АРХИТЕКТУРА КРОССПЛАТФОРМЕННОГО DNS PROXY СЕРВИСА

DOI 10.24411/2072-8735-2018-10269

Подкорытов Дмитрий Александрович,
ФГБОУ ВО "Курганский государственный университет" (КГУ), г. Курган, Россия

Флока Анатолий Борисович,
ФГБОУ ВО "Курганский государственный университет" (КГУ), г. Курган, Россия

Кулешов Сергей Викторович,
СПИИРАН, г. г. Санкт-Петербург, Россия, ***kuleshov@iias.spb.su***

Работа выполнена в рамках реализации Государственно-го задания на 2019 г. № 0073-2019-0005

Ключевые слова: *LAN, WAN, архитектура сетевого сервиса, DNS трафик, среда разработки, UDP Proxy, DNS Proxy.*

Предлагается вариант построения архитектуры сетевого сервиса, выполняющего функции UDP Proxy. Архитектура основана на пуле из нескольких сетевых процессов, работающих совместно на одном сетевом адресе. Предметом исследования является архитектура построения сетевого сервиса, выполняющего функции UDP Proxy. Цель исследования заключается в разработке архитектуры экспериментального сервиса DNS Proxy, который с одной стороны обеспечивает сокрытие DNS трафика, а с другой не содержит криптографических преобразований, требующих серьёзных вычислительных затрат, не требует изменений на стороне клиента, является кроссплатформенным и компактным. Программная реализация DNS Proxy производится с использованием средств разработки, ориентированных на создание кроссплатформенного ПО и надежных высоконагруженных сетевых сервисов, выбраны языки Erlang, C/C++, D, рассмотрены особенности программной реализации. Для оценки производительности программной реализации DNS Proxy построен и сконфигурирован испытательный стенд на базе ОС Windows 7. Из нескольких вариантов реализации приложения DNS Proxy выбрана реализация на языке D, который с одной стороны обеспечивает компактность кода, а с другой – близок к языку C/C++ по эффективности. Предложенная архитектура DNS Proxy предполагает использование промежуточного коммуникационного слоя, внедряемого между клиентом и сервером и состоит из следующих компонентов: интерфейсов LAN, WAN и канала безопасной передачи данных по публичным сетям между ними. Рассматривается пример уязвимости "человек посередине", которая нейтрализуются за счет авторизации и преобразования трафика к внутреннему виду для транспорта между LAN и WAN компонентами сервиса.

Информация об авторах:

Подкорытов Дмитрий Александрович, *Федеральное государственное бюджетное образовательное учреждение высшего образования "Курганский государственный университет" (КГУ), старший преподаватель кафедры программного обеспечения автоматизированных системг, г. Курган, Россия*

Флока Анатолий Борисович, *Федеральное государственное бюджетное образовательное учреждение высшего образования "Курганский государственный университет" (КГУ); старший преподаватель кафедры программного обеспечения автоматизированных систем, к.т.н., г. Курган, Россия*

Кулешов Сергей Викторович, *Федеральное государственное бюджетное учреждение науки Санкт-Петербургский институт информатики и автоматизации Российской академии наук (СПИИРАН), главный научный сотрудник лаборатории автоматизации научных исследований, д.т.н., г. Санкт-Петербург, Россия*

Для цитирования:
Подкорытов Д.А., Флока А.Б., Кулешов С.В. Архитектура кроссплатформенного DNS Proxy сервиса // T-Comm: Телекоммуникации и транспорт. 2019. Том 13. №5. С. 35-40.

For citation:
Podkorytov D.A., Floka A.B., Kuleshov S.V. (2019). Cross-platform DNS Proxy service architecture. *T-Comm*, vol. 13, no.5, pp. 35-40. (*in Russian*)

В современном цифровом мире правовое поле меняется достаточно быстро, и ограничения диктуются не только технологическими возможностями, но и юридическими нормами. Так на использование технологии Virtual Private Network (VPN) в некоторых случаях стали накладываться ограничения [1]. Вместе с этим реалии сетевых и телекоммуникационных технологий таковы, что многие виды услуг предоставляются компаниями и сообществами бесплатно, а прибыль извлекается из того, что остается за полем зрения потребителя услуг. В качестве источников получения такого дохода можно привести рекламу, сбор статистики, аналитику и торговлю собранной и обработанной информацией о клиентах, фишинг и продвижение услуг заказчика в сфере информационных технологий.

