моделей // Современные проблемы науки и образования, 2020. 5. 34. DOI: 10.17513/spno.30142.

15. Краснощеков В.В., Семенова Н.В., Алсалама А.М., Михолитсис А.Г. О точных и приближенных моделях в вузовском курсе теории вероятности // Современные наукоемкие технологии, 2021. 10. 149 – 154. DOI: 10.17513/snt.38869.

16. Краснощеков В.В., Семенова Н.В., Мухамед Б.М.М., Баккар М.М.А. О выборе из конечного и бесконечного поля в курсе теории вероятности // Современные наукоемкие технологии, 2022. 9. 138 – 143. DOI: 10.17513/snt.39322.

17. Краснощеков В.В., Семенова Н.В., Аббас А., Шбиб Х. Формирование у студентов вуза представлений о точности и надежности оценки вероятности // Современные наукоемкие технологии, 2024. 7. 163 – 170. DOI: 10.17513/snt.40102.

УДК 004.056.5
doi:10.18720/SPBPU/2/id25-411

Пап Адам[1]
бакалавр, студент предмагистратуры;
Рышавы Ондржей[2]
доцент, кандидат технических наук

АВТОМАТИЧЕСКОЕ ОБНАРУЖЕНИЕ АКТИВНОСТИ ВРЕДОНОСНЫХ ПРОГРАММ В ЛОКАЛЬНЫХ СЕТЯХ

[1] Россия, Санкт-Петербург,
Санкт-Петербургский политехнический университет
Петра Великого,
[1,2] Чешская Республика, Брно, Технический университет в Брно,
[1] adam.pap146@gmail.com, [2] rysavy@fit.vut.cz

Аннотация. В рамках работы был создан набор данных, из которого были извлечены IoC для каждого семейства вредоносных программ. Полученные IoC были проверены через платформу AlienVault OTX для подтверждения их релевантности. На тестовых данных обе IoC-модели, созданные из наборов данных, достигли точности 99,337% и 94,732% для набора данных №1 и №2 соответственно. IoC-модели набора №1 в реальной эксплуатации ложно классифицировали 3,03% коммуникационных окон как вредоносные. IoC-модели набора № 2 классифицировали

5,66% окон как вредоносные. В тестовой среде были запущены образцы различных семейств вредоносного ПО: модели набора № 1 классифицировали 7,14% как вредоносные, а модели №2 классифицировали 15,79% как вредоносные.

Ключевые слова: вредоносное ПО, набор данных, индикатор компрометации.

Adam Pap[1]
Bachelor, pre-master student;
Ondřej Ryšavý[2]
Docent, PhD in Computer Science

AUTOMATED DETECTION OF MALWARE ACTIVITY IN LOCAL NETWORKS

[1] Peter the Great St. Petersburg Polytechnic University,
St. Petersburg, Russia,
[1,2] Brno university of Technology, Brno, Czech republic,
[1] adam.pap146@gmail.com, [2] rysavy@fit.vut.cz

Abstract. As part of the work, a dataset was created from which an IoC for each malware family were extracted. These IoCs were then validated through the AlienVault OTX platform, in order to verify their relevance. On the test data, the two IoC models created from the datasets achieved an accuracy of 99,337% and 94,732% for dataset № 1 and № 2, respectively. The IoC models of dataset №1 falsely classified 3,03% of communication windows as malicious in real communication. IoC models of set №2 classified 5,66% as malicious. After the samples of different malware families were run on the machine, the IoC models of set №1 classified 7,14% of the windows as malicious. Set №2 models classified 15,79%.

Key words: malicious software, data set, compromise indicator.

ВВЕДЕНИЕ

В настоящее время, несмотря на изобилие информации, интернет-технологий и других достижений технического прогресса, которые значительно облегчают жизнь, появляется и много негативных аспектов, связанных с этим. Постоянные атаки в Интернете с целью получения или иного злоупотребления данными

отдельных лиц или групп людей с помощью вредоносного программного обеспечения привели к созданию различных инструментов мониторинга.

Слово «malware» является словом, образованным из двух английских слов: «malicious» (злонамеренный) и «software» (программное обеспечение). Его также можно определить как злонамеренный код или программное обеспечение, которое обычно пытается повредить, заблокировать или иным образом завладеть информацией в зараженной системе [1].

