УДК 004

А. Е. Софронов[1]✉

Автоматизированный стенд для сравнения производительности инструментов пассивного анализа сетевого трафика

[1]Национальный исследовательский ядерный университет «МИФИ»,
г. Москва, Российская Федерация
e-mail: sofronovae123456@gmail.com

Аннотация. В статье рассматривается проблема сравнения инструментов пассивного анализа сетевого трафика на основе формализованных критериев при идентичных условиях. Выявлено, что существующие исследования в данной области проводятся в условиях отсутствия общепризнанной строгой формализованной методики и сетевой лаборатории с различными методиками оценки, что затрудняет сопоставление результатов. Была представлена архитектура стенда, созданного по принципу «Инфраструктура как код», включающего модули генерации трафика, развертывания инструментов, сбора метрик и визуализации результатов. Разработанный стенд обеспечивает последовательное тестирование инструментов в идентичных условиях, автоматизированный сбор данных о потреблении ресурсов и числе обработанных пакетов. Проведена апробация стенда на инструментах Suricata и Zeek. Полученные результаты демонстрируют возможность сравнения решений в стандартизированных условиях и выявления их характеристик производительности.

Ключевые слова: пассивный анализ трафика, сетевая безопасность, виртуальный стенд, сравнительный анализ, автоматизация тестирования, инфраструктура как код

A. E. Sofronov[1]✉

Automated test bench for performance comparison of passive network traffic analysis tools

[1]National Research Nuclear University MEPhI (Moscow Engineering Physics Institute),
Moscow, Russian Federation
e-mail: sofronovae123456@gmail.com

Abstract. The article addresses the problem of comparing passive network traffic analysis tools based on formalized criteria under identical conditions. It has been found that existing research in this field is conducted in the absence of a generally accepted strict formalized methodology and network laboratory with various assessment methods, which makes it difficult to compare results. The architecture of a test bench created on the "Infrastructure as Code" principle was presented, including modules for traffic generation, tool deployment, metrics collection, and results visualization. The developed test bench provides sequential testing of tools under identical conditions, automated collection of data on resource consumption and the number of processed packets. The test bench was validated using Suricata and Zeek tools. The obtained results demonstrate the possibility of comparing solutions under standardized conditions and identifying their performance characteristics.

Keywords: passive traffic analysis, network security, virtual test bench, comparative analysis, test automation, infrastructure as code

Введение

В условиях усложнения сетевых инфраструктур и роста числа кибератак, обнаружение вторжений и анализ трафика становятся все более важными для безопасности предприятий [1]. На рынке представлено множество инструментов пассивного анализа. Различными научными школами ведутся разработки новых инструментов [2, 3]. Однако существует проблема их стандартизированного сравнения для выбора оптимального решения.

Сравнительные исследования таких инструментов часто проводятся в отсутствие строгой формализованной методики и идентичных условий, что приводит к вариативности результатов. Отсутствие стандартизированной методики и единой тестовой среды затрудняет формирование однозначных выводов о производительности этих инструментов и, как следствие, выбор более подходящего.

Существенным недостатком исследований является отсутствие воспроизводимости экспериментов. Для активного анализа эта проблема решена созданием виртуальных стендов с идентичными условиями тестирования [4]. Для пассивного анализа также целесообразно разработать новую платформу для средств пассивного мониторинга, использующую принципы виртуализации и автоматизации тестирования [5].

Разработка такого стенда обеспечит воспроизводимость результатов и поможет формализовать выбор оптимального инструмента для мониторинга сетевой инфраструктуры.

Исследование предметной области

Пассивный анализ сетевого трафика представляет собой неинвазивный метод мониторинга информационных систем, при котором не осуществляется вмешательство в сетевой трафик, а осуществляется только захват и анализ передаваемых данных. В отличие от активного анализа, который предполагает взаимодействие с исследуемыми объектами посредством отправки сетевых пакетов, пассивные методы не оставляют следов в системных журналах и не влияют на функционирование сетевых устройств, что делает их незаменимыми при решении задач мониторинга безопасности и диагностики проблем в высоконагруженных сетях [6].