Принятый в 2006 году Федеральный Закон «О персональных данных» фактически перечислил возникающие проблемы в сфере телекоммуникаций в правовой сфере, но не предложил механизмов решения этих проблем.

Рассмотрим организационно-технические аспекты работы сетевых сервисов в Интернет на примере доменной службы имен – DNS.

Доменная служба имен и технология проксирования

Доменная служба имен (DNS) является одним из ключевых компонентов инфраструктуры публичных сетей. С одной стороны, она позволяет осуществлять адресацию по символьным, мнемонически хорошо запоминаемым адресам, часто являющимся визитной карточкой и брендом, с другой стороны, она обеспечивает прозрачность адресации при доступе к ресурсу. Технология DNS дает возможность изменяться IP адресу, ассоциированному с доменным именем, обеспечивая постоянство доступа по доменному имени при смене IP адреса.

Вместе с этим, DNS является одним из старейших сервисов и несет в своей архитектуре ряд конструктивных особенностей, порождающих уязвимости концептуального характера:

1) Трафик DNS идет в открытом виде, что позволяет стороннему наблюдателю накапливать статистику (фишинг) и в некоторых случаях даже идентифицировать по собранным данным пользователя;

2) Некоторые провайдеры могут применять принудительное ограничение скорости для DNS-трафика (шейпинг). В результате, не смотря на скоростной канал связи с Интернет, создается дискомфорт для пользователя.

3) DNS подвержен атакам типа «человек посередине» (Man in the middle – MITM), что позволяет злоумышленнику не только пассивно накапливать статистику, но и заниматься активной подменой трафика между клиентом и сервером. Так по данным некоторых источников нужно только 10 секунд для того чтобы произвести накопление необходимой статистики и выполнить внедрение между клиентом и DNS сервером посредника, активно подменяющего трафик.

Некоторые проблемы были решены при внедрении стандарта DNSSEC – набора расширений протокола DNS, позволяющих минимизировать атаки, связанные с подменой DNS-адреса при разрешении доменных имён [2]. DNSSEC использует криптографические принципы для формирования электронной подписи каждого пакета и таким образом исключает внедрение активного посредника между клиентом и сервером DNS. К сожалению, DNSSEC имеет и ряд недостатков:

1) не предотвращает возможности отслеживания трафика сторонним наблюдателем, так как не выполняет его шифрование;

2) требует изменений в коде клиента для реализации поддержки стандарта DNSSEC;

3) приводит к увеличению интенсивности трафика в 6-7 раз;

4) требует периодической смены ключей шифрования на стороне сервера;

5) может быть предметом экспортных ограничений, так как в нем используются криптографические алгоритмы.

Одним из традиционных методов защиты от фишинга является проксирование (процесс обмена данным через прокси-сервер). Проксирование может быть реализовано различными способами, каждый из которых имеет свою сферу применения, а также свои достоинства и недостатки:

1) Технология VPN-сетей, проксирование на уровне Ethernet-пакетов.

Среди наиболее популярных реализаций следует выделить решения на базе OpenSSL, OpenVPN [3].

Достоинства подхода: проксирование осуществляется на сетевом уровне, инкапсуляции и шифрованию подлежит как правило весь трафик, шифрование канала связи исключает фишинг практически полностью.

Недостатки подхода: относительная сложность первичной настройки, необходимость периодической смены ключей, необходимость сертификации используемого программного или аппаратного продукта, использование ограниченного набора функций криптографических преобразований, для шифрования трафика требуются значительные вычислительные ресурсы, и иногда имеется задержка по времени.

Серьезным недостатком на данный момент являются также ограничения законодательства РФ на использование VPN решений.

2) Технология HTTP proxy, проксирование на прикладном уровне HTTP.

Достоинства подхода: кеширование траффика, маскирование адреса клиента.

Недостатки подхода: HTTP заголовки не шифруются при обмене данными, высокая потребность в вычислительных ресурсах, необходимость дублирования на стороне Proxy HTTP сессий пользователя. DNS трафик при этом не шифруется и оказывается доступен для фишинга и анализа третьей стороной.

3) Технология SOCKS 4/5 proxy, проксирование на уровне протоколов TCP и/или UDP.

