Internet Engineering Task Force (IETF)
Request for Comments: 8993
Category: Informational
ISSN: 2070-1721

M. Behringer, Ed.

B. Carpenter
Univ. of Auckland
T. Eckert
Futurewei USA
L. Ciavaglia
Nokia
J. Nobre
UFRGS

May 2021

A Reference Model for Autonomic Networking

Эталонная модель для сетей с самоуправлением

Аннотация

В этом документе описана эталонная модель для автоматической работы[1] (Autonomic Networking или AN) управляемой сети. Определено поведение автоматического узла, совместная работа элементов в контексте самоуправления и использование инфраструктуры автоматическими службами.

Статус документа

Документ не содержит какой-либо спецификации (Internet Standards Track) и публикуется с информационными целями.

Документ не связан с другими RFC и выбран для публикации редактором (RFC Editor) по своему усмотрению без каких-либо заявлений о ценности документа для внедрения или развёртывания. Документы, одобренные для публикации RFC Editor, не претендуют на статус стандартов Internet (см. раздел 2 в RFC 7841).

Информация о текущем статусе документа, найденных ошибках и способах обратной связи доступна по ссылке https://www.rfc-editor.org/info/rfc8993.

Оглавление

[1] В переводе используется принятая в русском языке терминология для автоматических и автоматизированных узлов (систем, элементов). Автоматическая система работает без привлечения человека или внешней системы управления, автоматизированная просто выполняет задание (сценарий), заданный человеком или внешней системой управления. Термин «самоуправляемый» в переводе используется как синоним термина «автоматический». *Прим. перев.*

1. Введение

В документе «Autonomic Networking: Definitions and Design Goals» [RFC7575] описаны фундаментальные концепции, лежащие в основе автономных сетей (Autonomic Networking), определены термины и высокоуровневая эталонная модель. В [RFC7576] рассматривается «разрыв» между традиционным подходом и автономизацией.

В этом документе эталонная модель рассматривается более подробно, чтобы на её основе можно было создавать архитектурно согласованные спецификации функций и протоколов.

Как отмечено в [RFC7575], цель работы не заключается в рассмотрении лишь полностью автоматических узлов или сетей. На деле большинство автоматических сетей будет работать с некоторыми автоматическими функциями, тогда как в остальной части сети будет применяться традиционное управление. Эталонная модель позволяет такой подход.

Например, в имеющейся неавтоматической сети можно регистрировать устройства традиционным способом для создания доверенной инфраструктуры на основе сертификатов. Эту доверенную инфраструктуру затем можно применить для активизации автоматической плоскости управления (Autonomic Control Plane или ACP) и выполнения традиционных сетевых операций через защищённую и самовосстанавливающуюся ACP (см. [RFC8368]).

Область действия описываемой модели ограничена сетями, которые в той или иной степени управляются квалифицированными операторами, - такие сети ещё называют профессионально управляемыми. В неуправляемых сетях возникают дополнительные вопросы защиты и доверия, которые данная модель не охватывает.

В этом документе описана первая фаза решения для автоматических сетей (Autonomic Networking или AN), которая проста и реализуема. Предполагается, что обретённый на этом этапе опыт будет использован со временем для создания обновлённых и расширенных спецификаций. Некоторые вопросы рассматриваются в документе с архитектурной точки зрения, но ещё не отражены в спецификациях реализаций. Соответствующие параграфы документа помечены звёздочкой (*).

2. Представление сети

В этом разделе рассматриваются элементы в сети с автоматическими функциями и разъясняется, как эти элементы совместно работают на высоком уровне. В последующих разделах подробно рассматривается внутреннее представления каждого элемента AN, а также сетевые функции (или интерфейсы) между элементами.

На рисунке 1 показано высокоуровневое представление сети с самоуправлением (AN). Сесть состоит из множества автоматических узлов, которые взаимодействуют между собой напрямую. Эти автоматические узлы обеспечивают в сети общий набор возможностей (свойств), который называется инфраструктурой автоматической сети (Autonomic Networking Infrastructure или ANI). Инфраструктура ANI обеспечивает такие функции, как именование, адресация, синхронизация, обнаружение и обмен сообщениями.

Автоматические функции обычно охватывают несколько узлов сети (возможно все). Неделимые элементы автоматической функции называют агентами автоматических служб (Autonomic Service Agent или ASA), их экземпляры создаются на узлах.

По горизонтали автоматические функции охватывают всю сеть, а также ANI. По вертикали сеть всегда реализует ANI и может включать один или несколько агентов ASA, которые могут быть автономными или включать в себя другие ASA (иерархия). Таким образом, ANI служит основой для автоматических функций.

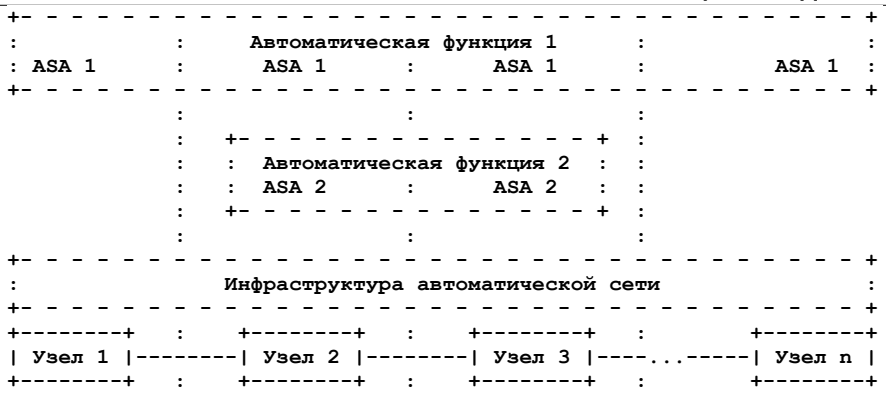

Рисунок 1. Высокоуровневое представление сети с самоуправлением.

3. Самоуправляемый элемент сети

В этом разделе описана общая архитектура элементов AN (параграф 3.1), отслеживание ими окружающей среды в таблице смежности (параграф 3.2) и конечный автомат, определяющий поведение элемента сети (параграф 3.3) на основе таблицы смежности.

3.1. Архитектура

В этом параграфе описывается элемент AN и его внутренняя архитектура. Эталонная модель из «Autonomic Networking: Definitions and Design Goals» [RFC7575] показывает источники информации, которые может использовать ASA - самообучение, изучение сети (с помощью обнаружения), Intent (см. параграф 7.2) и контуры обратной связи. Внутри автоматического узла имеется два уровня - уровень ASA и уровень ANI, причём первый использует услуги второго. Эта концепция показана на рисунке 2.

