспечения позволяет по-разному подходить к задаче обнаружения трафика ВПО. В каких-то случаях можно обойтись сигнатурным анализом, где-то можно применить статистические методы. В особенно сложных случая не обойтись без применения интеллектуальных методов анализа данных: нейронные сети, генетические алгоритмы, искусственные иммунные системы и т.д.

В настоящее время ведутся активные работы по внедрению подобных интеллектуальных методов в реальные системы анализа сетевого трафика. Из проведенного обзора следует, что работы с использованием распространённых нейросетевых архитектур (в частности, CNN и LSTM) позволяют достичь в отдельных случаях очень высокой точности, поэтому на данном этапе в первую очередь стоит обратить внимание на эти варианты решения данной задачи. В тех же работах, в которых авторы пытаются отойти от проверенных способов и применить не такие распространённые методы, результаты в среднем получаются недостаточно точными, чтобы внедрять такие методы на практике, однако в них виден определённый потенциал для развития, поэтому если с помощью нейросетей не удаётся достичь компромисса между точностью и затратами по ресурсам, то, возможно, стоит обратить внимание на альтернативные варианты.

СПИСОК ЛИТЕРАТУРЫ:

1. Farinholt B., Rezaeirad M., McCoy D., Levchenko K. Dark Matter: Uncovering the DarkComet RAT Ecosystem. Proceedings of The Web Conference 2020 WWW ’20: The Web Conference 2020. Taipei Taiwan: ACM, 2020. Dark Matter. P. 2109–2120. DOI: https://doi.org/10.1145/3366423.3380277.
2. Yaroslav Harakhavik Warzone: Behind the enemy lines. URL: https://research.checkpoint.com/2020/warzone-behind-the-enemy-lines/ (дата обращения: 24.10.2022).
3. Shusei Tomonaga RedLeaves – Malware Based on Open Source RAT. URL: https://blogs.jpcert.or.jp/en/2017/04/redleaves---malware-based-on-open-source-rat.html (дата обращения: 24.10.2022).
4. James T. Bennett, Barry Vengerik Behind the CARBANAK Backdoor. URL: https://www.mandiant.com/resources/blog/behind-the-carbanak-backdoor (дата обращения: 24.10.2022).
5. Check Point Research Ransomware Alert: Pay2Key. URL: https://research.checkpoint.com/2020/ransomware-alert-pay2key/ (дата обращения: 27.10.2022).
6. Gh0stCringe RAT Being Distributed to Vulnerable Database Servers. URL: https://asec.ahnlab.com/en/32572/ (дата обращения: 24.10.2022).