Согласно одной из возможных классификаций, инструменты пассивного анализа можно разделить на следующие категории: анализаторы протоколов (Wireshark, tcpdump), системы обнаружения и предотвращения вторжений (Snort, Suricata, Bro/Zeek), системы мониторинга и анализа потоков (netflow, sflow), а также специализированные решения для глубокого анализа пакетов (DPI (англ. Deep Packet Inspection)). Каждая категория имеет свою специфику работы и требования к аппаратным ресурсам, что усложняет процесс их сравнения и выбора оптимального инструмента для конкретных задач. В данной статье будет представлено использование стенда для сравнения решений класса NIDS (англ. Network Intrusion Detection System), таких как Suricata и Zeek [7, 8].

Для формирования полноценного понимания проблематики сравнения инструментов пассивного анализа была проведена систематизация научно-технической литературы и существующих решений. Анализ публикаций, посвященных сравнению средств пассивного анализа трафика, выявил несколько ключевых проблем.

Существующие исследования используют различные наборы метрик и методологии тестирования, что затрудняет сопоставление результатов разных работ. Во многих исследованиях внимание акцентируется на определенных аспектах работы инструментов, при недостаточном учете других важных параметров. Например, оценивается точность обнаружения угроз, но не учитывается потребление ресурсов или масштабируемость решения [9].

Важным аспектом исследования предметной области является определение ключевых метрик производительности инструментов пассивного анализа. На основе анализа литературы и практических требований к системам мониторинга были выделены следующие параметры, которые недостаточно учитываются в сравнительных статьях [10–12]:

доля потерянных пакетов при различных уровнях нагрузки в процентах от общего объема переданного трафика;

- использование процессора в процентах от одного ядра;
- потребление оперативной памяти в мегабайтах;
- динамика потребления ресурсов при увеличении нагрузки.

Параметры, связанные с точностью обнаружения различных угроз, не рассматриваются, так как они определяются не столько самим инструментом, сколько его конфигурацией: набором правил или сигнатур, точностью обучения модели (в случае ML-инструментов) или набором подключенных модулей [13]. Особое внимание было уделено технологиям виртуализации и средствам конфигурации, которые обеспечивают воспроизводимость экспериментов.

Постановка задачи

На основе проведенного исследования предметной области сформулирована цель работы – разработка виртуального стенда для стандартизированного сравнения инструментов пассивного анализа трафика с акцентом на оценку их производительности.

Для достижения поставленной цели необходимо решить следующие задачи:

– разработка архитектуры виртуального стенда, включающей модули, необходимые для воспроизводимости и наблюдаемости эксперимента;

– интеграция стенда с популярными инструментами пассивного анализа трафика (Suricata, Zeek).

Также для дальнейшего развития стенда важно было предусмотреть возможность расширения и поддержки новых инструментов и сценариев тестирования.

Основным результатом работы должен стать программный комплекс, позволяющий проводить объективное сравнение производительности различных инструментов пассивного анализа сетевого трафика в стандартизированных условиях с

возможностью воспроизведения экспериментов и расширения функциональности для поддержки новых инструментов и сценариев тестирования.

Архитектура виртуального стенда

Разработанный виртуальный стенд представляет собой программный комплекс для автоматизированного сравнительного анализа инструментов пассивного мониторинга сетевого трафика. Архитектура стенда основана на принципе «Инфраструктура как код» [14] и позволяет создавать идентичные тестовые среды для каждого эксперимента, что обеспечивает воспроизводимость результатов.

Основу стенда составляют следующие компоненты:

– отдельные виртуальные среды для каждого тестируемого инструмента (Suricata, Zeek), запускаемые последовательно для обеспечения идентичности условий тестирования;

– генератор трафика, развертываемый на каждой виртуальной машине и отвечающий за воспроизведение PCAP-файлов с заданной скоростью для имитации реальной сетевой нагрузки;

– система сбора метрик, осуществляющая мониторинг потребления ресурсов (CPU, RAM) и подсчитывающая количество обработанных пакетов;

– модуль визуализации результатов для построения графиков и таблиц на основе собранных метрик, запускаемый после выполнения тестирования всех инструментов.

Представленная на рис. 1 архитектура стенда была спроектирована в соответствии с принципом «Инфраструктура как код», что обеспечивает возможность создания идентичных тестовых сред при каждом запуске эксперимента. Такой подход необходим для обеспечения воспроизводимости результатов.