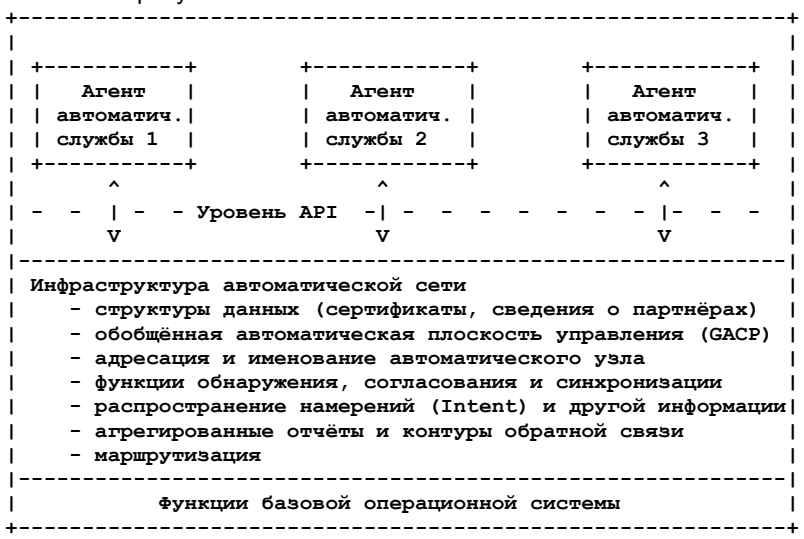

Рисунок 2. Модель автоматического узла.

Инфраструктура ANI (нижняя часть рисунка 2) содержит зависящие от узла структуры данных (например, сведения о доверии для самого узла и его партнёров), а также базовый набор функций, не зависящий от конкретного применения. Этой инфраструктуре следует быть общей и поддерживать разные ASA (верхняя часть рисунка 2). ANI включает адресацию и именование автоматических узлов, распространение информации, передачу отчётов, контуры обратной связи и маршрутизацию внутри плоскости управления ACP.

Обобщённая плоскость управления (Generalized ACP или GACP) - это сводка всех взаимодействий ANI с другими узлами и службами. Конкретная реализация GACP в этом документе обозначается ACP и описана в [RFC8994].

Варианты использования автоматики (самоуправления, самооптимизация и т. п.) реализованы как элементы ASA. Они используют службы и структуры данных базовой инфраструктуры ANI, которой следует быть самоуправляющейся.

Функции базовой ОС (нижняя часть рисунка 2) включают обычную ОС (например, сетевой стек и функции защиты).

Полные узлы автоматической сети AN имеют полную инфраструктуру ANI со всей функциональностью, описанной в этом документе. Позднее рабочая группа ANIMA может определить область действия ограниченных узлов с сокращённой ANI и чётко заданной минимальной функциональностью. Сейчас это не рассматривается.

3.2. Таблица смежности

Самоуправляемые сети AN основаны на прямом взаимодействии между узлами домена. Плоскость управления ACP обычно строится на пошаговой основе (hop-by-hop), поэтому многие взаимодействия в ANI основываются на таблице смежности. Некоторые взаимодействия вносят сведения в эту таблицу, а другие используют данные из таблицы.

Таблица смежности ANI содержит по меньшей мере сведения о смежных автоматических узлах: Node-ID, IP-адрес в плоскости данных, IP-адрес в ACP, домен и сертификат. Автоматический узел поддерживает актуальность сведений в таблице. Таблица содержит сведения лишь об узлах, поддерживающих AN, а узлы без самоуправления обычно не отслеживаются в таблице. Однако информация отслеживается независимо от статуса партнёров, в частности, таблица смежности содержит сведения о незарегистрированных узлах того же или иного домена. Таблица смежности может включать сведения о действительности смежных автоматических узлов и уровне доверия к ним.

В таблицу смежности поступают указанные ниже данные.

- Обнаружение локальных соединений (Link-local). Это взаимодействие происходит в плоскости данных с использованием лишь адресации IPv6 link-local, поскольку этот тип адресации сам является автоматическим. Этот способ позволяет узлу изучить автоматические узлы вокруг себя. Процедуру обнаружения локальных соединений описывают документы [RFC8990], [RFC8995], [RFC8994].

- Перенаправление от производителя. Новое устройство может получать информацию о местоположении его домашней сети через перенаправление от уполномоченного производителем удостоверяющего центра (Manufacturer Authorized Signing Authority или MASA), как указано в параграфе 5.3. Обычно это маршрутизируемый адрес.

- Неавтоматический ввод. Узел можно настроить вручную с участием автоматического партнёра, о котором узел может узнать из опций DHCP, DNS или иных неавтоматических механизмов. Обычно такие механизмы требуют того или иного участия администратора. Основной целью является обход (bypass) неавтоматического устройства или сети. Для новых устройств этот вопрос рассматривается в Приложениях A и B к [RFC8995].

Таблица смежности определяет поведения узла с самоуправлением, как описано ниже.

- Если узел не загрузился (bootstrap) в домен (т. е. не имеет сертификата домена), он проходит один за другим все узлы в таблице смежности, заявляющие присутствие в домене, и пытается загрузиться через них. Одним из возможных откликов является перенаправления через MASA производителя, которое будет введено в таблицу смежности (см. выше и [RFC8995]).

- Если смежный узел относится к тому же домену, он аутентифицирует данный узел и при успешной проверке подлинности организует ACP (см. [RFC8994]).

- Как только узел становится частью ACP в домене, он будет использовать GRASP [RFC8990] для поиска регистраторов домена и, возможно, других служб.

- Если узел является частью ACP и нашёл хотя бы один регистратор в домене с помощью GRASP, он запускает ASA и будет выступать как прокси присоединения для соседей, которым нужна загрузка (см. параграф 6.3.1.2).

- Возможно иное поведение, например, организация ACP с устройствами субдомена или других доменов. Это скорей всего будет контролироваться Intent, но этот вопрос выходит за рамки документа. Отметим, что Intent распространяется через ACP, поэтому узел может адаптировать управляемое Intent поведение лишь после присоединения к ACP. В настоящее время рабочая группа ANIMA не рассматривает предоставление Intent вне ACP, но это может быть рассмотрено позднее.

После присоединения узла к ACP он также узнает адреса ACP смежных узлов и добавит их в таблицу смежности для обеспечения коммуникаций внутри ACP. Последующие взаимодействия автоматических доменов будут выполняться внутри ACP. В настоящее время определены лишь согласование и синхронизация через GRASP [RFC8990]. GRASP работает в плоскости данных как источник сведений для построения таблицы смежности, а также внутри ACP.

Автоматические функции состоят из ASA. Они работают логически на базе инфраструктуры ANI и могут использовать таблицу смежности, ACP, согласование и синхронизацию через GRASP в ACP, Intent и другие функции ANI. Поскольку ANI обеспечивает лишь автоматическое взаимодействие внутри домена, автоматические функции могут также использовать любой другой контекст на узле, в частности, глобальную плоскость данных.

3.3. Конечный автомат

AN применяется в течение всего жизненного цикла узла. В этом параграфе описан конечный автомат для состояний, которые проходит самоуправляющийся узел в течение своей жизни.

Обычно предполагается, что устройство сохраняет связанное с доменом отождествление - Local Device Identifier (LdevID, см. параграф 5.2) - в постоянном хранилище, которое будет доступно после выключения и последующего включения питания. Для устройств, не способных постоянно сохранять LDevID выключения питания равносильно сбросу к заводским настройкам.

