Подходы к представлению результатов анализа сетевого трафика*

[1] *А. И. Гетьман <thorin@ispras.ru>,*
[1] *Ю. В. Маркин <ustas@ispras.ru>,*
[1] *Д. О. Обыденков, <obydenkov@ispras.ru>,*
[1,2] *В. А. Падарян <vartan@ispras.ru>,*
[1] *А. Ю. Тихонов <fireboo@ispras.ru>*
[1] *Институт системного программирования РАН,*
109004, Россия, г. Москва, ул. А. Солженицына,д. 25
[2]*Московский государственный университет имени М.В. Ломоносова,*
119991 ГСП-1, Москва, Ленинские горы

Аннотация. В статье предложены различные способы представления результатов анализа сетевого трафика, необходимость в которых возникает прежде всего в задачах обеспечения сетевой информационной безопасности. Рассмотрена возможность построения полного графа сетевых взаимодействий, а также создания временной диаграммы передачи пакетов. Эти компоненты используются при расследовании инцидентов нарушения ИБ. Временная диаграмма также применяется при анализе туннельных протоколов, поскольку позволяет аналитику определить, какие именно заголовки протоколов необходимо визуализировать. Для задач, связанных с обратной инженерией, а также отладкой сетевых протоколов, предлагается использовать журнал, в котором фиксируются ошибки разбора заголовков протоколов. Представленные графические компоненты либо не имеют аналогов среди opensource-инструментов, либо улучшают уже существующие opensource-решения.

Ключевые слова: анализ сетевого трафика; отладка сетевых протоколов; граф сетевых взаимодействий; визуализация; журнал ошибок разбора.

DOI: 10.15514/ISPRAS-2016-28(6)-7

Для цитирования: Гетьман А.И., Маркин Ю.В. Обыденков Д.О., Падарян В.А., Тихонов А.Ю. Подходы к представлению результатов анализа сетевого трафика. Труды ИСП РАН, том 28, вып. 6, 2016, стр. 103-110. DOI: 10.15514/ISPRAS-2016-28(6)-7

* Работа поддержана грантом РФФИ 15-07-07652 А

1. Введение

Во многих задачах обеспечения сетевой информационной безопасности требуется детальный анализ сетевого трафика. Среди таких задач можно выделить:

- расследование инцидентов нарушения ИБ
- анализ сетевой обстановки
- обратная инженерия/отладка сетевых протоколов

Подобные задачи, как правило, решаются не "на потоке", а путем выделения некоторого фрагмента трафика с помощью программы-сниффера и его последующего анализа. При решении этих задач критическими факторами, влияющими на скорость и эффективность, являются наличие в среде анализа графических компонент, позволяющих визуализировать различные аспекты сетевых взаимодействий, и возможности по переключению и синхронизации между этими компонентами.

2. Обзор существующих средств анализа

Большинство популярных инструментов анализа сетевого трафика либо не имеют графического интерфейса (Snort [1], The Bro Network Security Monitor [2]), либо изначально разрабатывались для решения других задач и удовлетворяют указанным требованиям лишь частично [3]. Наиболее популярным инструментом из второй группы является инструмент Wireshark [4]. Основным средством представления сетевой трассы в нём являются разобранные пакеты в виде списка, при этом только для одного выделенного пакета отображается полный стек протоколов и значения полей в заголовках этих протоколов. Пакет, как элемент списка, представляется посредством строки, состоящей из значений фиксированного набора полей, выделенных в заголовке протокола сетевого уровня (IP-адреса), а также полей заголовка протокола самого высокого уровня, который удалось разобрать. Фиксированность представления пакетов может вызывать трудности во многих случаях. В частности, это проявляется при анализе туннельных протоколов. Так протокол GRE [5] (рис. 1) предназначен для инкапсуляции пакетов сетевого уровня модели OSI в IP-пакеты: сетевой пакет, таким образом, содержит два заголовка протокола IP. В списке пакетов инструмент Wireshark отобразит поля последнего (верхнего) IP-заголовка, который с точки зрения туннельного соединения не является репрезентативным: чтобы понять, какой хост из внутренней сети инициировал взаимодействие, аналитику придется выделять каждый пакет и проверять значение поля сетевого адреса нижележащего IP-заголовка.