Основным назначением SOCKS является возможность обеспечить работу клиент-серверных приложений за границами межсетевого экранирования, также технология позволяет подключение к сетевым ресурсам от внешнего клиента.

Достоинства подхода: возможность прозрачного шифрования канала связи, в том числе и HTTP заголовков, возможность шифрования UDP трафика.

Недостатки подхода: сложность настройки, ограниченная номенклатура ПО, ограниченное количество криптографических функций. Возможно наличие недокументированного скрытого функционала, заложенного разработчиком ПО и протокола.

4) Технология Web proxy

Достоинства подхода: наиболее доступный и легко используемый вид проксирования, большое количество ПО, высокий уровень совместимости.

Недостатки подхода: существует некоторое множество веб-ресурсов (сайтов), которые работают некорректно, велика вероятность сбора данных третьей стороной на стороне Proxy. В случае открытия зараженной вирусом страницы ее адрес беспрепятственно проходит через антивирусную защиту.

5) Технология Browser proxy

Некоторые браузеры (Opera) содержат внутри себя предустановленную возможность осуществлять проксирование через свой облачный сервис. Данная технология, являясь разновидностью технологии Web proxy, наследует соответствующие достоинства и недостатки Web proxy.

Достоинства подхода: простота первичной настройки, доступность.

Недостатки подхода: ориентация на proxy-сервисы конкретной компании.

Существуют и другие варианты частичного решения проблемы повышения защищенности и производительности сервиса DNS [4-12].

Архитектура DNS Proxy

Для устранения перечисленных выше недостатков была разработана архитектура экспериментального сервиса DNS Proxy, который с одной стороны обеспечивает сокрытие DNS трафика, а с другой не содержит криптографических преобразований, требующих серьёзных вычислительных затрат, не требует изменений на стороне клиента, является кроссплатформенным и компактным.

Рис. 1. Схема взаимодействия в традиционном DNS (а) и сервисе DNS Proxy (б)

Архитектура предлагаемого сервиса DNS Proxy строится на следующих принципах:

1) Совместное использование одного порта сетевого адреса несколькими различными экземпляра сервиса достигается за счет использования опции REUSEADDR открытого сокета и внутренней многопоточности приложения;

2) Принцип изоляции: параллельно работающие экземпляры сервиса выполняются каждый в своем адресном пространстве, взлом одного из них не влияет на другие;

3) Принудительное ограниченное время жизни процесса минимизирует эффект накопления ошибок, и нейтрализует успешные попытки взлома сетевого сервиса.

4) Наличие нескольких параллельно работающих экземпляров сервиса на одном порту обеспечивают бесперебойность сетевой работы сервиса;

5) Наличие супервизора, который следит за тем, чтобы количество параллельно работающих экземпляров сервиса было достаточным и при этом не превышало заданного порога, реализуя принцип языка Erlang: “Keep calm and let it crash” [13].

Наиболее близкими аналогами предлагаемого решения, ориентированными на DNS, являются DNSCrypt [14-15] и T-DNS [16-18].

Основными отличиями предлагаемого проекта DNS Proxy от DNSCrypt и T-DNS являются:

1) поддержка не только сервиса DNS, но и любых (в том числе пиринговых) UDP сервисов: SIP, NTP, Torrent;

2) кроссплатформенность и легкость компиляции на разных ОС;

3) небольшой объем программного кода, позволяющий осуществить его анализ и верификацию;

4) поддержка неограниченного количества серверов, в отличие от DNSCrypt [19];

5) возможность использования любого ПО для реализации DNS-сервера, в отличие от технологии T-DNS, которая требует внедрения в тело существующего DNS сервера модуля, обеспечивающего транспорт через TLS, что ограничивает множество совместимого ПО на стороне сервера DNS [20–21].

6) не требует процедуры генерации ключей и их периодической смены, в отличие как от DNSCrypt, так и от T-DNS.

Программная реализация DNS Proxy

Выбор средств разработки для реализации DNS Proxy определялся ориентированностью на создание кроссплатформенного ПО и надежных высоконагруженных сетевых сервисов. Среди перспективных средств разработки были выбраны языки Erlang, C/C++, D. Рассмотрим особенности программной реализации с использованием перечисленных языков программирования.