3.3.1. Состояние 1 - заводские установки

Автоматический узел выпускается с завода в этом состоянии. Узел не имеет связанных с доменом настроек, в частности LDevID, и может использоваться в любой конкретной сети. Однако узел имеет заданный производителем идентификатор (Initial Device Identifier или IDevID) [IDevID]. Узлы без IDevID невозможно автоматически и безопасно зарегистрировать в домене, они требуют предварительной настройки вручную и в этом случае подготовка начинается из состояния 2.

Переходы

- Загрузка. Устройство регистрируется в домене, получая при этом доменное отождествление - LDevID. При успешной регистрации следующим будет состояние 2. регистрация описана в [RFC8995].

- Включение и выключение питания. Устройство теряет все таблицы состояний и остаётся в состоянии 1.

3.3.2. Состояние 2 - устройство зарегистрировано

Автоматический узел находится в зарегистрированном (enrolled) состоянии, если у него имеется LdevID и в настоящее время нет канала ACP. Узел может иметь дополнительную конфигурацию или состояние, если он находился, например, в состоянии 3, но потерял все свои каналы ACP. Идентификатор LDevID можно удалить с устройства только при сбросе к заводским настройкам и при этом будут удалены все прочие состояния устройства. Это гарантирует отсутствие устаревшего доменного состояния при регистрации из состояния 1.

Переходы

- Присоединение к ACP. Устройство организует канал ACP к смежному устройству (см. [RFC8994]). Следующим будет состояние 3.
- Сброс к заводским настройкам. Удаляются все конфигурации и доменное отождествление LdevID. Следующим будет состояние 1.
- Включение и выключение питания. Устройство теряет все таблицы состояния, но сохраняет доменное отождествление LdevID и остаётся в состоянии 2.

3.3.3. Состояние 3 - устройство подключено к ACP

В этом состоянии автоматический узел имеет хотя бы 1 канал ACP к другому устройству. Узел теперь может участвовать в других автоматических транзакциях, таких как запуск ASA (например, он должен включить прокси-ASA для присоединения, чтобы помочь другим устройствам войти в домен). К таким взаимодействиям могут применяться другие условия, например, для работы в качестве посредника в присоединении, устройство сначала должно найти регистратор загрузки.

Переходы

- Выход из ACP. Устройство отключает последний (единственный) канал ACP к смежному устройству. Следующим будет состояние 2.
- Сброс к заводским настройкам. Удаляются все конфигурации и доменное отождествление LdevID. Следующим будет состояние 1.
- Включение и выключение питания. Устройство теряет все таблицы состояния, но сохраняет доменное отождествление LdevID. Следующим будет состояние 2.

4. Инфраструктура AN

Инфраструктура ANI обеспечивает уровень общей функциональности в сети с самонастройкой AN. Она предоставляет элементарные функции и службы, а также расширения. Автоматическая функция, состоящая из агентов ASA на узлах, использует описанные в этом разделе функции.

4.1. Именование

Каждому автоматическому устройству следует назначать уникальное имя в домене. Схему именования следует делать согласованной внутри домена. Имена обычно назначаются регистратором во время первой загрузки и сохраняются в течение жизненного цикла устройства. Все регистраторы в домене должны следовать одной схеме именования.

При отсутствии специфической для домена схемы именования следует применять принятую по умолчанию схему, использующую такую же логику как схема адресации, описанная в [RFC8994]. Имя устройства в этом случае состоит из Registrar-ID (например, MAC-адрес регистратора) и номера устройства. Примером такого имени может служить

```
0123-4567-89ab-0001
```

Первые 3 поля этого имени образованы MAC-адресов, а четвёртое содержит порядковый номер устройства.

4.2. Адресация

Агенты ASA должны взаимодействовать друг с другом, используя автоматическую адресацию инфраструктуры ANI узла, на котором размещается агент. В этом параграфе описан подход к адресации ANI используемой агентами ASA. Подход к адресации в плоскости данных сети выходит за рамки этого документа. Эта адресация может настраиваться и управляться традиционным способом или согласовываться как услуга ASA. Один из примеров использования такой автономной функции представлен в [RFC8992].

Автоматическая адресация является функцией ANI (нижняя часть рисунка 2) и, в частности ACP. Агенты ASA не имеют собственных адресов. Они могут использовать вызовы API или схему автоматической адресации ANI. Требования к схеме автоматической адресации перечислены ниже.

- Адресация без вмешательства (Zero-touch) для простых сетей, которым следует иметь полное самоуправление адресацией и не требовать централизованного управления, инструментов и планирования.
- Незначительное вмешательство для сложных сетей. Здесь требуется участие оператора для автоматического управления адресами, которое следует ограничивать высокоуровневым руководством, выраженным в Intent.
- Гибкость. Схема адресации должна позволять узлам перемещаться по сети, а сеть должна иметь возможность расширяться, делиться и сливаться с другими сетями.
- Устойчивость. Возможность негативного влияния администратора на адресацию (и связность) следует предотвращать в контексте сети с самоуправлением.
- Стабильность. Схеме адресации следует быть стабильной, однако реализация должна поддерживать восстановление при неожиданной смене адресов.
- Поддержка виртуализации. Автоматические функции могут работать на уровне физической сети и устройств или на уровне виртуальных машин, контейнеров и сетей. В частности, автоматические узлы могут поддерживать ASA в виртуальных объектах. Инфраструктуре и схеме адресации, следует поддерживать это.
- Простота. Схеме адресации следует быть простой, чтобы упростить организацию сети и дать администратору возможность простого устранения неполадок в автоматических функциях.
- Расширяемость. Схема адресации должна работать в сетях любого размера.
- Обновляемость. Схема должна поддерживать разные концепции адресации в будущем.

Предлагаемая схема адресации описана в документе «An Autonomic Control Plane (ACP)» [RFC8994].

4.3. Обнаружение

Традиционно большая часть требуемой узлу информации предоставляется путём настройки или через северный интерфейс. Автоматическим функциям следует минимально полагаться на северный интерфейс или отказаться от этого совсем, поэтому им нужно самостоятельно обнаруживать партнёров и другие ресурсы в сети. В этом параграфе рассматриваются функции обнаружения в сети с самоуправлением (AN).

Во-первых, обнаружение узлов и их возможностей является основной функцией при организации домена с самоуправлением. Это включает взаимное обнаружение автоматических узлов, в первую очередь смежных, а во вторую - партнёров, не находящихся на канале (off-link). В принципе для этого можно использовать имеющиеся механизмы обнаружения или применить новые механизмы, созданные для самоуправляемого контекста. Важно то, что обнаружение должно работать в сети с неизвестной заранее топологией и в идеале не требовать какой-либо ручной настройки и с узлами, запускающимися с заводской настройкой, а также после отказа или внезапной смены топологии.

Во-вторых, для сетевых служб, таких как аутентификация, проверка полномочий и учёт (Authentication, Authorization, Accounting или AAA), также следует поддерживать обнаружение без настройки. Сеть с самоуправлением (AN) может использовать имеющиеся функции обнаружения, применять новые подходы или то и другое вместе.