Другим способом визуализации сетевых соединений в Wireshark является просмотр иерархий протоколов [6], присутствующих в трассе. Однако отображение осуществляется в виде общего дерева протоколов с указанием некоторой суммарной статистики соединений, но без возможности просмотра

и перехода к конкретным представителям соединений, соответствующих заданному типу вложенности протоколов, для просмотра параметров этих соединений и дальнейшего их анализа (рис. 2).

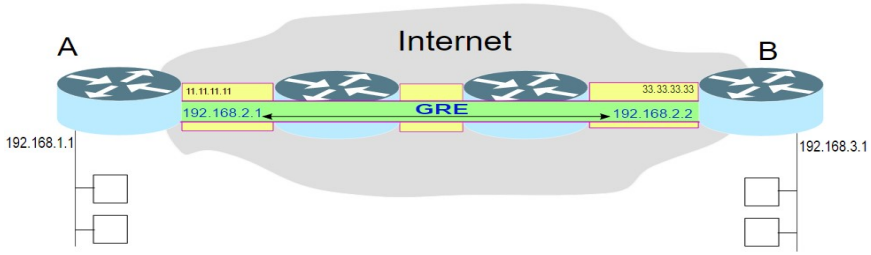

Рис. 1. Пример организации GRE-туннеля

Fig. 1. GRE tunneling scheme.

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
Frame	100.0	3000	100.0	2803776	160 k	0	0	0
Ethernet	100.0	3000	1.5	42000	2401	0	0	0
Internet Protocol Version 4	99.9	2996	2.1	59920	3426	0	0	0
User Datagram Protocol	34.0	1021	0.3	8168	467	0	0	0
Remote Procedure Call	33.9	1016	58.9	1651128	94 k	0	0	0
Network File System	33.9	1016	57.1	1602320	91 k	1016	1602320	91 k
Transmission Control Protocol	8.5	256	0.5	14244	814	103	3296	188
SSH Protocol	2.6	77	0.2	4752	271	77	4752	271
Rlogin Protocol	2.5	76	0.0	1300	74	76	1300	74
Stream Control Transmission Protocol	1.6	47	0.1	2540	145	33	1508	86
MTP 3 User Adaptation Layer	0.5	14	0.0	640	36	6	64	3
Signalling Connection Control Part	0.3	8	0.0	244	13	0	0	0
Malformed Packet	0.0	1	0.0	0	0	1	0	0
Internet Control Message Protocol	0.1	3	0.0	228	13	3	228	13
Data	56.0	1681	85.8	2406409	137 k	1681	2406409	137 k
Address Resolution Protocol	0.1	4	0.0	112	6	4	112	6

Рис. 2. Пример отображения вложенности взаимодействий в Wireshark.

Fig. 2. Wireshark's protocol hierarchy statistics.

3. Реализованные формы представления результатов анализа и их применение

При расследовании инцидента нарушения ИБ необходимо локализовать сетевые соединения, посредством которых этот инцидент возник и развивался во времени: аналитик должен обладать некоторым критерием (или множеством таких критериев) на содержимое сетевых пакетов. Одним из подходов к решению задачи локализации является представление сетевых взаимодействий посредством графа, в котором вершинам соответствуют стороны сетевого взаимодействия, а ребра отображают факт взаимодействия и возможно некоторые его характеристики, такие как интенсивность. При этом одна и та же сторона может участвовать сразу в нескольких взаимодействиях. Далее требуется провести детальный анализ выделенных соединений:

- проследить за порядком отправки/получения пакетов
- просмотреть значения полей интересующих протоколов

- восстановить данные протоколов прикладного уровня

Рассмотренные opensource-инструменты не предоставляют графических компонентов для работы с такими сценариями.

Предлагаемые компоненты представления результатов разбора опираются на модель описания данных [7], используемую ядром системы анализа, разрабатываемой в ИСП РАН. В отличие от Wireshark все сетевые пакеты (а не только выбранный пользователем) отображаются посредством дерева с выделенными заголовками инкапсулируемых протоколов (рис. 3). Таким образом, не возникает трудностей при работе с туннельными протоколами.