1) Для реализации первой версии DNS Proxy был выбран язык Erlang, который хорошо зарекомендовал себя как язык быстрого прототипирования и создания надежных сетевых сервисов. Для повышения скорости работы сервиса и обеспечения юридической чистоты криптографические преобразования не использовались.

2) Вторая версия проекта была реализована на языке программирования C/C++ и ориентирована на компактность и надежность кода с целью его дальнейшего анализа и рефакторинга. Объём исходного кода на C/C++ составил порядка 7Kb. Многопоточность реализуется в виде пула работающих параллельно процессов ОС. Каждый процесс работает в сво-

ем независимом адресном пространстве ОС, совместное использование сокета процессами обеспечено опцией режима сокета REUSEADDR. Для обеспечения бесперебойности работы сервиса все процессы контролируются супервизором, который следит за тем, чтобы количество работающих в ОС процессов сервиса было строго заданным. Если какой-либо процесс завершает работу, происходит его перезапуск супервизором из контекста другого параллельного процесса.

Благодаря использованию супервизора становится возможным ограничить время жизни каждого процесса: количеством обработанных сообщений и заданным временем (обычно несколько секунд). По достижении этих ограничений процесс заканчивается и перезапускается из контекста супервизора другого процесса. Такой подход способен усложнить злоумышленникам взлом и компрометацию сервиса.

Отсутствие анализа DNS трафика в этой версии позволяет использовать проект для проксирования любого UDP трафика, например, в задачах IP-телефонии (SIP).

3) При дальнейшем развитии проекта основной целью стало решение задачи переносимости между различными ОС, что не смог обеспечить язык C/C++. Язык D был выбран как язык программирования с одной стороны близкий к C/C++ по эффективности, а с другой стороны обеспечивающий компактность программного кода. Данная версия поддерживает авторизацию, используя REST подход, что позволяет не передавать пароль по сети в открытом виде.

Режим работы сервиса с выключенной авторизацией работает при использовании модуля udp_pipe, тогда как модули udp_lan_wan и udp_wan_lan применяются на стороне клиента и на стороне сервиса с включенной авторизацией. Многопоточность обеспечивается использованием механизма нитей языка D.

Дальнейшее развитие сервиса и связанное с этим увеличение нагрузки, потребует:

– использования иных схем авторизации, рассчитанных на большое количество пользователей и использующих отработанные отраслевые решения,

– применения сертифицированных средств криптографии (при необходимости),

– использования отдельных «балансировщиков» нагрузки,

– поддержки параллельной работы с несколькими адресами связи как со стороны локальной, так и со стороны глобальной сетей,

– реализации механизмов фильтрации и предотвращения DDoS-атак.

В случае использования сервиса для проксирования DNS потребуется разработка DNS сервера с кешированием,

Тестирование и оценка результатов

Для оценки производительности программной реализации DNS Proxy был сконфигурирован испытательный стенд на базе ОС Windows 7. Сервис был размещен на сетевом адресе 127.0.0.1. Режимы LAN (локальная сеть) и WAN (глобальная сеть) сопрягались через сетевой интерфейс петли (looback) и располагались на одном хосте. Обращение через LAN и WAN производилось к серверу DNS (программная реализация BIND9), находящемуся в пределах локальной сети с одним сетевым интерфейсом в 1 ГБит. Для измерения численных характеристик использовалось инструментальное средство dnsperf [22].

На языке программирования D были реализованы следующие варианты приложения DNS proxy:

1) однопоточное приложение с пулом сокетов для выполнения запросов к удаленному сервису. Выбор номера сокета выполнялся с помощью наложения битовой маски размером с пул сокетов (256 сокетов) на текущий индекс в цикле обработки сообщений. Время таймаута запроса подбиралось экспериментально для максимизации скорости при сохранении потерь на уровне 0.3-0.5%. В этом варианте упор сделан на кеширование в ОС операций с сокетами и минимизации таймаутов распределением запросов по пулу. Данный вариант архитектуры показал наилучшие скоростные характеристики, порядка 3000 запросов в секунду. Скорость работы сервера BIND9 была порядка 9000–10000 запросов в секунду;

2) многопоточное приложение, в котором на каждый входящий запрос создается новая асинхронная нить, возвращающая из своего контекста результат клиенту и завершающая работу. Скорость работы в этой архитектуре обеспечивалась на уровне 600 запросов в секунду. Понижение скорости работы по отношению к первому варианту можно объяснить накладными расходами на создание каждой новой нити;