Таким образом, механизмы обнаружения могут быть полностью объединены с автоматической сигнализацией (см. следующий параграф) или использовать независимые механизмы, такие как обнаружение служб через DNS или Service Location Protocol. Выбор может быть независимым для каждого агента ASA, хотя инфраструктура может требовать того или иного общего минимума (например, обнаружение механизма защищённой загрузки или источника распространения информации, см. параграф 4.7).

В фазе 1 сети с самоуправлением (Autonomic Networking) используют для обнаружения протокол GRASP [RFC8990].

4.4. Сигнализация между автоматическими узлами

Автоматические узлы должны взаимодействовать между собой, например, для согласования и/или синхронизации технических целей (т. е. параметров сети) любого типа и сложности. Для этого нужна та или иная форма сигнализации между узлами. Автоматические узлы, реализующие определённый вариант, могут выбрать свой сигнальный протокол, если он соответствует общей модели безопасности. Однако в общем случае обмен данными может потребоваться любой паре узлов с самоуправлением, поэтому нужен общий протокол сигнализации. Предпосылкой для этого является возможность узлов обнаруживать друг друга без предварительной настройки, как отмечено выше. Для обеспечения общего характера обнаружение и сигнализация должны быть способны решать любые технические задачи, включая те, которым нужны сложные структуры данных. В документе «GeneRic Autonomic Signaling Protocol (GRASP)» [RFC8990] более подробно описаны требования к обнаружению, согласованию и синхронизации в AN. Документ также определяет протокол GRASP для этих целей, включающий встроенный, но необязательный процесс обнаружения.

Обычно предполагается, что GRASP работает внутри ACP (см. параграф 4.6) и зависит от ACP в плане безопасности. Возможна кратковременная работа протокола без защиты в процессе начальной загрузки (bootstrap). На автоматическом узле обычно работает один экземпляр GRASP, используемый несколькими агентами ASA. Однако не исключается использование на узле нескольких экземпляров GRASP, возможно с разными свойствами защиты.

4.5. Маршрутизация

Все автоматические узлы в домене должны быть способны взаимодействовать друг с другом, а на последующих фазах - ещё и с автоматическими узлами других доменов. Поэтому ACP полагается на функцию маршрутизации. Для взаимодействия сетей с самоуправлением они должны поддерживать общий протокол маршрутизации. В документе ACP [RFC8994] определён протокол маршрутизации для AN.

4.6. Автоматическая плоскость управления

Плоскость управления ACP поддерживает протоколы управления в автоматической сети AN. В описанной здесь архитектуре она реализована как наложенная сеть. В документе «An Autonomic Control Plane (ACP)» [RFC8994] описаны детали реализации. Данный документ использует термин «наложенная» (overlay) для обозначения набора парных смежностей с базовой топологией соединений. Это может отличаться от толкования термина overlay в контексте маршрутизации. Примеры использования ACP приведены в [RFC8368].

4.7. Распространение информации (*)

Некоторые формы информации требуют распространения через домен с самоуправлением. Такое распространение происходит в плоскости управления ACP. Например Intent распространяется по домену, как описано в [RFC7575]. Намерения (Intent) являются языком правил в сети с самоуправлением AN (см. параграф 7.2). Это правила высокого уровня и менять их следует нечасто (дни). Поэтому такую информацию, как Intent, следует рассылать в лавинном режиме всем узлам автоматического домена и в настоящее время нет ощутимой потребности использовать более направленные методы распространения. Предполагается, что Intent будет «монолитным» и рассылаться будет целиком. Один из возможных методов распространения Intent и других форм данных рассмотрен в [GRASP-DISTRIB]. Intent и распространение информации не входят в задачи рабочей группы ANIMA.

5. Инфраструктура защиты и доверия

Автоматические сети AN сами защищают себя. Все протоколы по умолчанию являются защищёнными и не требуют привлечения администратора для явной настройки защиты за исключением установки инфраструктуры PKI.

Автоматические узлы взаимодействуют напрямую и это требует защиты. Поскольку сети AN не полагаются на настройку конфигурации, здесь нет опций настройки, таких как заранее распространённые ключи и вместо этого должна применяться доверенная инфраструктура, такая как PKI. В этом разделе описаны принципы инфраструктуры доверия. На первой фазе AN устройство 1) находится в доверенном домене и само является доверенным или 2) находится за пределами домена доверия и считается недоверенным.

Принятый по умолчанию метод автоматического запуска инфраструктуры доверия определён в документе «Bootstrapping Remote Secure Key Infrastructure (BRSKI)» [RFC8995]. Агенты ASA, требуемые для регистрации, описаны в параграфе 6.3. Автоматические узлы должны реализовать агенты ASA для регистрации и посредничества в присоединении. ASA-регистратор можно реализовать на части устройств.

5.1. Инфраструктура открытых ключей

Домен с самоуправлением использует модель PKI. Конем доверия является удостоверяющий центр (Certification Authority или CA). Регистратор выступает в качестве центра регистрации (Registration Authority или RA). Минимальная реализация автоматического домена содержит один CA, один регистратор и элементы сети.

5.2. Сертификат домена

Каждое устройство в домене самоуправления использует сертификат домена (LdevID) для отождествления себя. Новое устройство использует предоставленный производителем сертификат IdevID в процессе начальной загрузки для получения сертификата LdevID. Процесс получения доменного сертификата и его формат описаны в [RFC8995].

5.3. MASA

Уполномоченный производителем орган подписания (Manufacturer Authorized Signing Authority или MASA) является доверенной службой для устройств с начальной загрузкой. MASA позволяет владельцу отслеживать устройства в домене, обеспечивая регистратору аудит, проверку полномочий и маркеры владения в процессе начальной загрузки, чтобы помочь при проверке подлинности устройств, пытающихся присоединиться к домену самоуправления и позволить присоединяющемуся устройству проверить корректность домена. Детали службы MASA, безопасности и применения описаны в [RFC8995].

5.4. Субдомены (*)

По умолчанию субдомены считаются отдельными доменами. Это предполагает отсутствие доверия между доменом и его субдоменами, а также между субдоменами одного домена. В частности не создаётся ACP, а Intent действует лишь в домене, для которого намерения заданы явно.

Рабочей группе ANIMA следует определить дополнительные модели доверия, например, разрешающие полное или частично доверие между доменом и субдоменом.

5.5. Кросс-доменная функциональность (*)

По умолчанию разные домены не могут взаимодействовать, ACP не создаётся и доверия между доменами нет. В будущем могут быть созданы модели с полным или частичным доверием между доменами.

6. Автоматические агенты служб

В этом разделе описана работа автоматических служб в инфраструктуре ANI.

6.1. Общее описание ASA

Агент ASA определён в [RFC7575] как: «Агент, реализованный на автоматическом узле и выполняющий автоматическую функцию частично (распределенная функция) или полностью. Таким образом, это процесс, использующий функции инфраструктуры ANI для достижения своих целей, обычно путём взаимодействия с другими ASA по протоколу GRASP [RFC8990] или иным способом. Агент также взаимодействует с конкретными объектами (target) своей функции, используя любой подходящий механизм. Если функция агента не очень проста, ASA потребуется обрабатывать перекрывающиеся асинхронные операции. Поэтому агент может быть достаточно сложной частью программы, работающей на прикладном уровне над инфраструктурой ANI. Рекомендации по проектированию ASA приведены в [ASA-GUIDELINES].