7. Anton Kuzmenko, Oleg Kupreev, Haim Zigel QakBot technical analysis. URL: https://securelist.com/qakbot-technical-analysis/103931/ (дата обращения: 24.10.2022).
8. Paxson V. Bro: a system for detecting network intruders in real-time. Computer Networks. 1999, vol. 31, no. 23–24, p. 2435–2463. DOI: https://doi.org/10.1016/S1389-1286(99)00112-7.
9. Roesch M. Snort – Lightweight Intrusion Detection for Networks. Proceedings of the 13th USENIX Conference on System Administration: LISA '99/event-place: Seattle, Washington. USA: USENIX Association. 1999, p. 229–238.
URL: https://www.usenix.org/legacy/event/lisa99/full_papers/roesch/roesch.pdf (дата обращения: 25.11.2022).
10. Piskozub M., Spolaor R., Martinovic I. MalAlert: Detecting Malware in Large-Scale Network Traffic Using Statistical Features. SIGMETRICS Perform. Eval. Rev. 46, 3 (December 2018), p. 151–154. DOI: https://doi.org/10.1145/3308897.3308961.
11. AlAhmadi B.A., Martinovic I. MalClassifier: Malware family classification using network flow sequence behaviour. APWG Symposium on Electronic Crime Research (eCrime), San Diego, CA, USA. 2018, p. 1–13. DOI: https://doi.org/10.1109/ECRIME.2018.8376209.
12. Bapat R. et al. Identifying malicious botnet traffic using logistic regression. 2018 Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, USA. 2018, p. 266–271. DOI: https://doi.org/10.1109/SIEDS.2018.8374749.
13. Wei Wang, Ming Zhu, Xuewen Zeng, Xiaozhou Ye and Yiqiang Sheng. Malware traffic classification using convolutional neural network for representation learning. International Conference on Information Networking (ICOIN), Da Nang, Vietnam. 2017, p. 712–717. DOI: https://doi.org/10.1109/ICOIN.2017.7899588.
14. W. Niu, J. Zhou, Y. Zhao, X. Zhang, Y. Peng, C. Huang. Uncovering APT malware traffic using deep learning combined with time sequence and association analysis, Computers & Security. 2022, v. 120, p. 102809. DOI: https://doi.org/10.1016/j.cose.2022.102809.
15. Частикова В.А., Малыхина М.П., Жерлицын С.А., Воля Я.И. Сравнительный анализ некотрых алгоритмов роевого интеллекта при обнаружении сетевых атак нейросетевыми методами. Научный журнал КубГАУ, 2017, № 129(05). DOI: https://doi.org/10.21515/1990-4665-129-009.
16. Полянская М.С. Обнаружение вредоносной активности в зашифрованном трафике, представленном в виде временных рядов. Современные информационные технологии и ИТ-образование, [S.l.], т. 18, № 1, с. 144–151, 2022. ISSN 2411-1473. DOI: https://doi.org/10.25559/SITITO.18.202201.144-151.
17. Šmít D., Millar K., Page C., Cheng A., Chew H., & Lim C. (2017). Looking deeper: Using deep learning to identify internet communications traffic. URL: https://www.semanticscholar.org/paper/Looking-deeper%3A-Using-deep-learning-to-identify-Šmít-Millar/c037bc2f528c4eb0e025df3aa10587a42c6a2c62 (дата обращения: 25.11.2022).
18. Mekky H., Mohaisen A., Zhi-Li Zhang. Separation of benign and malicious network events for accurate malware family classification. IEEE Conference on Communications and Network Security (CNS), Florence, Italy. 2015, p. 125–133. DOI: https://doi.org/10.1109/CNS.2015.7346820.
19. Ren X., Gu H., Wei W. Tree-RNN: Tree structural recurrent neural network for network traffic classification. Expert Systems with Applications. 2021, v. 167, p. 114363. DOI: https://doi.org/10.1016/j.eswa.2020.114363.
20. Зуев В.Н. Обнаружение аномалий сетевого трафика методом глубокого обучения. Международный журнал «Программные продукты и системы». 2021, т. 25, с. 91–97. DOI: https://doi.org/10.15827/0236-235X.133.091-097.
21. Бурлаков М.Е., Ивкин А.Н. Система обнаружения вторжения на основе искуственной имунной системы. 2020. URL: https://www.researchgate.net/publication/341179692_SISTEMA_OBNARUZENIA_VTORZENIA_NA_OSNOVE_ISKUSSTVENNOJ_IMMUNNOJ_SISTEMY_INTRUSION_DETECTION_SYSTEM_BASED_ON_THE_ARTIFICIAL_IMMUNE_SYSTEM (дата обращения: 25.11.2022).
22. Li F. and Ye F. Adaptive and Lightweight Network Traffic Classification for Edge Devices. IEEE Transactions on Green Communications and Networking. Dec. 2022, vol. 6, no. 4, p. 2003–2014. DOI: https://doi.org/10.1109/TGCN.2022.3165221.
23. García S., Grill M., Stiborek J., Zunino A. An empirical comparison of botnet detection methods. Computers & Security. 2014, v. 45, p. 100–123. DOI: https://doi.org/10.1016/j.cose.2014.05.011.
24. Letteri I., Penna G.D., Vita L.D., Grifa M.T. MTA-KDD'19: A Dataset for Malware Traffic Detection. Proceedings of the Fourth Italian Conference on Cyber Security, Ancona, Italy, February 4th to 7th, 2020, v. 2597, p. 153–165. URL: http://ceur-ws.org/Vol-2597/paper-14.pdf (дата обращения: 25.11.2022).
25. Szumelda P., Orzechowski N., Rawski M., Janicki A. VHS-22 – A Very Heterogeneous Set of Network Traffic Data for Threat Detection. In Proceedings of the 2022 European Interdisciplinary Cybersecurity Conference (EICC '22). Association for Computing Machinery, New York, NY, USA. 2022, p. 72–78. DOI: https://doi.org/10.1145/3528580.3532843.