3) расширение архитектуры первого типа пулом нитей фиксированного размера и коммуникацию с этими нитями, реализованную с помощью обмена сообщениями между акторами языка D. Одному из акторов при поступлении DNS запроса пересылалась все необходимые данные для выполнения запроса, после чего актор работал асинхронно, после чего возвращал асинхронно ответ в сокет, принимающий сообщения. Синхронизация коммуникаций выполнялась средствами ОС в неявном для программиста виде. Использование пулов акторов размеров в 16 и 256, показало результаты близкие по скорости работы, обеспечивая быстродействие на уровне 1300-1900 запросов в секунду.

Данная архитектура не оправдала гипотезу повышения производительности из-за больших накладных расходов на синхронизацию средствами ОС и сокета;

4) приложение с переключаемой многозадачностью на D Fibers. Данный вариант архитектуры показал скоростные характеристики, схожие с первым типом архитектуры на пуле сокетов.

Заключение

Предложенная архитектура DNS Proxy предполагает использование промежуточного коммуникационного слоя, внедряемого между клиентом и сервером и состоит из следующих компонентов: интерфейсов LAN, WAN и канала безопасной передачи данных по публичным сетям между ними. При этом уязвимости типа «человек посередине» нейтрализуются за счет авторизации и преобразования трафика к внутреннему виду для транспорта между LAN и WAN компонентами сервиса.

Программные реализации для предложенной архитектуры выполнены на нескольких языках программирования при обеспечении кроссплатформенности и протестированы как на стресс-тестах (производилось определение максимальной

производительности в реальных условиях и в условиях перегрузок), так и для получения оценок быстродействия.

Дальнейшие направления работы по оптимизации скорости работы сервиса могут заключаться в следующем:

– оценка изменения скорости работы при асинхронной работе с сокетами с помощью функции select;

– оценка возможности синхронизации ответов в явном виде, не полагаясь на механизмы сокетов ОС;

– разделение сокетов для приема запросов от клиентов и ответа на них, что позволит изолировать кэш сокетов для запросов и ответов, что позволит входящим запросам не конкурировать с исходящими;

– оценка возможности использования для коммуникации с пулом задач сокетов UNIX, а не встроенного механизма пересылки сообщений акторам.

Работа выполнена в рамках реализации Государственного задания на 2019 г. № 0073-2019-0005

Литература

1. Новости Роскомнадзора // Электронный ресурс: Доступ — https://rkn.gov.ru/news/rsoc/news51440.htm (дата обращения 12.03.2019).

2. DNSSEC: DNS Security Extensions Securing the Domain Name System // Электронный ресурс: Доступ — https://www.dnssec.net/ (дата обращения 12.03.2019).

3. Страница проекта OpenVPN // Электронный ресурс: Доступ — https://openvpn.net/ (дата обращения 12.03.2019).

4. V.V. Alexandrov, S.V. Kuleshov and A.A. Zaytseva Active Data in Digital Software Defined Systems Based on SEMS Structures. // Logical Analysis of Data and Knowledge with Uncertainties in SEMS — A.E. Gorodetskiy (ed.), Smart Electromechanical Systems, Studies in Systems, Decision and Control 49, 2016, pp. 61-69.

5. OpenDNS repository // Электронный ресурс: Доступ — https://github.com/opendns (дата обращения 12.03.2019).

6. DNS Privacy Project Homepage // Электронный ресурс: Доступ — https://dnsprivacy.org/wiki/ (дата обращения 12.03.2019).

7. DNS Privacy — Current Work // Электронный ресурс: Доступ — https://dnsprivacy.org/wiki/display/DP/DNS+Privacy+-+Current+Work (дата обращения 12.03.2019).

8. DNS Privacy — The Solutions // Электронный ресурс: Доступ — https://dnsprivacy.org/wiki/display/DP/DNS+Privacy+-+The+Solutions (дата обращения 12.03.2019).

9. M. Dempsky DNSCurve: Link-Level Security for the Domain Name System // Электронный ресурс: Доступ — https://tools.ietf.org/html/draft-dempsky-dnscurve-01 (дата обращения 12.03.2019).