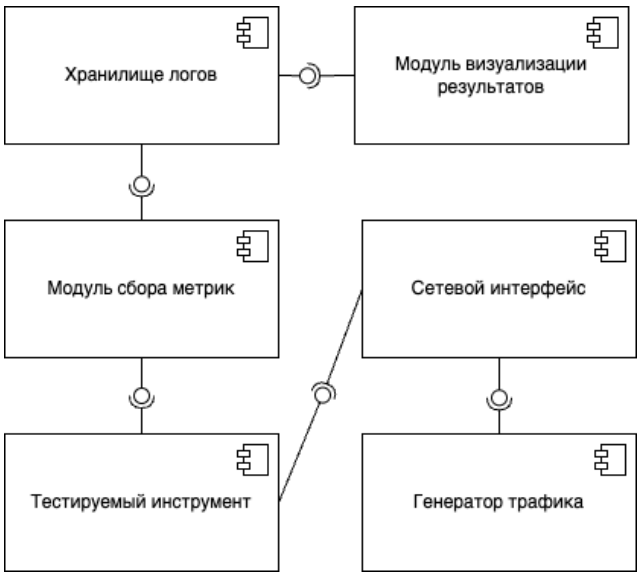

Рис. 1. UML-диаграмма компонентов виртуального стенда

Последовательный запуск виртуальных машин обеспечивает изоляцию тестирования и предотвращает взаимное влияние инструментов друг на друга. Такой подход гарантирует, что каждый инструмент тестируется в идентичных условиях, с использованием одинаковых ресурсов и на одном и том же наборе данных. Для обеспечения идентичности условий тестирования все виртуальные машины имеют одинаковую конфигурацию аппаратных ресурсов: 4 виртуальных ядра CPU и 4096 МБ оперативной памяти.

Взаимодействие между компонентами стенда управляется набором скриптов, что минимизирует влияние человеческого фактора на процесс тестирования. Полная автоматизация процедуры тестирования от развертывания виртуальных машин до сбора и визуализации результатов позволяет проводить серии экспериментов с различными параметрами и быстро получать сравнительные данные о производительности анализируемых инструментов.

Процесс проведения эксперимента подробно представлен на рис. 2 в виде блок-схемы, иллюстрирующей последовательность шагов от начала тестирования до получения итоговых результатов.