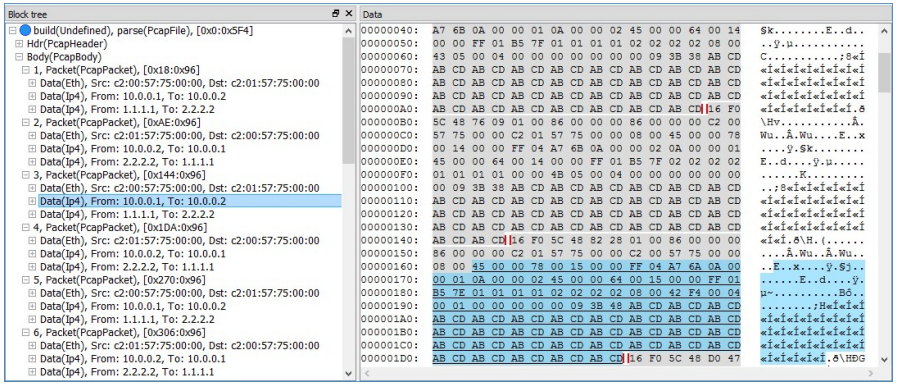

Рис. 3. Пример отображения стека протоколов для нескольких пакетов.

Fig. 3. ProtoSphere's protocol stack visualization for several packets.

Предлагается два способа визуализации сетевых взаимодействий:

- граф оконечных узлов (*Endpoints*)
- граф, детализирующий сетевые взаимодействия выбранного оконечного узла (*Nodes*)

Оба графа строятся по *дереву сетевых узлов*. Сетевой узел – это обобщение понятий отправителя и получателя для сетевых протоколов. Например, для протокола IPv4 сетевой узел описывает IP-адрес, тогда как для протокола TCP – порт. Вершина (сетевой узел) **B** дерева является дочерней по отношению к вершине **A**, если **B** характеризует отправителя (получателя) в заголовке некоторого протокола, вложенном (в рамках сетевого пакета) в заголовок нижележащего (согласно модели OSI) протокола, в котором выделен отправитель (получатель) **A**.

Граф *Endpoints* (рис. 4а) отображает сетевые соединения, относящиеся к протоколу самого низкого уровня в анализируемом файле. В качестве вершин здесь как правило выступают MAC- или IP-адреса. Ребра соединяют сетевые узлы, между которыми был передан хотя бы один сетевой пакет.

Граф *Nodes* (рис. 4б) детализирует сетевые взаимодействия заданного оконечного узла. При этом для каждого взаимодействия отображается весь стек сетевых протоколов. Дополнительно осуществляется фильтрация графа *Nodes* по содержимому сетевых пакетов, передаваемых между узлами.

Заметим, что комбинация графов *Endpoints* и *Nodes* также позволяет визуализировать результаты задачи расследования сетевой обстановки, когда по трафику требуется определить, какие сетевые службы (приложения) запущены на том или ином компьютере в сети.

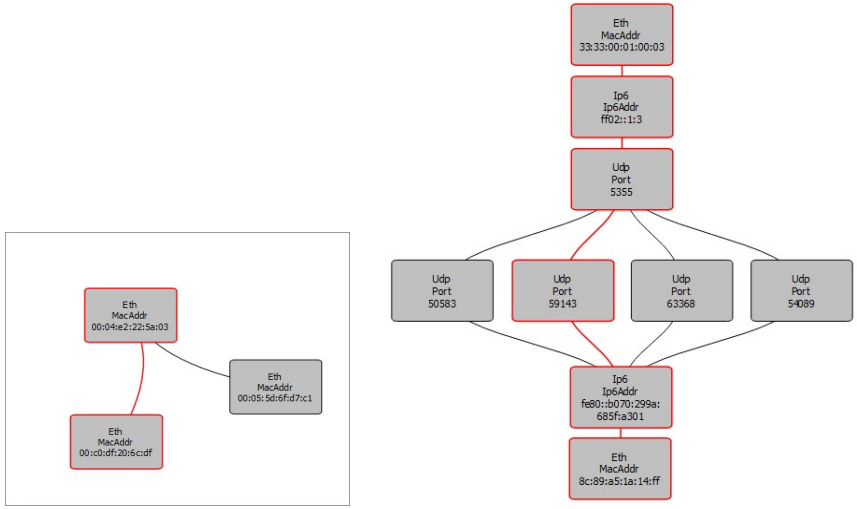

Рис. 4. (а) Пример графа Endpoints (б) Пример графа Nodes.

Fig. 4. (a) Endpoints graph (b) Nodes graph.

Для проведения детального анализа отдельного взаимодействия предлагается временна́я диаграмма, где каждый пакет отображается в виде стрелки с указанием отправителя и получателя (рис. 5). При этом можно указать, какие заголовки протоколов и какие поля в них должны отображаться над стрелками. Таким образом аналитик может адаптировать графический компонент под свои нужды. Следует отметить, что в анализаторе Wireshark есть аналогичный компонент [8], однако в нём отсутствует возможность настройки полей, значения которых будут отображаться для каждого пакета, что может быть неудобно, если интересующее поле отсутствует, или приводить к перегруженности отображения, если полей слишком много.

Вложенность взаимодействий с указанием параметров соединений отображается в дереве разбора (рис. 6). Именно в таком виде ядро инструмента хранит результаты анализа.

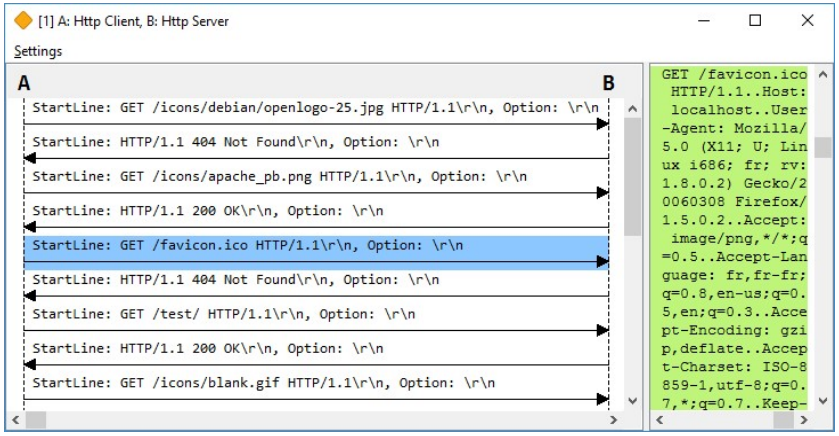

Рис. 5. Пример временно́й диаграммы.

Fig. 5. Time-based graph.

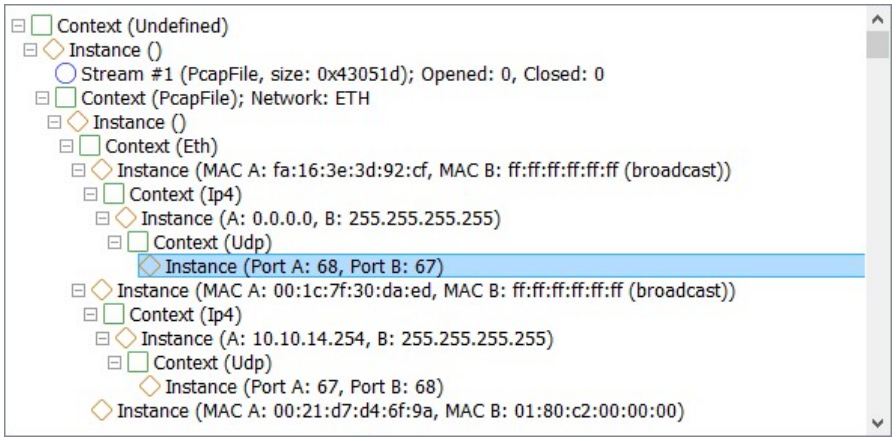

Рис. 6. Пример дерева разбора с отображением параметров соединений.

Fig. 6. ProtoSphere's protocol hierarchy statistics with display of connection settings.

При решении задач, связанных с обратной инженерией, а также отладкой сетевых протоколов, возникает необходимость в фиксации ошибок, возникающих при разборе заголовков этих протоколов. Анализаторы сетевого трафика, как правило, имеют модульную структуру: со временем появляются новые сетевые протоколы, и их необходимо поддерживать. Поддержка заключается в создании модуля, в котором локализована функциональность по работе с новым протоколом. При эксплуатации модуля могут проявляться ошибки разбора – несоответствия между кодом разборщика и данными, разбор которых осуществляется посредством данного разборщика. Журнал

ошибок, реализованный в системе, позволяет быстро локализовать такого рода ошибки. Ошибка разбора описывается текстовым сообщением и ссылкой на место ей возникновения – соответствующий пакет или сетевой сеанс. При работе с журналом поддерживается возможность фильтрации ошибок по протоколам.

Описанные графические компоненты синхронизированы друг с другом и допускают быстрое переключение между различными представлениями для повышения эффективности решения прикладных задач анализа сетевого трафика.

Список литературы

[1]. Snort. https://www.snort.org/, дата обращения: 10.10.2016
[2]. The Bro Network Security Monitor. https://www.bro.org/, дата обращения: 10.10.2016
[3]. Ю. В. Маркин, А. С. Санаров. Обзор современных инструментов анализа сетевого трафика. Препринты ИСП РАН, № 27, 2014
[4]. Wireshark. https://www.wireshark.org/, дата обращения: 10.10.2016
[5]. IETF RFC 2784. D. Farinacci, T. Li, S. Hanks, D. Meyer, P. Traina, Generic Routing Encapsulation, March 2000
[6]. The Protocol Hierarchy window. https://www.wireshark.org/docs/wsug_html_chunked/ChStatHierarchy.html, дата обращения: 10.10.2016
[7]. Гетьман А.И., Маркин Ю.В., Падарян В.А., Тихонов А.Ю.. Модель представления данных при проведении глубокого анализа сетевого трафика. Труды ИСП РАН, том 27, вып. 4, 2015 г., стр. 5-22. DOI: 10.15514/ISPRAS-2015-27(4)-1
[8]. Robert Shimonski. The Wireshark Field Guide: Analyzing and Troubleshooting Network Traffic. Elsevier Science & Technology Books, 2013, 128 p.

Methods of presenting the results of network traffic analysis★

[1]A. I. Get'man <thorin@ispras.ru>,
[1]Yu. V. Markin <ustas@ispras.ru>,
[1]D. O. Obydenkov <obydenkov@ispras.ru>,
[1,2]V. A. Padaryan <vartan@ispras.ru>,
[1]A. Yu. Tikhonov <fireboo@ispras.ru>
[1] Institute for System Programming of the Russian Academy of Sciences,
25, Alexander Solzhenitsyn st., Moscow, 109004, Russia
[2]Lomonosov Moscow State University,
GSP-1, Leninskie Gory, Moscow, 119991, Russian Federation

★ This work is supported by RFBR grant 15-07-07652 A

Abstract. The article proposes different methods of presenting network traffic analysis results, the need for which arises primarily in the area of network security. One of the most important tasks is to identify malicious traffic. For this purpose both the complete graph of network interactions and time-based packet diagram are presented. These components are used during investigation of information security violation incidents. The timing diagram is also used in analysis of tunneling protocols because it allows the analyst to determine which protocol headers are necessary to visualize. For tasks associated with reverse engineering and debugging of network protocols, it is proposed to use a journal which records protocol header parsing errors. Presented graphic components either have no analogues among the opensource tools or improve on existing opensource solutions.

Keywords: network traffic analysis, network protocols debugging, graph of network interactions, visualization, error log.

DOI: 10.15514/ISPRAS-2016-28(6)-7

For citation: Get'man A. I., Markin Yu. V., Obydenkov D. O., Padaryan V. A., Tikhonov A. Yu. Methods of presenting the results of network traffic analysis. *Trudy ISP RAN/Proc. ISP RAS*, vol. 28, issue 6, 2016, pp. 103-110 (in Russian). DOI: 10.15514/ISPRAS-2016-28(6)-7

References

[1]. Snort. https://www.snort.org/, accessed 10.10.2016
[2]. The Bro Network Security Monitor. https://www.bro.org/, accessed 10.10.2016
[3]. Yu. V. Markin, A. S. Sanarov. The modern network traffic analyzers overview. Preprinty ISP RAN [Preprints of ISP RAS], №27, 2014 (in Russian)
[4]. Wireshark. https://www.wireshark.org/, accessed 10.10.2016
[5]. IETF RFC 2784. D. Farinacci, T. Li, S. Hanks, D. Meyer, P. Traina, Generic Routing Encapsulation, March 2000
[6]. The Protocol Hierarchy window. https://www.wireshark.org/docs/wsug_html_chunked/ChStatHierarchy.html, accessed 10.10.2016
[7]. Get'man A. I., Markin Yu. V., Padaryan V. A., Tikhonov A. Yu. Model of data handling for in-depth analysis of network traffic. Trudy ISP RAN / Proc. ISP RAS, 2015, vol. 27, issue. 4, pp. 5-22 (in Russian). DOI: 10.15514/ISPRAS-2015-27(4)-1
[8]. Robert Shimonski. The Wireshark Field Guide: Analyzing and Troubleshooting Network Traffic. Elsevier Science & Technology Books, 2013, 128 p.