10. C. Huitema, M. Shore, etc Specification of DNS over Dedicated QUIC Connections // June 29, 2018 // Электронный ресурс: Доступ — https://datatracker.ietf.org/doc/draft-huitema-quic-dnsoquic/ (дата обращения 15.02.2019).

11. DNS Analysis and Privacy Tools // Электронный ресурс: Доступ — https://ant.isi.edu/software/tdns/index.html (дата обращения 12.03.2019).

12. DNS over HTTPS // Электронный ресурс: Доступ — https://developers.google.com/speed/public-dns/docs/dns-over-https (дата обращения 12.03.2019).

13. S. St. Laurent. Introducing Erlang. O'Reilly Media, 2017, 202 с.

14. DNSCrypt project // Электронный ресурс: Доступ — https://dnscrypt.info/ (дата обращения 12.03.2019).

15. Утечка DNS: что это такое и как ее устранить с помощью утилиты DNSCrypt // Электронный ресурс: Доступ — http://www.spy-soft.net/utechka-dns/ (дата обращения 12.03.2019).

16. T-DNS server proxy // Электронный ресурс: Доступ — https://ant.isi.edu/software/tdns/tdns-server-proxy/index.html (дата обращения 12.03.2019).

17. T-DNS client proxy // Электронный ресурс: Доступ — https://ant.isi.edu/software/tdns/tdns-client-proxy/index.html (дата обращения 12.03.2019).

18. Liang Zhu, Zi Hu, John Heidemann, etc. T-DNS: Connection-Oriented DNS to Improve Privacy and Security // USC/ISI Technical Report ISI-TR-688, Feb. 2014 URL: https://www.isi.edu/~johnh/PAPERS/Zhu14a.pd (дата обращения 12.03.2019).

19. How to Boost Your Internet Security with DNSCrypt // Электронный ресурс: Доступ — https://lifehacker.com/how-to-boost-your-internet-security-with-dnscrypt-510386189 (дата обращения 12.03.2019).

20. DNS over TLS // Электронный ресурс: Доступ — https://support.opendns.com/hc/en-us/community/posts/115019265903-DNS-over-TLS (дата обращения 12.03.2019).

21. Android getting "DNS over TLS" support to stop ISPs from knowing what websites you visit // Электронный ресурс: Доступ — https://www.xda-developers.com/android-dns-over-tls-website-privacy/ (дата обращения 15.02.2019).

22. DNSPerf — DNS Speed Benchmark // Электронный ресурс: Доступ — https://www.dnsperf.com/ (дата обращения 11.02.2019).

CROSS-PLATFORM DNS PROXY SERVICE ARCHITECTURE

Dmitry A. Podkorytov, *Kurgan State University, Kurgan, Russia*
Anatoly B. Floka, *Kurgan State University, Kurgan, Russia*
Sergey V. Kuleshov*, Saint-Petersburg Institute for Informatics and Automation of Russian Academy of Science, Saint-Petersburg, Russia,*
kuleshov@iias.spb.su

Abstract
The article proposes the variant of building a network service architecture that performs the functions of UDP Proxy. The architecture is based on a pool of several network processes working together on a single network address. The subject of study is the architecture of a network service development that performs the functions of UDP Proxy. The purpose of the study is to develop an experimental DNS Proxy service architecture, which, on the one hand, provides for hiding DNS traffic, and on the other hand, does not contain cryptographic transformations that require significant computational costs, does not require changes on the client side. The proposed experimental DNS Proxy service is cross-platform and compact. For the software implementation of the DNS Proxy we use the development tools focused on creating cross-platform software and reliable high-load network services. The languages Erlang, C/C++, D are selected, the features of the software implementation are considered. To evaluate the performance of the software implementation of the DNS Proxy, a test bench based on the Windows 7 operating system is built and configured. We choose the implementation in D, which on the one hand provides compact code, and on the other is close to C/C++ by efficiency. The proposed DNS Proxy architecture assumes the use of an intermediate communication layer that is implemented between the client and the server and consists of the following components: LAN, WAN interfaces and a secure data transmission channel over public networks between them. We consider an example of a man-in-the-middle vulnerability that is neutralized by authorizing and converting traffic to an internal view for transport between the LAN and WAN components of the service.