26. Sean Gallagher. Nearly half of malware now use TLS to conceal communications. 2021. URL: https://news.sophos.com/en-us/2021/04/21/nearly-half-of-malware-now-use-tls-to-conceal-communications/ (дата обращения: 25.11.2022).

REFERENCES:

[1] Farinholt B., Rezaeirad M., McCoy D., Levchenko K. Dark Matter: Uncovering the DarkComet RAT Ecosystem. Proceedings of The Web Conference 2020 WWW '20: The Web Conference 2020. Taipei Taiwan: ACM, 2020. Dark Matter. P. 2109–2120. DOI: https://doi.org/10.1145/3366423.3380277.
[2] Yaroslav Harakhavik Warzone: Behind the enemy lines. URL: https://research.checkpoint.com/2020/warzone-behind-the-enemy-lines/ (accessed: 24.10.2022).
[3] Shusei Tomonaga RedLeaves – Malware Based on Open Source RAT. URL: https://blogs.jpcert.or.jp/en/2017/04/redleaves---malware-based-on-open-source-rat.html (accessed: 24.10.2022).
[4] James T. Bennett, Barry Vengerik Behind the CARBANAK Backdoor. URL: https://www.mandiant.com/resources/blog/behind-the-carbanak-backdoor (accessed:24.10.2022).
[5] Check Point Research Ransomware Alert: Pay2Key. URL: https://research.checkpoint.com/2020/ransomware-alert-pay2key/ (дата обращения: 27.10.2022).
[6] Gh0stCringe RAT Being Distributed to Vulnerable Database Servers. URL: https://asec.ahnlab.com/en/32572/ (accessed: 24.10.2022).
[7] Anton Kuzmenko, Oleg Kupreev, Haim Zigel QakBot technical analysis. URL: https://securelist.com/qakbot-technical-analysis/103931/ (accessed: 24.10.2022).
[8] Paxson V. Bro: a system for detecting network intruders in real-time. Computer Networks. 1999, vol. 31, no. 23–24, p. 2435–2463. DOI: https://doi.org/10.1016/S1389-1286(99)00112-7.
[9] Roesch M. Snort - Lightweight Intrusion Detection for Networks. Proceedings of the 13th USENIX Conference on System Administration: LISA '99/event-place: Seattle, Washington. USA: USENIX Association. 1999, p. 229–238.
URL: https://www.usenix.org/legacy/event/lisa99/full_papers/roesch/roesch.pdf (accessed: 25.11.2022).
[10] Piskozub M., Spolaor R., Martinovic I. MalAlert: Detecting Malware in Large-Scale Network Traffic Using Statistical Features. SIGMETRICS Perform. Eval. Rev. 46, 3 (December 2018), p. 151–154. DOI: https://doi.org/10.1145/3308897.3308961.
[11] AlAhmadi B.A., Martinovic I. MalClassifier: Malware family classification using network flow sequence behaviour. APWG Symposium on Electronic Crime Research (eCrime), San Diego, CA, USA. 2018, p. 1–13. DOI: https://doi.org/10.1109/ECRIME.2018.8376209.
[12] Bapat R. et al. Identifying malicious botnet traffic using logistic regression. 2018 Systems and Information Engineering Design Symposium (SIEDS), Charlottesville, VA, USA. 2018, p. 266–271. DOI: https://doi.org/10.1109/SIEDS.2018.8374749.
[13] Wei Wang, Ming Zhu, Xuewen Zeng, Xiaozhou Ye and Yiqiang Sheng. Malware traffic classification using convolutional neural network for representation learning. International Conference on Information Networking (ICOIN), Da Nang, Vietnam. 2017, p. 712–717. DOI: https://doi.org/10.1109/ICOIN.2017.7899588.
[14] W. Niu, J. Zhou, Y. Zhao, X. Zhang, Y. Peng, C. Huang. Uncovering APT malware traffic using deep learning combined with time sequence and association analysis, Computers & Security. 2022, v. 120, p. 102809. DOI: https://doi.org/10.1016/j.cose.2022.102809.
[15] Chastikova V.A., Malykhina M.P., Zherlitsyn S.A., Volya Y.I. Comparative analysis of some swarm intelligence algorithms with detection of network attacks using neural network methods. 2017, № 129(05). DOI: https://doi.org/10.21515/1990-4665-129-009 (in Russian).
[16] Polyanskaya M.S. Detection of Malicious Activity in Encrypted Traffic Presented as a Time Series. Modern Information Technologies and IT-Education, [S.l.], v. 18, no. 1, p. 144–151, 2022. ISSN 2411-1473. DOI: https://doi.org/10.25559/SITITO.18.202201.144-151 (in Russian).
[17] Šmít D., Millar K., Page C., Cheng A., Chew H., & Lim C. (2017). Looking deeper: Using deep learning to identify internet communications traffic. URL: https://www.semanticscholar.org/paper/Looking-deeper%3A-Using-deep-learning-to-identify-Šmít-Millar/c037bc2f528c4eb0e025df3aa10587a42c6a2c62 (accessed: 25.11.2022).
[18] Mekky H., Mohaisen A., Zhi-Li Zhang. Separation of benign and malicious network events for accurate malware family classification. IEEE Conference on Communications and Network Security (CNS), Florence, Italy. 2015, p. 125–133. DOI: https://doi.org/10.1109/CNS.2015.7346820.
[19] Ren X., Gu H., Wei W. Tree-RNN: Tree structural recurrent neural network for network traffic classification. Expert Systems with Applications. 2021, v. 167, p. 114363. DOI: https://doi.org/10.1016/j.eswa.2020.114363.