Можно выделить по меньшей мере 3 класса агентов ASA:

- простые ASA небольшого размера, которые могут работать где угодно;
- сложные, возможно многопоточные ASA с высокими требованиями к ресурсам, которые работают лишь на некоторых узлах;
- инфраструктурные ASA, использующие базовые функции ANI для поддержки самой инфраструктуры ANI, которые должны работать на всех автоматических узлах.

Автоматические узлы и их агенты ASA знают свои возможности и ограничения, связанные с оборудованием, микрокодом (firmware) и установленными программами, т. е. «осознают себя» (self-aware).

Роль автоматического узла зависит от Intent и поведения окружающей сети, включая поведение пересылки, свойства агрегирования, топологическое местоположение, пропускную способность, свойства туннеля или трансляции и т. п. Например, узел может выступать в качестве резервного для соседа, если его возможности позволяют это.

После начальной фазы обнаружения свойства узла и его соседей определяют поведение конкретного узла. Узел и его агенты ASA не имеют предварительной настройки для конкретной сети, где узел устанавливается.

Поскольку агенты ASA будут взаимодействовать с ANI, они будут зависеть от соответствующих интерфейсов API[1]. Желательна переносимость ASA между разными операционными системами, поэтому от API требуется универсальность. Интерфейс API для протокола GRASP описан в [RFC8991]. В общем случае агенты ASA будут разрабатываться и кодироваться специалистами в конкретной технологии и варианте применения, а не специалистами в части инфраструктуры ANI и её компонентов. Кроме того, они могут представляться на разных языках программирования, в частности, на языках, поддерживающих объекты, а также традиционные переменные и структуры. При разработке API это следует учитывать.

[1]Application programming interface - интерфейс с прикладными программами.

Должна быть возможность запуска ASA как непривилегированных процессов (пользовательское пространство), за исключением тех, которым требуются привилегии ядра (такие как инфраструктурные ASA). Также очень желательна возможность динамической загрузки ASA на работающем узле.

Поскольку автоматические системы должны быть самовосстанавливающимися, очень важно кодировать ASA с использованием отказоустойчивых методов программирования. Все ошибки в процессе работы должны обнаруживаться с выполнением подобающих действий по восстановлению с минимальным нарушением работы, но также следует предусматривать полный перезапуск ASA. Такие ситуации, как отказы при обнаружении или согласовании должны считаться обычными, при этом ASA следует повторять отказавшую операцию, предпочтительно с экспоненциальным ростом интервала повтора в случае продолжающихся отказов. При запуске ASA в нескольких потоках, эти потоки должны отслеживаться на предмет отказов и зависаний с выполнением соответствующих действий. Следует обращать внимание на «сборку мусора», чтобы у ASA не возникало нехватки ресурсов. Участия оператора не предполагается и в худшем случае агент ASA должен быть способен самостоятельно перезапуститься.

Агенты ASA автоматически используют возможности защиты, обеспечиваемой инфраструктурой ANI, в частности ACP и GRASP. Однако в дополнение к этому агенты сами отвечают за свою защиту, особенно при взаимодействии с конкретными объектами (target) своей функции. Поэтому при разработке ASA должен выполняться анализ безопасности сверх использования защиты ANI.

6.2. Управление жизненным циклом ASA

Агенты ASA, работающие в данной инфраструктуре ANI, могут происходить от разных провайдеров и преследовать разные цели. Управлению агентами ASA и их взаимодействием с ANI следует придерживаться общих принципов работы и соответствовать базовой модели управления жизненным циклом, обеспечивающим стандартные процессы:

- установки ASA, состоящей из копирования кода ASA на узел и его запуска;
- развёртывания ASA, связывающего экземпляр ASA с управляемым сетевым устройством (устройствами) или функцией;
- контроль исполнения ASA, задающий цикл управления ASA.

Жизненный цикл также определяет взаимодействия ASA с ANI в разных состояниях. Важные взаимодействия указаны ниже.

- Самоописание экземпляров ASA в конце развёртывания, формат которого должен определять информацию, требуемую ждя управления агентами ASA со стороны ANI.
- Контроль контура управления ASA в процессе исполнения. Сигнализация передаёт форматированные сообщения для управления исполнением ASA (по меньшей мере запуском и остановкой контура управления).

6.3. Конкретные ASA в инфраструктуре автоматической сети

Описанные ниже функции обеспечивают требуемую существенную функциональность в автоматических сетях AN и поэтому обязательны для реализации на автоматических узлах без ограничений. Они описаны здесь как агенты ASA, включающие базовые инфраструктурные компоненты. Детали реализации могут отличаться.

Три первых агента (поручительство, посредник присоединения, регистратор присоединения) совместно поддерживают процесс регистрации, описанный в разделе 5. Более подробное описание дано в [RFC8995].

6.3.1. Регистрационные ASA

6.3.1.1. ASA-поручитель

Этот агент ASA включает функцию автоматического узла, который выполняет начальную загрузку в домен с помощью посредника в присоединении (join proxy ASA). Такой узел называют поручителем (pledge) в процессе регистрации. Он должен по умолчанию устанавливаться на всех узлах, которым нужна начальная загрузка без предварительной настройки (zero-touch bootstrap).

6.3.1.2. ASA-посредник присоединения

Этот агент ASA включает функцию автоматического узла, которая помогает незарегистрированным смежным устройствам зарегистрировать себя в домене. Этот агент ASA должен устанавливаться на всех узлах, хотя в локальной сети требуется лишь 1 активный посредник присоединения (см. также [RFC8995]).

6.3.1.3. ASA-регистратор присоединения

Этот агент ASA включает функцию регистратора присоединения (Join Registrar) к автоматической сети AN. Такой агент не требуется устанавливать на всех узлах, достаточно разместить его на узлах с функцией Join Registrar.

6.3.2. ACP ASA

Этот агент ASA включает функцию плоскости управления ACP в автоматической сети AN. В частности, он обнаруживает другие потенциальные узлы ACP и поддерживает организацию и разрыв каналов ACP. Этот агент ASA должен устанавливаться на всех узлах. Подробное описание приведено в параграфе 4.6 и [RFC8994].

6.3.3. ASA для распространения информации (*)

Этот агент ASA выходит за рамки работы группы ANIMA и здесь представлен лишь в качестве справки.

ASA включает функцию распространения информации в AN. В частности, он анонсирует доступность Intent и другой информации всем остальным автоматическим узлам. Этот агент не требуется устанавливать на всех узлах, достаточно разместить его на узлах, реализующих функции распространения информации (см. параграф 4.7).

Отметим, что распространение информации может быть реализовано как функция в любом агенте ASA. Более подробное описание распространения информации дано в [GRASP-DISTRIB].

7. Управление и программируемость

В этом разделе рассматривается управление и программирование AN.

7.1. Управление (частично) автоматической сетью

Автоматическое управление обычно сосуществует в большинстве сетей с традиционными методами управления. Таким образом, в большинстве сред автоматическое управление будет определяться для отдельных функций. Примеры перекрытия функций приведены ниже.