Для выявления деятельности вредоносного программного обеспечения можно использовать индикаторы компрометации или индикаторы атаки.

Индикатор компрометации (англ. Indicator of Compromise, IoC) – это криминалистические данные, которые используются в рамках кибербезопасности для подтверждения или опровержения наличия кибератак. Помимо прочего, они также применяются при разработке различных стратегий защиты от упомянутых атак. IoC можно использовать для выявления слабых мест в безопасности данной системы, а также для определения способа, которым была осуществлена кибератака.

IoC обычно встречаются в нескольких формах. К наиболее распространенным из них относятся, например, подозрительная коммуникация, исходящая с конкретного компьютера, большое количество неудачных попыток входа в систему, коммуникация с каким-либо экзотическим IP-адресом и др. Таким образом, в рамках определения IoC можно сказать, что они указывают на поведение или данные, которые свидетельствуют о том, что произошло проникновение в систему, нарушение ее данных или другое нападение. Данные, полученные из IoC, находят применение в первую очередь при поиске угроз, поскольку эти данные обычно становятся доступными только после взлома системы [2].

Индикаторы атаки (англ. Indicators of Attack, IoA) можно определить как поведение или шаблоны (англ. patterns), используемые для идентификации активной атаки. Основное различие между IoA и IoC заключается в статусе атаки, то есть в том,

осуществляется ли атака в данный момент или уже была осуществлена. IoA идентифицирует намерения вредоносного ПО и техники, используемые во время атаки на компьютерную систему или сеть. Если IoA обнаруживает активную атаку, то IoC используются для более тщательного анализа уже завершенной атаки. С помощью IoA можно предотвратить данную атаку во время ее осуществления [2].

Своевременная идентификация IoA обычно означает, что ситуацию или данные еще можно спасти, прежде чем она усугубится и ухудшится. IoC, с другой стороны, могут указать, кто стоит за атакой, как она произошла и какие инструменты были использованы.

Таким образом, основное различие заключается в том, что IoC основаны на известной вредоносной деятельности определенной семьи вредоносных программ, в то время как IoA основаны на конкретных тактиках, техниках и процедурах, используемых злоумышленниками, как показано на рис. 1 [3].

Целью данной работы является анализ сетевого взаимодействия вредоносного ПО и последующее выявление значимых признаков, которые позволят разработать подходящий метод его автоматического обнаружения. *Задачи исследования*:

1. Ознакомиться с коммуникациями вредоносных программ и определить дополнительные источники данных, полезные для выявления признаков вредоносного ПО в сетевых сообщениях.

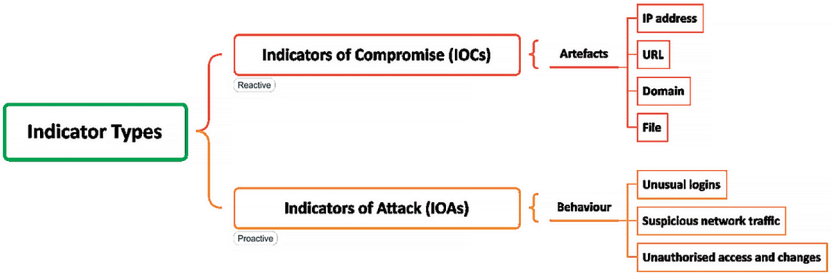

Рис. 1. Различие между индикатором компрометации IoC и индикатором атаки IoA [3]

2. Создать наборы данных для разработки и тестирования инструмента, сосредоточившись на извлечении значимых признаков из сетевого трафика.

3. Разработать и реализовать метод выявления вредоносных программ, который сочетает различные техники, в качестве отдельного инструмента.

Методы и ход исследования

Наборы данных были созданы с помощью инструмента, разработанного в рамках бакалаврской работы «Обнаружение коммуникации вредоносного ПО в сетевых потоках» [4]. Набор данных, который был создан с целью разработки и тестирования инструмента для обнаружения вредоносных программ в сетевой коммуникации, также использовался для анализа, с помощью которого были идентифицированы важные свойства отдельных семейств вредоносных программ. Инструмент, с помощью которого был создан данный набор данных, использует песочницу Triage.