[20] Zuev V.N. Network anomalies detection by deep learning. Software & Systems. 2021, vol. 34, no. 1, p. 91–97. DOI: https://doi.org/10.15827/0236-235X.133.091-097 (in Russian).
[21] Burlakov Mikhail & Ivkin A. (2020). Intrusion detection system based on the artifitial immune system. URL: https://www.researchgate.net/publication/341179692_SISTEMA_OBNARUZENIA_VTORZENIA_NA_OSNOVE_ISKUSSTVENNOJ_IMMUNNOJ_SISTEMY_INTRUSION_DETECTION_SYSTEM_BASED_ON_THE_ARTIFICIAL_IMMUNE_SYSTEM (accessed: 25.11.2022) (in Russian).
[22] Li F. and Ye F. Adaptive and Lightweight Network Traffic Classification for Edge Devices. IEEE Transactions on Green Communications and Networking. Dec. 2022, vol. 6, no. 4, p. 2003–2014. DOI: https://doi.org/10.1109/TGCN.2022.3165221.
[23] García S., Grill M., Stiborek J., Zunino A. An empirical comparison of botnet detection methods. Computers & Security. 2014, v. 45, p. 100–123. DOI: https://doi.org/10.1016/j.cose.2014.05.011.
[24] Letteri I., Penna G.D., Vita L.D., Grifa M.T. MTA-KDD'19: A Dataset for Malware Traffic Detection. Proceedings of the Fourth Italian Conference on Cyber Security, Ancona, Italy, February 4th to 7th, 2020, v. 2597, p. 153–165. URL: http://ceur-ws.org/Vol-2597/paper-14.pdf (accessed: 25.11.2022).
[25] Szumelda P., Orzechowski N., Rawski M., Janicki A. VHS-22 – A Very Heterogeneous Set of Network Traffic Data for Threat Detection. In Proceedings of the 2022 European Interdisciplinary Cybersecurity Conference (EICC '22). Association for Computing Machinery, New York, NY, USA. 2022, p. 72–78. DOI: https://doi.org/10.1145/3528580.3532843.
[26] Sean Gallagher. Nearly half of malware now use TLS to conceal communications. 2021. URL: https://news.sophos.com/en-us/2021/04/21/nearly-half-of-malware-now-use-tls-to-conceal-communications/ (accessed: 25.11.2022).

Поступила в редакцию – 25 ноября 2022 г. Окончательный вариант – 20 марта 2023 г.
Received – November 25, 2022. The final version – March 20, 2023.