- Автоматические функции могут применять традиционные методы и протоколы (например, SNMP и NETCONF) для выполнения задач управления внутри или вне ACP.
- Автоматические функции могут вызывать конфликты с некоторыми традиционными методами и протоколами.
- Традиционные функции могут использовать ACP, например, когда ещё нет доступности в плоскости данных.

Автоматические намерения (Intent) определяются на высоком уровне абстракции. Однако в силу необходимости обращения к отдельным управляемым элементам, автоматическому управлению нужны коммуникации на более низких уровнях (например, команды и запросы). Предполагается, что настройка конфигурации таких элементов будет выполняться, например, с использованием NETCONF и модулей YANG, а мониторинг - с помощью SNMP и MIB.

Конфликты могут возникать между принятым по умолчанию автоматическим поведением, автоматическими намерениями Intent и традиционными методами управления. Разрешение таких конфликтов достигается в автоматическом управлении с помощью приоритизации [RFC7575], где ручному управлению и управлению на уровне узла отдаётся более высокий приоритет. Таким образом, принятое по умолчанию автоматическое поведение имеет низший приоритет, затем следуют автоматические намерения (Intent), в высший приоритет имеют зависящие от узла методы управления, такие как использование командного интерфейса.

7.2. Намерения (*)

В текущих спецификациях реализаций Intent не рассматривается и в этом параграфе обсуждаются темы дальнейших исследований. Параграф содержит обзор Intent и способы управления намерениями. Intent и сетевое управление на основе правил (Policy-Based Network Management или PBNM) уже описаны в IETF (например, Policy Core Information Model или PCIM) и других органах стандартизации (Standards Development Organization или SDO), например, целевой группе по распределенному управлению (Distributed Management Task Force или DMTF).

Намерения (Intent) можно описать в абстрактной, декларативной политике высокого уровня, используемой для работы автоматического домена, такого как сеть предприятия [RFC7575]. Намерения следует ограничивать лишь высокоуровневыми рекомендациями, т. е. они не должны напрямую определять правила для каждого элемента сети в отдельности.

Intent можно уточнять до политики более низкого уровня с использованием различных подходов. Предполагается, что позволит приспособить намерения к возможностям управляемых устройств. Intent может содержать сведения о ролях и функциях, которые можно транслировать на конкретные узлы [RFC7575]. Одним из возможных уточнений Intent является применение правил «событие-условия-действие» (Event-Condition-Action или ECA).

Для Intent можно настраивать различные параметры, которые обычно предоставляет оператор. Некоторые из этих параметров могут влиять на поведение конкретных автоматических функций, а также способ использования Intent для управления автоматическим доменом.

Более подробное рассмотрение Intent приведено в [ANIMA-INTENT]. Для распространения Intent и других типов информации применяется протокол GRASP, см. [GRASP-DISTRIB].

7.3. Агрегированные отчёты (*)

Агрегированные отчёты не включены в текущие спецификации реализаций и в этом параграфе рассматриваются темы дальнейших исследований.

Автоматическим сетям AN следует минимизировать вовлечение операторов. С точки зрения поведения сети это выполняется с помощью автоматических намерений Intent, предоставляемых оператором. Аналогичным образом отчётам с описанием рабочего состояния сети следует агрегировать информацию от разных элементов сети для представления эффективности исполнения Intent. Поэтому отчёты в AN следует предоставлять на уровне сети [RFC7575].

В AN может происходить много событий одновременно, как и в традиционных сетях. Однако при подготовке отчёта для администратора эти события следует агрегировать для, чтобы избежать уведомлений от отдельных элементов сети. В этом контексте могут применяться алгоритмы для выбора включаемых в отчёт сведений (например, фильтры), способа их представления и сопоставления с другими событиями. Кроме того, события в отдельном элементе могут компенсироваться изменениями в других элементах для поддержки в масштабе сети поведения, заданного автоматическим намерением Intent.

Отчётность в AN может использовать тот же уровень абстракции, что и Intent. В этом контексте агрегированное представление текущего состояния сети AN можно использовать для переключения в другие режимы управления. Несмотря на то, что автоматическому управлению следует минимизировать участие человека, некоторые события могут требовать участия (действий) администратора.

7.4. Контуры обратной связи с NOC (*)

Контуры обратной связи нужны в AN для того, чтобы позволить вмешательство администратора или централизованной системы управления при сохранении принятого по умолчанию поведения. Через этот контур администратор должен получить приглашение от принятого по умолчанию действия для его подтверждения или переопределения.

Однонаправленные уведомления в центр управления сетью (Network Operations Center или NOC), которые не предлагают заданных по умолчанию действий и не допускают переопределения в рамках транзакции, рассматриваются подобно традиционным службам уведомления, таким как syslog. Предполагается их сосуществование с автономными методами, но этот вопрос в документе не рассматривается.

7.5. Контуры управления (*)

Контуры управления не рассматриваются в текущих спецификациях реализации и в этом разделе рассмотрены темы для дальнейших исследований.

Контуры управления служат в сетях AN для обеспечения базового механизма, позволяющего автоматической системе (самостоятельно) адаптироваться к различным факторам, способным менять цели, которых пытается достичь AN, или способы достижения этих целей. Например, при изменении потребностей пользователей, задач бизнеса и самой инфраструктуры ANI адаптация позволяет ANI изменить предоставляемые службы и ресурсы с учётом изменений.

Контуры управления работают для непрерывного наблюдения и сбора данных, позволяющих автоматической системе управления понять изменения в поведении управляемой системы, а затем предпринять действия по смене состояния этой системы в соответствии с целью. Адаптивные системы переносят принятие решения от статических заранее заданных команд к динамическим процессам, выполняемым в процессе работы.

Большинство автоматических систем использует замкнутый контур управления с обратной связью Таким контурам следует иметь возможность динамически меняться в процессе работы для приспособления к меняющимся потребностям пользователей, задачам бизнеса и изменениям в инфраструктуре ANI.

7.6. API (*)

В [RFC8991] определена концептуальная схема API для протокола базовой автоматической сигнализации (GeneRic Autonomic Signaling Protocol или GRASP). Этот API разработан для взаимодействия между агентами ASA по протоколу GRASP. Полная спецификации Standards Track API не включены в текущий спецификации реализаций.

Большинство API являются статическими, что означает их предопределённость и использование инвариантных механизмов работы с данными. Автоматическим сетям AN следует иметь возможность использования динамических API в дополнение к статическим.

Динамические API извлекают данные с использованием базового механизма и затем позволяют клиенту просматривать полученные данные и работать с ними. Такие API обычно используют самоанализ и/или рефлексию. Самоанализ помогает программам проверять типы и свойства объектов в процессе работы, а рефлексия позволяет программе манипулировать атрибутами, методами и/или метаданными объекта.

API должны быть способны выражать и сохранять семантику моделей данных. Например, программные соглашения [Meyer97] основаны на том, что интенсивно использующая программы система (такая как AN) является набором компонентов, чьи взаимодействия основаны на точно определённых спецификациях взаимных обязательств, которые нужно соблюдать. Обычно это включает указание:

- предварительных условий, которые должны быть выполнены до запуска исполнения метода;
- условий, которые должны быть выполнены при завершении исполнения метода;
- инвариантных атрибутов, которые не должны меняться в процессе исполнения метода.