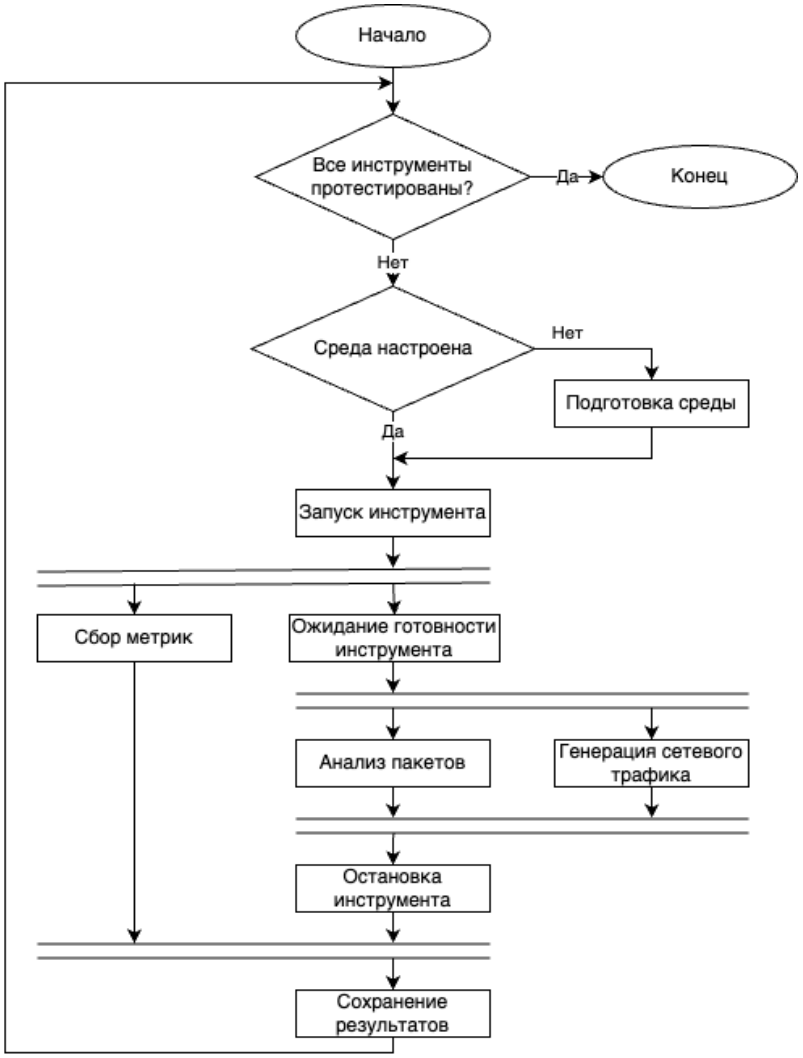

Рис. 2. Блок-схема проведения эксперимента

Весь процесс тестирования полностью автоматизирован с использованием скриптов на языке Bash [15]. Основные операции включают:

– подготовка среды – скрипты setup_*.sh выполняют установку и конфигурирование инструментов;

– запуск инструментов – скрипты run_*.sh отвечают за корректный запуск инструментов с необходимыми параметрами;

– ожидание готовности – скрипты wait_*.sh обеспечивают синхронизацию запуска генератора трафика с моментом, когда инструмент полностью готов к приему данных;

– остановка и сохранение результатов – скрипты stop_*.sh и collect_*.sh корректно завершают работу инструментов и собирают необходимую статистику.

Управление всем процессом осуществляется через основной скрипт test.sh, который последовательно выполняет все этапы тестирования и генерирует итоговые графики с результатами.

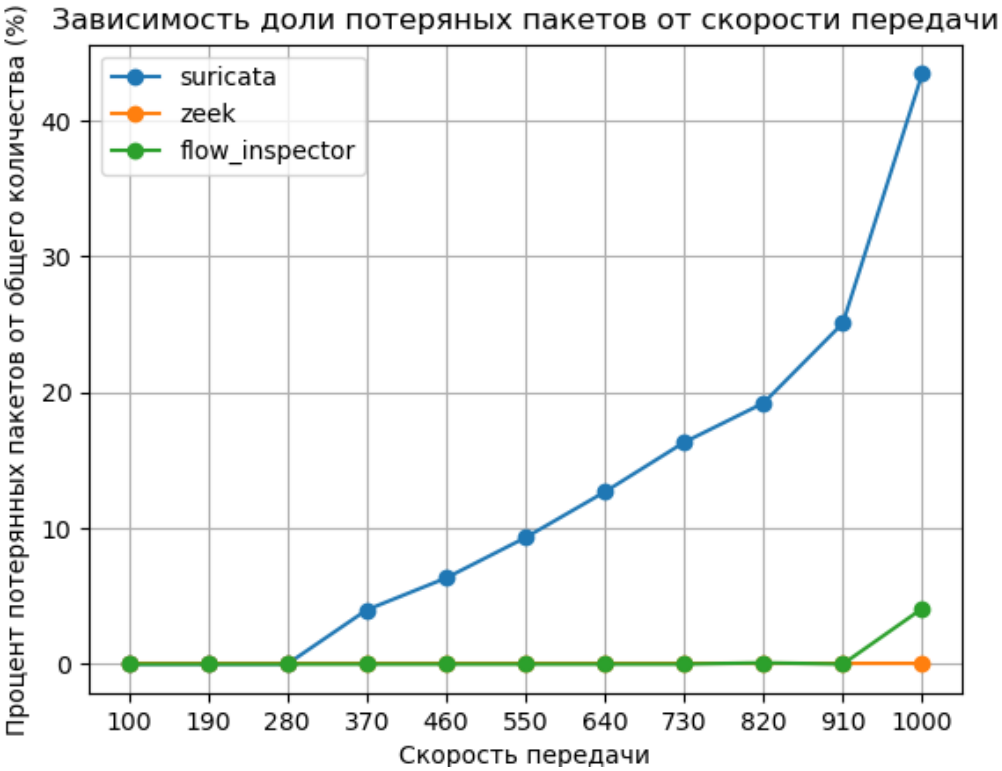

Рис. 3. График зависимости доли потерь пакетов от скорости передачи (в мегабайтах/секунду)

Система сбора метрик основана на библиотеке psutil и позволяет получать детальную информацию о потреблении ресурсов процессами. Для каждого инструмента собираются следующие метрики:

– процент загрузки процессора процессами инструмента;

– число пакетов, успешно захваченных и обработанных инструментом (рис. 3);

– объем используемой RAM в мегабайтах (рис. 4).

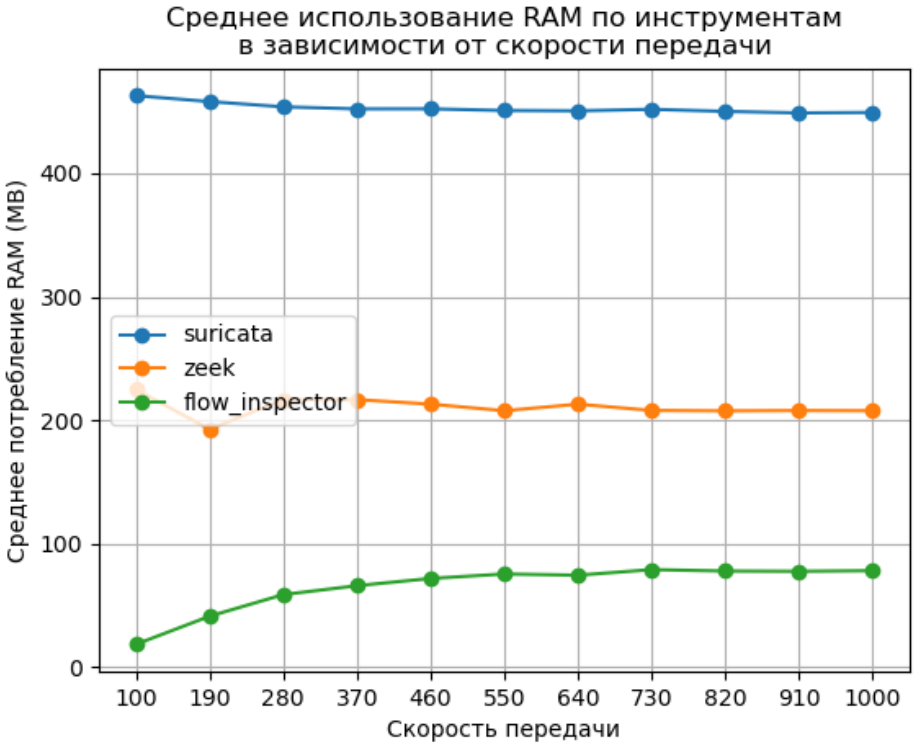

Рис. 4. График зависимости среднего потребления оперативной памяти от скорости передачи (в Мб/с)

Метрики собираются с интервалом в 1 секунду и агрегируются для последующего анализа. Для визуализации результатов используются скрипты на языке Python с библиотекой Matplotlib [16]. Архитектура стенда спроектирована с учетом возможности легкого расширения функциональности. Для интеграции нового инструмента достаточно создать соответствующие скрипты настройки, запуска, ожидания и остановки, следуя существующему шаблону. Стенд поддерживает работу с различными PCAP-файлами, что позволяет тестировать инструменты на разных типах трафика. Система конфигурирования позволяет легко модифицировать диапазон скоростей передачи, количество итераций и другие параметры экспериментов.

Таким образом, разработанный виртуальный стенд представляет собой полноценную платформу для проведения объективного сравнительного анализа инструментов пассивного мониторинга сетевого трафика, обеспечивая воспроизводимость результатов и возможность оценки различных аспектов производительности этих инструментов.

Заключение

В рамках данного исследования был разработан универсальный виртуальный стенд для объективного сравнительного анализа инструментов пассивного мониторинга сетевого трафика. Стенд реализует полностью автоматизирован-

ный процесс тестирования, включающий последовательное развертывание тестовых сред и проведение экспериментов с различной скоростью трафика.

Все поставленные задачи исследования были успешно решены: разработана модульная архитектура стенда, реализовано автоматизированное развертывание тестовых окружений, обеспечена интеграция с популярными инструментами анализа и создана система объективной оценки производительности.

Основным преимуществом разработанного стенда является обеспечение стандартизированной среды тестирования, которая гарантирует идентичность условий для всех сравниваемых решений. Это позволяет исключить влияние внешних факторов на результаты экспериментов и сосредоточиться на реальных различиях в производительности инструментов.

В качестве перспективных направлений развития стенда можно выделить расширение набора тестируемых инструментов и внедрение дополнительных метрик.

Практическая значимость разработанного стенда заключается в возможности его применения для выбора оптимального инструмента анализа трафика при проектировании систем сетевой безопасности предприятий. Это позволит точнее прогнозировать поведение инструментов в реальных условиях эксплуатации.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Прошина Т. Д., Сальникова Н. А. Анализ сетевых атак и их проявлений //Тенденции развития науки и образования. – 2022. – №. 84-2. – С. 29-33.

2. Свидетельство о государственной регистрации программы для ЭВМ № 2023619143 Российская Федерация. Система пассивного анализа трафика вычислительной сети : № 2023617407 : заявл. 20.04.2023 : опубл. 04.05.2023 / Е. А. Басыня, Н. Е. Изъюров, К. Г. Когос [и др.] ; заявитель федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский ядерный университет «МИФИ».

3. Изъюров, Н. Е. Разработка инструмента пассивного анализа трафика вычислительной сети / Н. Е. Изъюров // Интерэкспо Гео-Сибирь. – 2023. – Т. 7, № 1. – С. 211-223.

4. Свидетельство о государственной регистрации программы для ЭВМ № 2024667030 Российская Федерация. Виртуальная сетевая лаборатория для проведения тестирования инструментов активного анализа информационных систем : № 2024665669 : заявл. 05.07.2024 : опубл. 18.07.2024 / Е. А. Басыня, Г. К. Крючков, К. Г. Когос [и др.] ; заявитель федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский ядерный университет «МИФИ».

5. Epishkina A. V., Kanner A. M., Kanner T. M. Comprehensive Testing of Software and Hardware Data Security Tools Using Virtualization //Advanced Technologies in Robotics and Intelligent Systems: Proceedings of ITR 2019. – Springer International Publishing, 2020. – С. 79-87.

6. Басыня, Е. А. Системное администрирование и информационная безопасность / Е. А. Басыня. – Новосибирск : Новосибирский государственный технический университет, 2018. – 79 с.

7. Бондяков А. С. Основные режимы работы системы предотвращения вторжений (IDS/IPS Suricata) для вычислительного кластера //Современные информационные технологии и ИТ-образование. – 2017. – Т. 13. – №. 3. – С. 31-37.

8. Маркин Ю. В., Санаров А. С. Обзор современных инструментов анализа сетевого трафика //Препринты ИСП РАН. – 2014. – №. 27. – С. 1.

9. Bada G. K., Nabare W. K., Quansah D. Comparative analysis of the performance of network intrusion detection systems: Snort suricata and bro intrusion detection systems in perspective //International Journal of Computer Applications. – 2020. – Т. 176. – №. 40. – С. 39-44.

10. Abdel-Basset M. et al. An optimization model for appraising intrusion-detection systems for network security communications: Applications, challenges, and solutions //Sensors. – 2022. – Т. 22. – №. 11. – С. 4123.

11. Nasr K., El Kalam A. A. A novel metric for the evaluation of idss effectiveness //ICT Systems Security and Privacy Protection: 29th IFIP TC 11 International Conference, SEC 2014, Marrakech, Morocco, June 2-4, 2014. Proceedings 29. – Springer Berlin Heidelberg, 2014. – С. 220-233.

12. Niknami N., Inkrott E., Wu J. Towards analysis of the performance of idss in software-defined networks //2022 IEEE 19th International Conference on Mobile Ad Hoc and Smart Systems (MASS). – IEEE, 2022. – С. 787-793.

13. Басыня, Е. А. Комплексный мониторинг информационной инфраструктуры предприятия / Е. А. Басыня, Д. С. Худяков // Защита информации. Инсайд. – 2020. – № 6(96). – С. 28-33.

14. Morris K. Infrastructure as code: managing servers in the cloud. – O'Reilly Media, Inc., 2016.

15. Newham C. Learning the bash shell: Unix shell programming. – O'Reilly Media, Inc., 2005.

16. Tosi S. Matplotlib for Python developers. – Packt Publishing Ltd, 2009.