Keywords: *LAN, WAN, network service architecture, DNS traffic, IDE, UDP Proxy, DNS Proxy.*

References

1. Roskomnadzor news // URL: https://rkn.gov.ru/news/rsoc/news51440.htm (Data Access 12.03.2019) (*in Russian*)
2. DNSSEC: DNS Security Extensions Securing the Domain Name System // URL: https://www.dnssec.net/ (Data Access 12.03.2019).
3. OpenVPN // URL: https://openvpn.net/ (Data Access 12.03.2019).
4. V.V. Alexandrov, S.V. Kuleshov and A.A. Zaytseva (2016). Active Data in Digital Software Defined Systems Based on SEMS Structures. Logical Analysis of Data and Knowledge with Uncertainties in SEMS - A.E. Gorodetskiy (ed.), *Smart Electromechanical Systems, Studies in Systems, Decision and Control* 49, pp. 61-69
5. OpenDNS repository // URL: https://github.com/opendns (Data Access 12.03.2019).
6. DNS Privacy Project Homepage // URL: https://dnsprivacy.org/wiki/ (Data Access 12.03.2019).
7. DNS Privacy - Current Work // URL: https://dnsprivacy.org/wiki/display/DP/DNS+Privacy+-+Current+Work (Data Access 12.03.2019).
8. DNS Privacy - The Solutions // URL: https://dnsprivacy.org/wiki/display/DP/DNS+Privacy+-+The+Solutions (Data Access 12.03.2019).
9. M. Dempsky DNSCurve: Link-Level Security for the Domain Name System // URL: https://tools.ietf.org/html/draft-dempsky-dnscurve-01 (Data Access 12.03.2019).
10. C. Huitema, M. Shore, etc Specification of DNS over Dedicated QUIC Connections // June 29, 2018 // URL: https://datatracker.ietf.org/doc/draft-huitema-quic-dnsoquic/ (Data Access 15.02.2019).
11. DNS Analysis and Privacy Tools // URL: https://ant.isi.edu/software/tdns/index.html (Data Access 12.03.2019).
12. DNS over HTTPS // URL: https://developers.google.com/speed/public-dns/docs/dns-over-https (Data Access 12.03.2019).
13. S. St. Laurent. Introducing Erlang. O'Reilly Media, 2017, 202 p.
14. DNSCrypt project // URL: https://dnscrypt.info/ (Data Access 12.03.2019).
15. DNS leak: what it is and how to fix it with the DNSCrypt utility // URL: http://www.spy-soft.net/utechka-dns/ (Data Access 12.03.2019) (In Russ)
16. T-DNS server proxy // URL: https://ant.isi.edu/software/tdns/tdns-server-proxy/index.html (Data Access 12.03.2019).
17. T-DNS client proxy // URL: https://ant.isi.edu/software/tdns/tdns-client-proxy/index.html (Data Access 12.03.2019).
18. Liang Zhu, Zi Hu, John Heidemann, etc. T-DNS: Connection-Oriented DNS to Improve Privacy and Security // USC/ISI Technical Report ISI-TR-688, Feb. 2014 URL: https://www.isi.edu/~johnh/PAPERS/Zhu14a.pd (Data Access 12.03.2019).
19. How to Boost Your Internet Security with DNSCrypt // URL: https://lifehacker.com/how-to-boost-your-internet-security-with-dnscrypt-510386189 (Data Access 12.03.2019).
20. DNS over TLS // URL: https://support.opendns.com/hc/en-us/community/posts/115019265903-DNS-over-TLS (Data Access 12.03.2019).
21. Android getting "DNS over TLS" support to stop ISPs from knowing what websites you visit // URL: https://www.xda-developers.com/android-dns-over-tls-website-privacy/ (Data Access 15.02.2019).
22. DNSPerf - DNS Speed Benchmark // URL: - https://www.dnsperf.com/ (Data Access 11.02.2019).

Information about authors:

Dmitry A. Podkorytov, *Kurgan State University, Senior Lecturer, Department of Software, Automated Systems, Kurgan, Russia*

Anatoly B. Floka, *Kurgan State University, Senior Lecturer, Department of Software, Automated Systems, PhD. Tech., Kurgan, Russia*

Sergey V. Kuleshov, *Saint-Petersburg Institute for Informatics and Automation of Russian Academy of Science; Principal Researcher, Laboratory for Research Automation; Dr. of Tech. Sc., Saint-Petersburg, Russia*