7.7. Модели данных (*)

Модели данных не рассматриваются в текущих спецификациях реализации и этот параграф посвящён направлениям последующих исследований.

Определения модели данных и информационной модели адаптированы из [SUPA-DATA].

Информационная модель - это представление концепций, представляющих интерес для среды, в форме, независимой от репозитория, языка определения данных, языка запросов, языка реализации и протокола. Модель данных - это представление тех же концепций в форме, зависящей от хранилища, языка определения данных, языка запросов, языка реализации и протокола.

Полезность информационной модели заключается в определении объектов и их взаимоотношений технологически нейтральным способом. Это формирует концептуальный словарь, который могут использовать ANI и ASA. Модель данных представляет собой привязанное к технологии отображение всей или части информационной модели, которая будет применяться всей системой или её частью.

Система может иметь несколько моделей данных. Системы поддержки работы, например, обычно имеют несколько типов хранилищ, таких как SQL и NoSQL, чтобы воспользоваться преимуществами каждого из них. Если в автоматической системе нужны несколько моделей данных, следует использовать информационную модель, чтобы гарантировать связь концепций каждой модели данных без технологической предвзятости.

Модель данных важна для некоторых типов функций, таких как цикл адаптивного управления MRACL (Model-Reference Adaptive Control Loop. В более общем смысле модель данных может служить для определения объектов, атрибутов, методов и взаимоотношений программной системы (например, ANI, автоматического узла или агента ASA). Модель данных можно использовать при разработке API, а также любого языка для взаимодействия с сетью AN.

8. Координация между автоматическими функциями (*)

Координация между автоматическими функциями не включена в текущие спецификации реализаций и в этом разделе рассматриваются направления будущих исследований.

8.1. Проблема координации (*)

Разные автоматические функции могут конфликтовать по установкам некоторых параметров. Например, функция энергосбережения может захотеть отключить резервный канал, а функция распределения нагрузки - не желать этого. Администратор должен быть способен понять и разрешить такие взаимодействия, чтобы обеспечить в автоматической сети AN заданную (желаемую) производительность.

Между автоматическими функциями может быть несколько типов взаимодействия, например:

- кооперация, когда автоматическая функция может улучшить поведение или производительность другой автоматической функции (например, функция предсказания трафика, используемая функцией распределения);
- зависимость, когда одна автоматическая функция не может работать без наличия или доступности другой в сети AN;
- конфликт метрических значений, когда метрика влияет на параметры других автоматических функций (т. е. один параметр устанавливается разными автоматическими функциями).

Проблема координации, выходящая за рамки отдельных случаев, может быстро станет неразрешимой в больших сетях. Определение общего функционального блока для координации является первым шагом к системному решению проблемы. Жизненный цикл координации состоит из трёх этапов, указанных ниже.

- Во время сборки можно создать «статический план взаимодействий» на основе взаимосвязей функций и атрибутов. Этот план можно использовать для установки правил и задания приоритетов для выявленных конфликтов.
- Во время развёртывания автоматические функции ещё не активны или не действуют в сети. Создаётся динамический план взаимодействий» для каждого экземпляра каждой автоматической функции применительно к используемым ресурсам, включая выполняемые действия и отношения между ними. Этот план обеспечивает базу для выявления конфликтов при работе, их классификации и планирования подходящей стратегии и механизмов координации.
- При возникновении конфликтов в процессе работы арбитраж определяет стратегия координации. Кроме того, могут наблюдаться и интерферировать новые зависимости, что ведёт к обновлению плана взаимодействий с адаптацией стратегии и механизмов.

Можно разработать множество стратегий и механизмов, диапазон которых варьируется от базовых подходов случайный процесс или процесс на основе маркера) до более сложных вариантов с разделением по времени и иерархической оптимизацией, а также может применяться комплексный подход, такой как оптимизация по нескольким целям и применение других теорий управления и семейств алгоритмов.

8.2. Функциональный блок координации (*)

Общий функциональный блок координации является желательной частью эталонной модели ANIMA. Он предоставит средства обеспечения гарантии свойств сети и предсказуемой производительности или поведения, такие как стабильность и быстрое схождение при наличии нескольких взаимодействующих автоматических функций.

Для общего функционального блока координации требуется:

- общее описание автоматических функций, их атрибутов и жизненного цикла;
- общее представление информации и знаний (например, план взаимодействий);
- общий интерфейс команд (управления) между «агентом» координации и автоматическими функциями.

Могут также предоставляться руководства, рекомендации или BCP для аспектов, относящихся к стратегии и механизмам координации.

9. Вопросы безопасности

В этом разделе отдельно рассматриваются внутренние и внешние атаки. При внешней атаке все элементы сети и протоколы управляются и работают с защитой, а внешний злоумышленник может наблюдать пакеты в пути, внедрять и повторно использовать пакеты (replay). Внутренний атакующий имеет доступ к автоматическим узлам или иным средствам (например, удалённое исполнение кода на узле путём использования независимых от ACP уязвимостей платформы узла) для создания пакетов с произвольным содержимым в защищённых каналах ACP.

Если система имеет уязвимости в реализации или при работе (настройки), внешний атакующий может использовать такие уязвимости, чтобы стать внутренним (проникнуть в сеть).

9.1. Защита от внешних атак

Здесь предполагается, что все системы, вовлечённые в сеть AN, защищены и работают в в соответствии с текущей практикой (опытом). Методы защиты включают традиционные методы реализации и эксплуатации (такие как безопасный код, строгие алгоритмы случайных значений, надёжные пароль и т. п.), а также специфические механизмы AN (такие как защищённый сервис MASA).

Традиционные методы защиты реализации и эксплуатации выходят за рамки документа. Специфические для AN протоколы и методы также должны следовать традиционным методам защиты, поскольку пакеты, которые могут просматриваться или внедряться внешним, включают:

- защищённые от изменения;
- аутентифицированные;
- защищённые от replay-атак;
- защищённые в части конфиденциальности (шифрованные).

Кроме того, протоколам AN следует быть стойкими к отбрасыванию пакетов и перехвату с участием человека (man-in-the-middle или MITM). Эти требования задаются в документах AN Standards Track, определяющих применяемые методы, в частности в [RFC8990], [RFC8994], [RFC8995].

Большинство сообщений AN передаётся в криптографически защищённой плоскости управления ACP. Незащищёнными сообщениями AN вне ACP являются лишь сообщения простого метода обнаружения, описанные в параграфе 2.5.2 [RFC8990], - сообщения DULL (Discovery Unsolicited Link-Local), для которых заданы детальные правила.

Если сообщение AN наблюдаемо для посторонних, те могут получить важные сведения о конфигурации сети, применяемых методах защиты, отдельных пользователях и картинах их трафика. Из зашифрованных сообщений AN все равно можно получить некоторую информацию путём анализа трафика.