Triage – это веб-приложение – песочница, в которое через веб-браузер можно загружать различные образцы вредоносных программ для анализа. Анализ такого образца состоит из статического и динамического анализа. Результаты обоих анализов можно скачать через веб-браузер или приложение с доступом к API в формате PDF или JSON (англ. Javascript object notation).

Набор, созданный в рамках этой работы, состоит из 17 семейств вредоносных программ, и каждое семейство имеет по 5 образцов вредоносных ПО. Семейства были выбраны на основе веб-сайта AnyRun. В рамках анализа основное внимание уделялось индикаторам компрометации отдельных образцов вредоносного ПО. Анализ этих криминалистических данных был выполнен с помощью методов, изложенных в книге J. Jacobs и B. Rudis [5] и языка Python, а именно с использованием IPython notebook.

Прежде чем объяснить метод обнаружения, было бы целесообразно объяснить, на основе каких данных проводилось обнаружение и как были эти данные получены. Каждый тип IoC (IP,

домен, URL) представлен двумя моделями (далее именуемыми «нечеткими множествами»), чтобы можно было более точно определить, является ли данный IoC релевантным или нет. Это означает, что хотя данный IoC получает в первом нечетком множестве определенный балл, расчет которого объясняется далее, однако оценка AlienVaultOTX может быть ниже или равна 0, и, следовательно, при обнаружении данный индикатор не будет иметь такого веса или значимости, как индикаторы, получившие гораздо более высокую оценку.

Расчет значений баллов отдельных IoC в первом множестве рассчитывается как количество вхождений IoC в данной семье, деленное на общее количество всех IoC в семье, таким образом был получен балл среднего вхождения данного индикатора в рамках одной семьи.

Во втором нечетком множестве оценка отдельных IoC рассчитывается иначе, а именно с использованием AlietVault OTX, из которого выбирается количество импульсов, представляющих собой набор IoC, сообщенных сообществом платформы AlienVault. Подсчитывается количество пассивных DNS-записей (имена хостов, домены), связанных с данным индикатором, как показано на рис. 2.

Затем это делится на значение 2, что дает сырое (англ. raw) значение для данного индикатора. Однако необходимо, чтобы значения в этих нечетких множествах были нормализованы, поскольку они находятся в разных диапазонах. Нормализация числовых значений – это процесс, при котором происходит масштабирование значений до известного диапазона. В данной работе это диапазон:

1: $pulse\_count = get\_pulses\_OTX(ioc)$
2: $passive\_dns\_count = get\_passive\_DNS\_records(ioc)$
3: $raw\_membership\_val = (pulse\_count + passive\_dns\_count)/2.0$
4: $max\_val = get\_max\_raw\_memebership$
5: $min\_val = get\_min\_raw\_memebership$
6: $normalized\_val = (raw\_membership\_val - min\_val)/(max\_val - min\_val)$

Рис. 2. Псевдокод, описывающий расчет балла IoC с помощью значений из AlienVault OTX

0 – 1. Масштабирование выполняется с помощью метода Min-Max, формула которого выглядит следующим образом:

$$X_{norm} = (X - X_{\min})(X_{\max} - X_{\min}) \quad (1)$$

После того, как значения правильно нормализованы, они записываются в нечеткие множества, с которыми далее работают при обнаружении.

Сам метод обнаружения, разработанный в рамках данной работы, функционирует следующим образом. После извлечения IoA из отдельных зафиксированных потоков с помощью инструмента «Suricata» создаются коммуникационные окна. Для каждого исходного IP-адреса отдельные потоки распределяются по окнам на основе временной метки (англ. timestamp). Причина, по которой потоки группируются в коммуникационные окна, заключается в том, что желательно знать, какую коммуникацию осуществляли отдельные станции или компьютеры в определенные временные интервалы. Благодаря этому можно выявить контекст сетевых событий. Само по себе это может быть ключом к выявлению паттернов (англ. patterns) коммуникации вредоносного ПО или других аномалий, которые произошли в данном временном интервале сетевой коммуникации. Эти коммуникационные окна уже содержат отдельные IoA.

Теперь необходимо определить, насколько эти IoA совпадают с моделями IoC. То есть IoA обозначим как нечеткое множество FA, а нечеткие множества отдельных моделей обозначим как FC. Затем выполняется пересечение между этими двумя нечеткими множествами $FR = FA \cap FC$. Пересечение между этими множествами выполняется с помощью функции min, которая сравнивает два значения и возвращает меньшее из них. Оценка отдельных IoA в нечетком множестве FA равна 1, потому что эти IoA встречались в перехваченной коммуникации, то есть степень их встречаемости определена, то есть равна 1. Таким образом, в результирующем пересечении FR будут находиться только IoC данной семьи вредоносных программ, которые были найдены в рамках перехваченной коммуникации, соответственно, перехваченного коммуникационного окна.

Далее необходимо выяснить, достаточно ли количество зафиксированных IoC в проникновении FR, чтобы данное окно можно было объявить потенциально вредоносным. Для этого необходимо рассчитать так называемое пороговое значение (англ. treshold value) для каждой семьи вредоносных программ. Пороговое значение получается как сумма баллов IoC (например IP-адресов), которые были найдены в отдельных образцах конкретных семейств вредоносных программ. На практике данные модели использовались для обнаружения образцов вредоносных программ, на основе которых эти модели были созданы. То есть, постепенно брались образцы отдельных семейств вредоносных программ в качестве коммуникации, которую необходимо проанализировать и, в случае необходимости, обнаружить в ней вредоносную коммуникацию. Если данный IoC из модели встречается в данном образце, его достигнутый балл прибавляется к общей сумме. В конце концов, для каждой выборки конкретного семейства вредоносных программ будет вычислено только среднее значение этих двух сумм.

Результаты исследования

Эксперименты, приведенные в табл. 1, были проведены над перехваченной коммуникацией, созданной на основе полученных данных отдельных наборов данных.

Таблица 1

Таблица результатов

Режим	Набор данных	Всего окон	Отмечено вредоносных	Эталонное вредоносных	Процент неправильных обнаружений
Оффлайн	№1	5734	458	456	-
Оффлайн	№2	1101	558	548	-
Онлайн	№1	96	4	-	3,03%
Онлайн	№2	72	6	-	5,66%

Выводы

В заключении важно отметить направления, по которым дальнейшая разработка инструмента представляется наиболее перспективной. Во-первых, безусловно, было бы полезно предоставить методу больше данных для построения сильных IoC-моделей, содержащих достаточно информации, чтобы метод мог принимать решение, является ли данное окно вредоносным или нет. Во-вторых, интересным расширением могло бы быть использование алгоритма машинного обучения для определения пороговых значений для каждого окна отдельно, а не глобально.

В целом созданный инструмент обнаружения может найти применение на практике, прежде всего, как средство отслеживания вредоносных сообщений в сетевых коммуникациях. Онлайн-обнаружение нуждается в улучшении, прежде всего в том, как определяется порог.

ЛИТЕРАТУРА

1. Malware or malicious software definition [Electronic recourse] // Malwarebytes. URL: https://www.malwarebytes.com/malware.

2. Asiri M., Saxena N., Gjomemo R., Burnap P. Understanding Indicators of Compromise against Cyber-Attacks in Industrial Control Systems: A Security Perspective. ACM Transactions on Cyber-Physical Systems, 2023. Vol. 7 (2). 15. 1 – 33. DOI: 10.1145/3587255.l.

3. Maciuitis M. Importance of IOC Detection Rules [Electronic recourse] // Talanos Cybersecurity. July 13, 2023. URL: https://www.talanoscybersecurity.com/blogs/news/importance-of-ioc-detection-rules.

4. Korvas V. Detekce komunikace malware v síťových tocích: bachelor's thesis / V. Korvas; supervisor doc. Ing. Ondřej Ryšavý, Ph.D. [Electronic recourse]. Brno University of Technology. Brno, 2023. 43 p. URL: https://www.vut.cz/www_base/zav_prace_soubor_verejne.php?file_id=2520.

5. Jacobs J., Rudis B. Data-driven security: analysis, visualization and dashboards / John Wiley & Sons, Indianapolis, Indiana, USA, 2014. 352 p. URL: https://virtualmmx.ddns.net/gbooks/Data-DrivenSecurity.pdf.