9.2. Риск внутренних атак

Сеть AN содержит автоматические устройства, формирующие распределенную систему с самоуправлением. Устройства в домене имеют свидетельства, полученные от общего доверенного центра (trust anchor) и могут применять их для организации взаимного доверия. Это значит, что любое устройство внутри домена доверия может по умолчанию использовать все распределенные функции во всем автоматическом домене злонамеренным способом.

Внутренний злоумышленник или внешний атакующий при наличии протокольных уязвимостей или работы без защиты могут использовать указанные ниже способы получить управление сетью AN.

- Внедрение обманного устройства в домен доверия за счёт нарушения (обхода) методов проверки подлинности. Это зависит от корректности спецификации, реализации и работы протоколов AN.
- Нарушение работы устройства, уже включённого в домен доверия и изменение его поведения. Эта угроза не является специфической для описанного здесь решения и применима ко всем сетям.
- Использование ещё неизвестных уязвимостей в AN или ином протоколе. Эта угроза относится к любой сети.

Указанные выше угрозы в принципе сравнимы с угрозами для других решений - при наличии ошибок в проектировании, реализации или эксплуатации невозможно гарантировать безопасность. Однако распределенная природа AN и особенно ACP значительно расширяет фронт атак. Например, взломанное устройство может иметь полную доступность по протоколу IP ко всем другим устройствам в ACP, а также применять все методы и протоколы AN.

Поэтому на следующей фазе работы ANIMA рекомендуется добавить модель защиты субдомена и не раскрывать возможным нарушителям весь домен. Кроме того, следует рассмотреть дополнительные механизмы защиты на уровне агентов ASA для автоматических функций с высоким уровнем рисков.

10. Взаимодействие с IANA

Этот документ не запрашивает действий IANA.

11. Литература

11.1. Нормативные документы

[IDevID] IEEE, "IEEE Standard for Local and metropolitan area networks - Secure Device Identity", IEEE 802.1AR, <https://1.ieee802.org/security/802-1ar>.

[RFC8990] Bormann, C., Carpenter, B., Ed., and B. Liu, Ed., "GeneRic Autonomic Signaling Protocol (GRASP)", RFC 8990, DOI 10.17487/RFC8990, May 2021, <https://www.rfc-editor.org/info/rfc8990>.

[RFC8994] Eckert, T., Ed., Behringer, M., Ed., and S. Bjarnason, "An Autonomic Control Plane (ACP)", RFC 8994, DOI 10.17487/RFC8994, May 2021, <https://www.rfc-editor.org/info/rfc8994>.

[RFC8995] Pritikin, M., Richardson, M., Eckert, T., Behringer, M., and K. Watsen, "Bootstrapping Remote Secure Key Infrastructure (BRSKI)", RFC 8995, DOI 10.17487/RFC8995, May 2021, <https://www.rfc-editor.org/info/rfc8995>.

11.2. Дополнительная литература

[ANIMA-INTENT] Du, Z., Jiang, S., Nobre, J. C., Ciavaglia, L., and M. Behringer, "ANIMA Intent Policy and Format", Work in Progress, Internet-Draft, draft-du-anima-an-intent-05, 14 February 2017, <https://tools.ietf.org/html/draft-du-anima-an-intent-05>.

[ASA-GUIDELINES] Carpenter, B., Ciavaglia, L., Jiang, S., and P. Peloso, "Guidelines for Autonomic Service Agents", Work in Progress, Internet-Draft, draft-ietf-anima-asa-guidelines-00, 14 November 2020, <https://tools.ietf.org/html/draft-ietf-anima-asa-guidelines-00>.

[GRASP-DISTRIB] Liu, B., Ed., Xiao, X., Ed., Hecker, A., Jiang, S., Despotovic, Z., and B. Carpenter, "Information Distribution over GRASP", Work in Progress, Internet-Draft, draft-ietf-anima-grasp-distribution-02, 8 March 2021, <https://tools.ietf.org/html/draft-ietf-anima-grasp-distribution-02>.

[Meyer97] Meyer, B., "Object-Oriented Software Construction (2nd edition)", Prentice Hall, ISBN 978-0136291558, 1997.

[RFC7575] Behringer, M., Pritikin, M., Bjarnason, S., Clemm, A., Carpenter, B., Jiang, S., and L. Ciavaglia, "Autonomic Networking: Definitions and Design Goals", RFC 7575, DOI 10.17487/RFC7575, June 2015, <https://www.rfc-editor.org/info/rfc7575>.

[RFC7576] Jiang, S., Carpenter, B., and M. Behringer, "General Gap Analysis for Autonomic Networking", RFC 7576, DOI 10.17487/RFC7576, June 2015, <https://www.rfc-editor.org/info/rfc7576>.

[RFC8368] Eckert, T., Ed. and M. Behringer, "Using an Autonomic Control Plane for Stable Connectivity of Network Operations, Administration, and Maintenance (OAM)", RFC 8368, DOI 10.17487/RFC8368, May 2018, <https://www.rfc-editor.org/info/rfc8368>.

[RFC8991] Carpenter, B., Liu, B., Ed., Wang, W., and X. Gong, "GeneRic Autonomic Signaling Protocol Application Program Interface (GRASP API)", RFC 8991, DOI 10.17487/RFC8991, May 2021, <https://www.rfc-editor.org/info/rfc8991>.

[RFC8992] Jiang, S., Ed., Du, Z., Carpenter, B., and Q. Sun, "Autonomic IPv6 Edge Prefix Management in Large-Scale Networks", RFC 8992, DOI 10.17487/RFC8992, May 2021, <https://www.rfc-editor.org/info/rfc8992>.

[SUPA-DATA] Halpern, J. and J. Strassner, "Generic Policy Data Model for Simplified Use of Policy Abstractions (SUPA)", Work in Progress, Internet-Draft, draft-ietf-supa-generic-policy-data-model-04, 18 June 2017, <https://tools.ietf.org/html/draft-ietf-supa-generic-policy-data-model-04>.

Благодарности

Вклад в работу и отклики предоставили Sheng Jiang, Roberta Maglione, Jonathan Hansford, Jason Coleman, Artur Hecker. Полезные рецензии представили Joel Halpern, Radia Perlman, Tianran Zhou, Christian Hopps.

Участники работы

Значительный вклад в документ внесли John Strassner (Huawei), Bing Liu (Huawei), Pierre Peloso (Nokia).

Адреса авторов

Michael H. Behringer (editor)
Email: Michael.H.Behringer@gmail.com

Brian Carpenter
School of Computer Science
University of Auckland
PB 92019
Auckland 1142
New Zealand
Email: brian.e.carpenter@gmail.com

Toerless Eckert
Futurewei USA
2330 Central Expy
Santa Clara, CA 95050
United States of America
Email: tte+ietf@cs.fau.de

Laurent Ciavaglia
Nokia
Villarceaux
91460 Nozay
France
Email: laurent.ciavaglia@nokia.com

Jéferson Campos Nobre
Federal University of Rio Grande do Sul (UFRGS)
Av. Bento Gonçalves, 9500
Porto Alegre-RS
91501-970
Brazil
Email: jcnobre@inf.ufrgs.br

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru