

RPC: Remote Procedure Call Protocol Specification Version 2

Спецификация протокола RPC версии 2

Статус документа

Это документ содержит проект стандарта протокола Internet для сообщества Internet и служит приглашением к дискуссии в целях развития и совершенствования протокола. Текущее состояние стандартизации и статус протокола можно узнать из текущей версии документа Internet Official Protocol Standards (STD 1). Допускается свободное распространение документа.

Авторские права

Авторские права (Copyright (c) 2009) принадлежат IETF Trust и лицам, указанным в качестве авторов документа. Все права сохраняются.

К этому документу применимы права и ограничения, перечисленные в BCP 78 и IETF Trust Legal Provisions и относящиеся к документам IETF (<http://trustee.ietf.org/license-info>), на момент публикации данного документа. Прочтите упомянутые документы внимательно.

Аннотация

Этот документ определяет версию 2 протокола удалённого вызова процедур (Remote Procedure Call или RPC) от Open Network Computing (ONC) в том виде, в котором он развёрнут и принят. Документ отменяет RFC 1831.

Оглавление

1. Введение.....	2
1.1. Уровни требований.....	2
2. Отличия от RFC 1831.....	2
3. Терминология.....	2
4. Модель RPC.....	2
5. Транспорт и семантика.....	3
6. Независимость от привязки.....	3
7. Проверка подлинности.....	3
8. Требования к протоколу RPC.....	3
8.1. Программы и процедуры RPC.....	3
8.2. Аутентификация, целостность и приватность.....	4
8.3. Назначение номеров программ.....	4
8.4. Другие применения протокола RPC.....	4
8.4.1. Пакетная обработка.....	4
8.4.2. Широковещательные вызовы RPC.....	5
9. Протокол сообщений RPC.....	5
10. Протоколы аутентификации.....	6
10.1. Пустая (Null) аутентификация.....	7
11. Стандарт маркировки записей.....	7
12. Язык RPC.....	7
12.1. Пример службы, описанной на языке RPC.....	7
12.2. Спецификация языка RPC.....	7
12.3. Замечания по синтаксису.....	8
13. Взаимодействие с IANA.....	8
13.1. Запрос номеров в IANA.....	8
13.2. Защита прошлых назначений.....	8
13.3. Назначение номеров RPC.....	8
13.3.1. Номера, выделяемые IANA.....	8
13.3.2. Номера, выделяемые локальным администратором.....	8
13.3.3. Временные номера.....	8
13.3.4. Резервный блок.....	9
13.3.5. Субблоки номеров RPC.....	9
13.4. Назначение номеров для вариантов аутентификации RPC.....	9
13.4.1. Правила назначения.....	9
13.4.2. Варианты и псевдо-варианты аутентификации.....	9
13.5. Выделение номеров для статуса аутентификации.....	10
13.5.1. Правила назначения.....	10
14. Вопросы безопасности.....	10
Приложение А. Системная аутентификация.....	10
Приложение В. Запрос связанных с RPC номеров в IANA.....	11
Приложение С. Выделенные номера.....	11
Нормативные документы.....	30
Дополнительная литература.....	30

1. Введение

Документ задаёт версию 2 протокола удалённого вызова процедур ONC RPC. Задан протокол сообщений на языке внешнего представления данных (eXternal Data Representation или XDR) [RFC4506] и предполагается, что читатели знакомы с этим языком. Документ не пытается обосновать системы удалённого вызова процедур или описать их применение. Для понимания концепций удалённых вызовов процедур рекомендуется статья Birrell и Nelson [XRPC].

1.1. Уровни требований

Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не нужно** (SHALL NOT), **следует** (SHOULD), **не следует** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе интерпретируются в соответствии с [RFC2119].

2. Отличия от RFC 1831

Этот документ отменяет [RFC1831] как полномочное описание RPC, не меняя протокол в линии. Основные отличия от RFC 1831 указаны ниже.

- Добавлено приложение, описывающее способы запроса разработчиками новых номеров программ RPC, номеров вариантов аутентификации и статуса аутентификации в IANA, а не в Sun Microsystems.
- Добавлен раздел 13. Взаимодействие с IANA, описывающий прошлую политику выделения номеров и правила IANA для назначения номеров в будущем.
- Уточнена спецификация языка RPC в соответствии с текущим применением.
- Улучшен раздел 14. Вопросы безопасности с учётом опыта работы с вариантами строгой защиты.
- Задана спецификация новых ошибок аутентификации, часто возникающих в современных реализациях RPC.
- Обновлено заявления IETF о правах интеллектуальной собственности.

3. Терминология

В этом документе обсуждаются клиенты, вызовы, серверы, отклики, службы, программы, процедуры и версии. У каждого удалённого вызова процедуры есть 2 стороны - активный клиент, который вызывает серверную сторону, и сервер, который возвращает ответ. Сетевая служба - это набор из одной или нескольких удалённых программ. Удалённая программа реализует 1 или несколько удалённых процедур, а процедуры, их параметры и результаты документируются в спецификациях протоколов конкретных программ. Сервер может поддерживать не одну версию удалённой программы для совместимости с меняющимися протоколами. Например, сетевая файловая служба может состоять из двух программ - одна может работать с приложениями высокого уровня, такими как управление доступом к файлам и блокировка, а другая - работать с низкоуровневыми операциями ввода-вывода и иметь такие процедуры, как чтение и запись. Клиент сетевой службы будет вызывать процедуры, связанные с этими двумя программами от имени пользователя (клиента).

Термины «клиент» и «сервер» применяются лишь к конкретным транзакциям - конкретный элемент оборудования (хост) или программы (процесс или программа) может быть как клиентом, так и сервером. Например, программа, предоставляющая услуги удалённого исполнения, может быть клиентом сетевой файловой службы.

4. Модель RPC

Протокол ONC RPC основан на модели удалённого вызова процедуры, похожей на модель локального вызова. В локальном случае вызывающий помещает аргументы в некое определённое место (например, в окно регистров), а затем передаёт управление процедуре и в конечном итоге получает управление обратно. В этот момент результаты процедуры извлекаются из определённого места и вызывающий продолжает исполнение.

Модель удалённого вызова похожа. Один поток (thread) управления логически проходит через два процесса - процесс вызывающего и процесс сервера. Вызывающий сначала передаёт сообщение с вызовом серверному процессу и ждёт (блокирует) ответное сообщение. Сообщение вызова включает параметры процедуры, а отклик - её результаты. После получения отклика результаты процедуры извлекаются и вызывающий возобновляет исполнение. На серверной стороне процесс бездействует в ожидании вызывающего сообщения. Получив сообщение, серверный процесс извлекает параметры процедуры, вычисляет результат, передаёт сообщение с откликом и ждёт следующего сообщения с вызовом.

В этой модели в каждый момент активен лишь один из двух процессов, однако модель служит лишь для примера. Протокол ONC RPC не накладывает ограничений на реализуемую модель параллелизма и можно применять разные модели. Например, реализация может сделать вызовы RPC асинхронными, чтобы клиент мог выполнять полезную работу в процессе ожидания отклика от сервера. Другой вариант заключается в создании сервером отдельной задачи для обработки входящих вызовов, чтобы можно было принимать другие запросы во время обработки уже полученного.

Имеется несколько важных различий между локальными и удалёнными вызовами процедур.

- Обработка ошибок - при удалённых вызовах нужно обрабатывать отказы удалённого сервера и сети.
- Глобальные переменные и побочные эффекты. Поскольку у сервера нет доступа к пространству адресов клиента, скрытые аргументы нельзя передать как глобальные переменные или вернуть как побочный эффект.
- Производительность - удалённые процедуры обычно работают как минимум на порядок медленней, чем локальные вызовы.
- Аутентификация - поскольку удалённые вызовы процедур передаются через незащищённые сети, требуется проверка подлинности, препятствующая одному элементу представиться как другой (замаскироваться).

Несмотря на наличие инструментов для автоматической генерации клиентских и серверных библиотек для данной службы, протоколы должны разрабатываться очень тщательно.

5. Транспорт и семантика

Протокол RPC можно реализовать на основе разных транспортных протоколов. Область определения протокола RPC включает лишь спецификацию и интерпретацию сообщений, не включая способов их передачи. Однако приложение может захотеть получить информацию о транспортном уровне (и, возможно, управлять им) через интерфейс, заданный в этом документе. Например, транспортный протокол может вносить ограничения на максимальный размер сообщений RPC или быть ориентирован на потоки, подобно TCP [RFC0793], где размер не ограничивается. Клиент и сервер должны согласовать выбор транспортного протокола.

Важно отметить, что RPC не пытается обеспечить надёжность и приложению может потребоваться знать тип транспортного протокола, используемого RPC. Если приложению известно, что протокол работает на основе надёжного транспорта, такого как TCP, большая часть работы уже сделана. Однако при работе на основе транспорта без гарантий, такого как UDP [RFC0768], приложение должно реализовать свои правила для тайм-аутов, повтора передачи и обнаружения дубликатов, поскольку протокол RPC не предоставляет таких услуг.

Независимость протокола RPC от транспорта ведёт к отсутствию привязки конкретной семантики требований к удалённым процедурам и их выполнению. Семантика может быть выведена из базового транспортного протокола (но её следует указывать явно). Рассмотрим, например, работу RPC на основе транспорта без гарантий, такого как UDP. Если приложение повторяет сообщения с запросом RPC после тайм-аута и не получает отклика, оно не будет знать, сколько раз была выполнена запрошенная процедура. При получении отклика приложение будет знать, что процедура выполнена по меньшей мере один раз.

Сервер может запоминать полученные от клиента запросы и не выполнять их повторно для обеспечения некоторой семантики выполнения процедуры не более одного раза. Сервер может реализовать это на основе идентификатора транзакции, помещаемого в каждое сообщение RPC. Этот идентификатор используется, прежде всего, клиентом RPC для сопоставления откликов с запросами. Однако клиентское приложение может запоминать идентификатор после выполнения вызова и не использовать тот же идентификатор в других вызовах, чтобы обеспечить семантику исполнения не более одного раза. Серверу не разрешается проверять идентификатор иначе, чем на равенство.

При использовании «надёжного» транспорта, такого как TCP, приложение может по отклику сделать вывод о выполнении процедуры в точности 1 раз, но при отсутствии отклика оно не может считать, что процедура не была выполнена. Отметим, что даже при использовании ориентированных на соединения протоколов, вроде TCP, приложению все равно нужны тайм-ауты и повторные соединения для обработки отказов сервера.

Помимо протоколов, основанных на дейтаграммах и соединениях, имеются иные варианты транспорта. Например, протокол «запрос-отклик», такой как [VMTP], может быть естественным транспортным решением для RPC. В ONC RPC сейчас применяются транспортные протоколы TCP и UDP. В разделе 11. Стандарт маркировки записей описан механизм, реализованный в ONC RPC для ориентированного на соединения и потоки транспорта, такого как TCP. Механизмы, которые будут применяться для передачи сообщений ONC RPC будущим транспортом с другими структурными характеристиками, следует задавать в RFC категории Standards Track, как только дополнительный транспорт задан.

6. Независимость от привязки

Привязка конкретного клиента к конкретному сервису и параметрам транспорта не являются частью спецификации протокола RPC. Эта важная и необходимая функция оставлена программам вышележащего уровня. Разработчики могут представлять протокол RPC как сетевую инструкцию перехода в подпрограмму (jump-subroutine instruction или JSR), загрузчик делает JSR используемой и сам применяет JSR для решения задачи. Точно так же программа привязки делает RPC используемой и может применять RPC для решения задачи.

7. Проверка подлинности

Протокол RPC предоставляет в каждом сообщении поля, требуемые для предоставления клиента (и сервера) другой стороне. На основе аутентификации могут создаваться механизмы защиты и контроля доступа с возможностью поддержки нескольких протоколов. Поле в заголовке RPC указывает применяемый для аутентификации протокол. Сведения о протоколах аутентификации приведены в параграфе 8.2. Аутентификация, целостность и приватность.

8. Требования к протоколу RPC

Протокол RPC должен обеспечивать:

- однозначное указание вызываемой процедуры;
- средства сопоставления откликов с запросами;
- средства аутентификации клиентов и серверов.

Помимо этих требования заслуживают поддержки указанные ниже функции, обнаруживающие смену протокола, ошибки реализации, пользователей и администраторов сети.

- Несоответствие протоколов RPC.
- Несоответствие версии протокола удалённой программы.
- Ошибки протокола (например, некорректное указание параметров процедуры).
- Причины отказа при аутентификации.
- Любые другие причины, по которым запрошенная процедура не была вызвана.

8.1. Программы и процедуры RPC

Сообщение RPC с вызовом процедуры содержит 3 целочисленных поля без знака - номер удалённой программы, номер версии удалённой программы и номер удалённой процедуры, - которые однозначно указывают вызываемую

процедуру. Номера программ централизованно администрирует IANA. После получения разработчиком номера программы он может реализовать свою программу и первая реализация будет вероятно иметь версию номер 1, но использование номера 0 **недопустимо**. Поскольку большинство новых протоколов развиваются, поле версии (version) в сообщении с вызовом указывает, версию, используемую вызывающей стороной. Номера версий позволяют поддерживать старый и новый протокол в одном серверном процессе.

Номер процедуры указывает вызываемую процедуру. Эти номера документируются в спецификации протокола конкретной программы. Например, спецификация протокола файловой службы может задать номер 5 для процедуры чтения (read) и 12 - для процедуры записи (write).

Протокол сообщений RPC, как и протокол удалённой программы, может изменяться, поэтому сообщение с вызовом указывает также номер версии RPC, который для описанной здесь версии протокола имеет значение 2.

Сообщение с откликом на запрос содержит сведения, достаточные, чтобы различать указанные ниже случаи ошибок.

- Удалённая реализация RPC не поддерживает протокол версии 2 и возвращает старший и младший номера поддерживаемых версий.
- Удалённая программа недоступна на удалённой системе.
- Удалённая программа не поддерживает запрошенный номер версии и возвращает старший и младший номера поддерживаемых версий.
- Процедуры с запрошенным номером не существует (обычно это связано с ошибкой протокола или программы на стороне клиента).
- Параметры удалённой процедуры представляются серверу некорректными (мусор) - обычно это связано с рассогласованием клиента и сервера.

8.2. Аутентификация, целостность и приватность

Положения для аутентификации клиента серверу (и наоборот) являются частью протокола RPC. Сообщение с вызовом включает два поля аутентификации - свидетельство (credential) и верификатор (verifier), а отклик содержит лишь поле верификатор (verifier) для отклика. Спецификация протокола RPC определяет все три поля как неинтерпретируемые (opaque), как представлено ниже на языке XDR [RFC4506]).

```
enum auth_flavor {
    AUTH_NONE      = 0,
    AUTH_SYS       = 1,
    AUTH_SHORT      = 2,
    AUTH_DH         = 3,
    RPCSEC_GSS      = 6,
    /* могут дополняться другими */
};

struct opaque_auth {
    auth_flavor flavor;
    opaque body<400>;
};
```

Иными словами, любая структура opaque_auth является перечисляемым значением auth_flavor, а за ним следует до 400 байтов, которые не интерпретируются (opaque) реализацией протокола RPC. Интерпретация и семантика данных в полях аутентификации задаётся отдельной, независимой спецификацией протокола проверки подлинности.

Если параметры аутентификации отклонены, сообщение с откликом содержит информацию о причинах отклонения. Как показано RPCSEC_GSS, auth_flavor позволяет поддерживать также защиту целостности и приватности.

8.3. Назначение номеров программ

Номера программ разделены на группы, показанные ниже.

0x00000000	резерв
0x00000001 - 0x1fffffff	для распределения IANA
0x20000000 - 0x3fffffff	выделяет локальный администратор
	(некоторые блоки выделены здесь)
0x40000000 - 0x5fffffff	временные (динамические)
0x60000000 - 0x7fffffff	резерв
0x7f000000 - 0x7fffffff	не распределены
0x80000000 - 0xffffffff	резерв

Первая группа содержит номера, распределяемые IANA, которые должны быть одинаковыми для всех сайтов. Второй диапазон предназначен для приложений конкретного сайта и применяется в основном для отладки новых программ. Когда на сайте разработано приложение, которое может представлять интерес для других, ему следует запросить номер из первого диапазона. Разработчики приложений могут запрашивать блоки номеров программ RPC из первого диапазона, как описано в Приложении В. Третья группа номеров предназначена для приложений, задающих номера программ динамически, а последние группы зарезервированы на будущее.

8.4. Другие применения протокола RPC

Этот протокол предназначен для вызова удалённых процедур. Обычно каждое сообщение с вызовом сопоставляется с сообщением отклика, однако сам протокол служит для передачи сообщений, с помощью которых могут быть реализованы другие (не связанные с вызовом процедур) протоколы.

8.4.1. Пакетная обработка

Пакетная обработка может быть полезна в случаях, когда клиенту нужно передать серверу произвольно большую последовательность сообщений с вызовами. Обычно для этого применяются потоковые транспортные протоколы с

надёжной доставкой (такие как TCP). При пакетной обработке клиент не ожидает отклика от сервера, а сервер не возвращает отклик на пакетный вызов. Цепочка пакетных вызовов обычно завершается легитимной операцией вызова удалённой процедуры для очистки конвейера (pipeline) и получения позитивного подтверждения.

8.4.2. Широковещательные вызовы RPC

В широковещательных протоколах клиент передаёт в сеть широковещательный вызов и ждёт множества откликов. Для этого требуется использовать основанный на пакетах транспортный протокол (такой как UDP). Серверы, поддерживающие широковещательные протоколы, обычно отвечают лишь при успешном выполнении вызова и молчат в случае ошибки, но это может зависеть от приложения. Принципы широковещательных RPC применимы и к групповой передаче (multicast) - запросы RPC будут отправляться на групповой адрес.

9. Протокол сообщений RPC

В этом разделе определён протокол сообщений RPC на языке описания данных XDR [RFC4506].

```
enum msg_type {
    CALL = 0,
    REPLY = 1
};
```

Отклик на сообщение с вызовом может принимать две формы - восприятие или отказ.

```
enum reply_stat {
    MSG_ACCEPTED = 0,
    MSG_DENIED = 1
};
```

Если вызов воспринят, возвращается результат попытки вызова удалённой процедуры.

```
enum accept_stat {
    SUCCESS = 0, /* Вызов RPC завершился успешно */
    PROG_UNAVAIL = 1, /* У удалённой стороны нет экспортируемой
        программы */
    PROG_MISMATCH = 2, /* У удалённой стороны нет поддерживаемой
        версии */
    PROC_UNAVAIL = 3, /* Программа не поддерживает процедуру */
    GARBAGE_ARGS = 4, /* Процедура не может декодировать параметры */
    SYSTEM_ERR = 5 /* Например, ошибка выделения памяти */
};
```

Ниже указаны возможные причины отклонения сообщения с вызовом.

```
enum reject_stat {
    RPC_MISMATCH = 0, /* Номер версии RPC не равен 2 */
    AUTH_ERROR = 1 /* Удалённая сторона не смогла
        аутентифицировать вызывающего */
};
```

Ниже указаны причины отказа при аутентификации.

```
enum auth_stat {
    AUTH_OK = 0, /* Успех */
    /*
     * Отказы на удалённой стороне
     */
    AUTH_BADCRED = 1, /* Непригодное свидетельство */
    AUTH_REJECTEDCRED = 2, /* Клиент должен начать новую сессию */
    AUTH_BADVERF = 3, /* Непригодный верификатор */
    AUTH_REJECTEDVERF = 4, /* Срок действия верификатора
        или использован повтор (replay) */
    AUTH_TOOWEAK = 5, /* Отвергнуто по соображениям
        безопасности */
    /*
     * Локальный отказ
     */
    AUTH_INVALIDRESP = 6, /* Поддельный верификатор в отклике */
    AUTH_FAILED = 7, /* Неизвестная причина */
    /*
     * Ошибки AUTH_KERB - отменены, см. [RFC2695]
     */
    AUTH_KERB_GENERIC = 8, /* Общая ошибка kerberos */
    AUTH_TIMEEXPIRE = 9, /* Просроченное свидетельство */
    AUTH_TKT_FILE = 10, /* Проблема с файлом квитанции */
    AUTH_DECODE = 11, /* Не удалось декодировать
        authenticator */
    AUTH_NET_ADDR = 12, /* Неверный адрес сети в квитанции */
    /*
     * Ошибки RPCSEC_GSS, связанные с GSS
     */
    RPCSEC_GSS_CREDPROBLEM = 13, /* Нет свидетельства для
        пользователя */
    RPCSEC_GSS_CTXPROBLEM = 14 /* Проблема с контекстом */
};
```

По мере добавления механизмов аутентификации список кодов может расширяться. Агентство IANA будет выдавать новые номера auth_stat в порядке их запроса (First Come First Served), как указано в разделе 13. Взаимодействие с IANA и приложении В.

Все сообщения RPC начинаются с идентификатора транзакции xid, за которым следует двухэлементное объединение (union). Дискриминантом union является тип сообщения (msg_type), который может принимать два значения. Поле xid в

сообщении с откликом (REPLY) всегда соответствует одноименному полю в исходном сообщении с вызовом (CALL). Важно подчеркнуть, что поле `xid` служит лишь для сопоставления откликов с запросами у клиента и обнаружения сервером повторов передачи, серверная сторона не может считать это поле порядковым номером.

```
struct rpc_msg {
    unsigned int xid;
    union switch (msg_type mtype) {
        case CALL:
            call_body cbody;
        case REPLY:
            reply_body rbody;
    } body;
};
```

В теле вызова RPC версии 2 поле `rpcvers` должно иметь значение 2. Поля `prog`, `vers` и `proc` задают удалённую программу, номер её версии и вызываемую в этой программе процедуру. За этими полями следуют два параметра аутентификации - `cred` (свидетельство для проверки подлинности) и `verf` (верификатор для аутентификации). После этих параметров указываются параметры, передаваемые удалённой процедуре в соответствии с конкретным протоколом.

Роль `vers` состоит в проверке свидетельства аутентификации (`cred`). Отмтим, что два эти элемента разделены исторически, но всегда применяются вместе как один логический объект.

```
struct call_body {
    unsigned int rpcvers;          /* Должно быть 2 */
    unsigned int prog;
    unsigned int vers;
    unsigned int proc;
    opaque_auth cred;
    opaque_auth verf;
    /* Здесь начинаются параметры процедуры */
};
```

Тело отклика на вызов RPC показано ниже.

```
union reply_body switch (reply_stat stat) {
    case MSG_ACCEPTED:
        accepted_reply areply;
    case MSG_DENIED:
        rejected_reply rreply;
} reply;
```

Отклик на воспринятый сервером вызов RPC может указывать ошибку, несмотря на принятие запроса. Первое поле служит для проверки аутентификации и генерируется сервером для подтверждения себя клиенту. Далее следует объединение (union) с перечисляемым дискриминатором `accept_stat`, ветвь `SUCCESS` в объединении зависит от протокола, а ветви `PROG_UNAVAIL`, `PROC_UNAVAIL`, `GARBAGE_ARGS` и `SYSTEM_ERR` имеют тип `void`. Ветвь `PROG_MISMATCH` задаёт младшую и старшую версию удалённой программы, поддерживаемой сервером.

```
struct accepted_reply {
    opaque_auth verf;
    union switch (accept_stat stat) {
        case SUCCESS:
            opaque results[0];
            /*
             * Здесь начинаются зависящие от процедуры результаты
             */
        case PROG_MISMATCH:
            struct {
                unsigned int low;
                unsigned int high;
            } mismatch_info;
        default:
            /*
             * Void. Варианты включают PROG_UNAVAIL, PROC_UNAVAIL,
             * GARBAGE_ARGS, SYSTEM_ERR.
             */
            void;
    } reply_data;
};
```

Вызов RPC может быть отвергнут сервером по двум причинам - отсутствие совместимой версии протокола RPC (`RPC_MISMATCH`) или отказ при аутентификации (`AUTH_ERROR`). При несоответствии версий сервер возвращает младший и старший номер поддерживаемой версии RPC. При отказе проверки подлинности указывается статус отказа.

```
union rejected_reply switch (reject_stat stat) {
    case RPC_MISMATCH:
        struct {
            unsigned int low;
            unsigned int high;
        } mismatch_info;
    case AUTH_ERROR:
        auth_stat stat;
};
```

10. Протоколы аутентификации

Выше было отмечено, что параметры аутентификации не интерпретируются, но открыты для остальной части протокола RPC. В этом разделе рассматриваются два стандартных варианта аутентификации. Разработчики могут предлагать свои варианты с такими же правилами назначения им номеров, какие применяются для номеров программ. Вариант со свидетельством или верификаторами относится к значению поля `flavor` в структуре `opaque_auth`. Номера

вариантов, подобно номерам программ RPC, администрируются централизованно и разработчики могут назначать новым вариантам номера в соответствии с методами, указанными в Приложении В. Свидетельства и верификаторы представляются как неинтерпретируемые данные переменного размера (поле body в структуре opaque_auth).

В этом документе описаны два варианта проверки подлинности. Пустая (Null) аутентификация, описанная в следующем параграфе, является обязательной и **должна** быть доступна во всех реализациях. Системная аутентификация (AUTH_SYS) описана в Приложении А. Разработчики могут включать AUTH_SYS в свои реализации для поддержки имеющихся приложений. Более защищённые варианты аутентификации указаны в разделе 14. Вопросы безопасности.

10.1. Пустая (Null) аутентификация

Зачастую нужны вызовы, когда клиенту не важно своё отождествление, а серверу все равно, кто является клиентом. В таких случаях в сообщениях RPC в качестве свидетельства, верификатора, а также верификатора в отклике применяется значение AUTH_NONE. Связанные данные для AUTH_NONE не заданы и рекомендуется, чтобы размер неинтерпретируемых данных был нулевым (нет данных).

11. Стандарт маркировки записей

Когда сообщения RPC передаются через транспорт в форме потока байтов (как в TCP), требуется отделять одно сообщение от другого для обнаружения и исправления возможных ошибок протокола. Это называется маркировкой записей (record marking или RM). Одно сообщение RPC помещается в одну запись RM.

Запись состоит из одного или нескольких фрагментов. Фрагмент имеет 4-байтовый заголовок, за которым может следовать до $2^{31} - 1$ байтов данных. Байты кодируют целые числа без знака, как и в случае с целыми числами XDR, от старшего байта к младшему. Заголовок представляет два значения - логическое, указывающее, является ли фрагмент последним (установленный в 1 бит указывает последний фрагмент), и 31-битовое целое число без знака, указывающее размер данных во фрагменте. Логическое значение занимает старший бит заголовка, а размер - 31 младший бит (отметим, что эта спецификация записи **не** использует стандартный формат XDR).

12. Язык RPC

Как и для описания типов данных XDR на формальном языке, нужно формально описать и процедуры, которые работают с этими типами данных XDR. Язык RPC является расширением языка XDR путём добавления деклараций program, procedure, version. Ключевые слова program и version зарезервированы в языке RPC и реализации компиляторов XDR **могут** резервировать эти слова даже при их включении в «чистые» описания XDR (без RPC). Ниже приведён пример описания сущности языка.

12.1. Пример службы, описанной на языке RPC

Ниже представлен пример простой программы проверки связи.

```
program PING_PROG {
    /*
     * Последняя и наибольшая версия
     */
    version PING_VERS_PINGBACK {
        void
        PINGPROC_NULL(void) = 0;
        /*
         * Ping для клиента возвращает время кругового обхода
         * (в мксек). Значение -1 указывает тайм-аут.
         */
        int
        PINGPROC_PINGBACK(void) = 1;
    } = 2;

    /*
     * Исходная версия
     */
    version PING_VERS_ORIG {
        void
        PINGPROC_NULL(void) = 0;
    } = 1;
} = 1;

const PING_VERS = 2; /* Последняя версия */
```

Первая описанная версия - PING_VERS_PINGBACK имеет 2 процедуры: PINGPROC_NULL и PINGPROC_PINGBACK. PINGPROC_NULL не имеет аргументов и не возвращает результата, но полезна для расчёта времени кругового обхода между клиентом и сервером. По соглашению, процедуре 0 в любом протоколе RPC следует иметь такую же семантику и никогда не требовать аутентификации. Вторая процедура применяется для клиента, чтобы заставить сервер выполнить операцию ping в сторону клиента и вернуть интервал времени, затраченного на операцию (в микросекундах). PING_VERS_ORIG является исходной версией протокола и не включает процедуру PINGPROC_PINGBACK. Это полезно для совместимости со старыми клиентскими программами и по мере развития самой программы может быть полностью исключено из протокола.

12.2. Спецификация языка RPC

Язык RPC идентичен языку XDR, заданному в RFC 4506, за исключением добавления program-def, описанного ниже.

```
program-def:
    "program" identifier "{"
    version-def
```

```

version-def *
}" " =" constant ";"

version-def:
"version" identifier "{"
  procedure-def
  procedure-def *
}" " =" constant ";"

procedure-def:
  proc-return identifier "(" proc-firstarg
  (" type-specifier )" * ")" " =" constant ";"

proc-return: "void" | type-specifier

proc-firstarg: "void" | type-specifier

```

12.3. Замечания по синтаксису

- Ключевые слова `program` и `version` не могут использоваться в качестве идентификаторов.
- Имя версии, а также номер версии не может включаться более 1 раза в области определения программы.
- Имя процедуры, а также номер процедуры не может включаться более 1 раза в области определения версии.
- Идентификаторы программ относятся к одному пространству имён с идентификаторами констант и типов.
- Номера программ, версий и процедур могут быть только беззнаковыми целыми числами.
- Имеющиеся компиляторы языка RPC обычно не поддерживают более одного спецификатора типа в списках аргументов процедур, обычно аргументы помещаются внутрь структуры.

13. Взаимодействие с IANA

Назначение номеров программ RPC, вариантов аутентификации и статуса аутентификации в прошлом выполнялось компанией Sun Microsystems, Inc (Sun). Это неприемлемо для протокола IETF Standards Track, поскольку такая работа отдана агентству по назначению номеров Internet (Internet Assigned Numbers Authority или IANA). Этот документ предлагает передать полномочия по выделению номеров программ RPC, вариантов аутентификации и статуса аутентификации от Sun Microsystems, Inc. в IANA и описать, как IANA будет поддерживать и выделять эти номера. Пользователи протоколов RPC выиграют от наличия независимого органа, отвечающего за назначение номеров.

13.1. Запрос номеров в IANA

В Приложении В указаны сведения, которые нужно направить в IANA для запроса одного или нескольких номеров RPC, и применяемые правила. IANA сохраняет запрос для документирования и включает в общедоступный реестр:

- краткое описание цели использования;
- выделенные номера программ;
- короткие идентификаторы (строки).

13.2. Защита прошлых назначений

Компания Sun выделила значения из пространства номеров программ RPC и вариантов аутентификации RPC с момента исходного развёртывания RPC. Назначения от Sun Microsystems остаются действительными и будут сохранены. Компания Sun сообщила в IANA все выделенные в обоих пространствах номера и процедура передачи на этом завершилась. Выделенные в настоящее время значения приведены в Приложении С. Текущие номера статусов аутентификации указаны в разделе 9 этого документа (определение `enum auth_stat`).

13.3. Назначение номеров RPC

Последующая практика IANA будет касаться разделения пространства 32-битовых номеров, как указано в параграфе 8.3. Подробные сведения о распределении указанных в параграфе 8.3 блоков приведены ниже.

13.3.1. Номера, выделяемые IANA

Первый блок администрирует IANA с защитой ранее выделенных компанией Sun значений. Эти значения относятся к десятичному диапазону 100000-399999 (0x000186a0 - 0x00061a7f), поэтому IANA начнёт назначение с десятичного номера 400000. Отдельные номера следует распределять в порядке запросов (First Come First Served), а блоки следует выделять по правилам, зависящим от размера блока.

13.3.2. Номера, выделяемые локальным администратором

Блок, распределяемый локальными администраторами, доступен для использования в любом административном домене, аналогично диапазонам частных адресов IP. Ожидаемое использование будет заключаться в создании локальным доменом «агентства» (authority) по назначению номеров из этого диапазона. Этому органу следует задать правила и процедуры назначения номеров RPC из данного диапазона. Домену следует быть достаточно изолированным, чтобы вероятность взаимодействия с приложениями RPC из других доменов была минимальна, поскольку иначе могут возникать конфликты номеров RPC и связанные с ними отказы приложений. В отсутствие локального администратора этот блок можно использовать в стиле Private Use, как указано в [RFC5226].

13.3.3. Временные номера

Блок временных (Transient) номеров может применяться любым приложением RPC по мере доступности. Этот диапазон предназначен для служб, которые могут сообщать динамически заданный номер программы RPC своим

клиентам с помощью любого подходящего механизма. Например, можно использовать общую память, если клиент и сервер размещаются в одной системе, или сетевое сообщение (RPC или иное), распространяющее выбранный номер.

Временный блок не администрируется. Службы RPC используют этот диапазон, выбирая номер и пытаясь зарегистрировать его в привязке RPC локальной системы (см. процедуры RPCBPROC_SET и PMAPPROC_SET в [RFC1833]). При успешной регистрации (другие службы RPC не используют этот номер) RPC Bindery назначает его запрашившему приложению RPC. Регистрация действует до завершения RPC, которое обычно происходит при перезагрузке системы, закрывающей все приложения, включая службу RPC, использовавшую временный номер. При отказе регистрации (номер занят другим приложением RPC) запрашивающий должен выбрать другой номер и повторить запрос. Для предотвращения конфликтов рекомендуется выбирать случайный номер из этого диапазона.

13.3.4. Резервный блок

Резервный блок (Reserved) сохранён на будущее. Приложениям RPC недопустимо использовать номера из этого блока, пока это не будет разрешено IESG.

13.3.5. Субблоки номеров RPC

Номера RPC обычно выделяются для конкретных служб RPC. Однако некоторым приложениям нужно несколько номеров RPC на одну службу. Наиболее распространенным примером является служба RPC, которой нужно иметь множество экземпляров одновременно на конкретном сайте. В протоколе RPC нет идентификатора экземпляра, поэтому нужен механизм мультиплексирования запросов RPC между экземплярами сервиса или каждый экземпляр должен использовать свой уникальный номер RPC.

В таких случаях с разными номерами может использоваться один или разные протоколы RPC. Номера могут назначаться приложением динамически или устанавливаться административным решением для сайта. По возможности, службам RPC с динамическим назначением номеров RPC следует использовать номера из блока временных номеров (параграф 13.3.3), а если это невозможно, можно запросить субблок номеров RPC.

Назначение субблоков номеров RPC зависит от размера запрашиваемого субблока и выполняется по процедуре Specification Required или IESG Approval из параграфа 4.1 в [RFC5226].

Размер субблока	Метод выделения	Ответственный
до 100 номеров	First Come First Served	IANA
до 1000 номеров	Specification Required	IANA
более 1000 номеров	IESG Approval	IESG

Отметим, что размер субблока может быть любым и в таблице приведены максимальные размеры разных категорий.

Субблоки размером до 100 номеров IANA может выделять в порядке поступления запросов (First Come First Served). Описание службы (RPC Service Description), включённое в диапазон должно указывать, как будет управляться субблок и, как минимум, должно быть указано, будет он применяться с одним или несколькими протоколами RPC, а также динамическое или статическое (административное) назначение адресов.

Субблоки размером до 1000 должны подробно документироваться, включая описание протокола или протоколов RPC, которые будут использоваться в диапазоне, а также описание использования номеров из субблока.

Субблоки размером более 1000 должны документироваться, как указано выше, а назначение должно происходить после одобрения IESG. Предполагается, что такие случаи будут редкими.

Для предотвращения множественных запросов больших блоков номеров предлагаются приведённые ниже правила.

Запросы с числом номеров до 100 (включительно) обрабатываются по процедуре First Come First Served. Пороговое значение 100 применяется к общему числу номеров RPC, выделяемых физическому или юридическому лицу. Например, если сначала запрошено 70 номеров, а затем то же лицо запрашивает ещё 40, запрос на 40 номеров будет обрабатываться по процедуре Specification Required. Пока общее число номеров не превышает 1000, IANA может не использовать процедуру Specification Required для дополнительных запросом с числом номеров меньше 100. Если физическое или юридическое лицо имеет менее 1000, а затем запрашивает дополнительный блок номеров так, что общее число превышает 1000, для дополнительного запроса требуется процедура IESG Approval.

13.4. Назначение номеров для вариантов аутентификации RPC

Второе пространство номеров предназначено для идентификаторов механизма аутентификации (flavor). Этот номер служит для выбора механизма проверки подлинности, который может применяться для сообщения RPC. Идентификатор используется в поле flavor структуры opaque_auth.

13.4.1. Правила назначения

В Приложении В указаны сведения, направляемые в IANA для запроса одного или нескольких номеров аутентификации RPC и правила выделения номеров. IANA сохраняет запрос для документирования и включает в общедоступный реестр:

- короткие идентификаторы (строки);
- выделенные номера аутентификации;
- краткое описание цели использования.

13.4.2. Варианты и псевдо-варианты аутентификации

Тенденции защиты RPC отошли от создания новых вариантов аутентификации, применяемых в AUTH_DH [DH], и сосредоточились на использовании имеющегося варианта RPCSEC_GSS [RFC2203] и создании новых механизмов GSS-API¹ для него. Хотя RPCSEC_GSS является выбранным вариантом аутентификации, применение нового механизма RPCSEC_GSS с сетевой файловой системой (Network File System или NFS) [RFC1094] [RFC1813],

¹Generic Security Services Application Programming Interface - базовый интерфейс прикладных программ служб защиты.

[RFC3530] требует регистрации псевдо-вариантов (pseudo-flavor), используемых для однозначного согласования механизмов защиты, как указано в [RFC2623]. Имеющиеся псевдо-варианты представлены в диапазоне 390000-390255, а новые будут выделяться IANA из этого же блока по процедуре First Come First Served.

Для запросов, не связанных с псевдо-вариантами, IANA будет выделять номера вариантов аутентификации RPC, начиная с 400000 по процедуре First Come First Served, чтобы избежать конфликтов с уже выделенными номерами.

Для вариантов аутентификации или механизмов RPCSEC_GSS, применяемых в Internet, настоятельно рекомендуется публиковать RFC категории Informational или Standards Track с описанием поведения и параметров механизма аутентификации.

13.5. Выделение номеров для статуса аутентификации

Ещё одно пространство номеров предназначено для значений статуса аутентификации auth_stat, описывающих природу проблем, возникших при проверке подлинности, или подтверждения аутентификации. Полный исходный список значений приведён в разделе 9 (auth_stat). Предполагается, что новые значения будут добавляться редко, но время от времени новые варианты аутентификации могут приводить к расширению списка значений. Номера следует выделять по процедуре First Come First Served, чтобы избежать конфликтов с уже выделенными значениями.

13.5.1. Правила назначения

В Приложении В указаны сведения, направляемые в IANA для запроса одного или нескольких значений auth_stat и правила выделения номеров. IANA сохраняет запрос для документирования и включает в общедоступный реестр:

- короткие идентификаторы (строки);
- выделенные номера auth_stat;
- краткое описание цели использования.

14. Вопросы безопасности

Системная аутентификация (AUTH_SYS), описанная в Приложении А, считается небезопасной из-за отсутствия верификатора, позволяющего проверить свидетельство (credential). AUTH_SYS **не следует** применять для служб, позволяющих клиенту изменять данные. Метод AUTH_SYS **недопустимо** указывать как **рекомендуемый** или **требуемый** для любой службы RPC со статусом Standards Track.

Метод AUTH_DH, как отмечено в параграфах 8.2 и 13.4.2, считается устаревшим и небезопасным (см. [RFC2695]). AUTH_DH **не следует** применять для служб, позволяющих клиенту изменять данные. Метод AUTH_DH **недопустимо** указывать как **рекомендуемый** или **требуемый** для любой службы RPC со статусом Standards Track.

В [RFC2203] определён новый вариант защиты - RPCSEC_GSS, позволяющий использовать механизмы GSS-API [RFC2743] для защиты RPC. Всем нетривиальным программам RPC, разрабатываемым впредь, следует соответствующим образом реализовать защиту на основе RPCSEC_GSS. В [RFC2623] описано, как это сделано в широко распространённой программе RPC.

Службы RPC со статусом Standards Track **должны** требовать поддержку RPCSEC_GSS, а также **должны** требовать поддержку псевдо-варианта аутентификации с подходящим уровнем безопасности в зависимости от потребности в проверке подлинности, защите целостности (т. е. неотказуемости) и приватности данных.

Приложение А. Системная аутентификация

Клиент может пожелать идентифицировать себя, как это делается в системе UNIX(tm). Этот вариант свидетельства клиента называется AUTH_SYS. Неинтерпретируемые (opaque) данные свидетельства представляют собой структуру

```
struct authsys_parms {
    unsigned int stamp;
    string machinename<255>;
    unsigned int uid;
    unsigned int gid;
    unsigned int gids<16>;
};
```

Поле stamp содержит произвольный идентификатор, который может генерировать вызывающая машина, machinename указывает имя этой машины (например, krypton). Поле uid содержит идентификатор вызывающего пользователя, gid - идентификатор действующей группы, а gids содержит массив идентификаторов групп, в которые входит вызывающий пользователь. Верификатору, сопровождающему свидетельство (credential), следует иметь значение AUTH_NONE (см. выше). Отметим, что свидетельство будет уникальным лишь в рамках отдельного домена имён машин, uid и gid.

Значением верификатора в отклике сервера может быть AUTH_NONE или AUTH_SHORT. В случае AUTH_SHORT байты строки верификатора содержат неинтерпретируемую (opaque) структуру, которая может быть передана серверу вместо исходного свидетельства варианта AUTH_SYS. Сервер может хранить кэш, сопоставляющий сокращённые opaque-структуры (передаются обратно в стиле верификатора отклика AUTH_SHORT) с исходными свидетельствами вызывающего. Вызывающий может сэкономить пропускную способность и ресурсы серверного процессора, используя сокращённое свидетельство. Сервер может в любой момент сбросить сокращённую opaque-структуру. В этом случае сообщение с вызовом удалённой процедуры будет отвергнуто с ошибкой аутентификации (причина AUTH_REJECTEDCRED). Тогда клиент может попытаться применить исходное свидетельство AUTH_SYS.

Следует отметить, что использование этого варианта аутентификации само по себе не обеспечивает какой-либо защиты для пользователя и поставщика услуги. Обеспечиваемую такой схемой проверку подлинности можно считать легитимной лишь при наличии внешней защиты использующих схему приложений и сети, а также использовании привилегированных транспортных адресов на взаимодействующих конечных точках (примером может служить использование привилегированных портов TCP/UDP в системах UNIX, но не все системы реализуют механизмы привилегированных транспортных адресов).

Приложение В. Запрос связанных с RPC номеров в IANA

Номера программ RPC, вариантов и статуса аутентификации, которые должны быть уникальными во всех сетях, выделяются Internet Assigned Number Authority (IANA). Для запроса одного или блока номеров нужно отправить в IANA по адресу iana@iana.org запрос, содержащий указанные ниже сведения.

- Тип запрашиваемых номеров (для программ, вариантов аутентификации или кодов аутентификации).
- Число запрашиваемых номеров.
- Имя физического лица или название компании, которая будет использовать номера.
- Строка идентификатора, связывающая номера со службой.
- Адрес электронной почты для контактов со службой, использующей номера.
- Краткое описание цели использования номеров.
- При запросе номеров для вариантов аутентификации и псевдо-вариантов для использования с RPCSEC_GSS и NFS, отображения, аналогичные приведённым в параграфе 4.2 [RFC2623].

Запросить конкретные номера нельзя, они распределяются по порядку (First Come First Served).

Для всех вариантов и статусов аутентификации RPC, применяемых в Internet, настоятельно рекомендуется публикация RFC категории Informational или Standards Track с описанием поведения и параметров механизма аутентификации.

Приложение С. Выделенные номера

#		
#	Выделенные Sun номера RPC	
#		
#	Описание и владелец	Номер RPC Краткое имя
#	-----	-----
portmapper	100000	pmapprog portmap rpcbind
remote stats	100001	rstatprog
remote users	100002	rusersprog
nfs	100003	nfs
yellow pages (NIS)	100004	ypprog ypserv
mount demon	100005	mountprog
remote dbx	100006	dbxprog
yp binder (NIS)	100007	ypbindprog ypbind
shutdown msg	100008	wall
yppasswd server	100009	yppasswdprog yppasswdd
ether stats	100010	etherstatprog
disk quotas	100011	rquota
spray packets	100012	spray
3270 mapper	100013	ibm3270prog
RJE mapper	100014	ibmrjprog
selection service	100015	selnsvcprog
remote database access	100016	rdatabaseprog
remote execution	100017	rexec
Alice Office Automation	100018	aliceprog
scheduling service	100019	schedprog
local lock manager	100020	lockprog llockmgr
network lock manager	100021	netlockprog nlockmgr
x.25 inr protocol	100022	x25prog
status monitor 1	100023	statmon1
status monitor 2	100024	statmon2
selection library	100025	selnlibprog
boot parameters service	100026	bootparam
mazewars game	100027	mazeprog
yp update (NIS)	100028	ypupdateprog ypupdate
key server	100029	keyserveprog
secure login	100030	securecmdprog
nfs net forwarder init	100031	netfwdiprog
nfs net forwarder trans	100032	netfwdtprog
sunlink MAP	100033	sunlinkmap
network monitor	100034	netmonprog
lightweight database	100035	dbaseprog
password authorization	100036	pwdauthprog
translucent file svc	100037	tfsprog
nse server	100038	nseprog
nse activate daemon	100039	nse_activate_prog
sunview help	100040	sunview_help_prog
pnf install	100041	pnf_prog
ip addr allocator	100042	ipaddr_alloc_prog
show filehandle	100043	filehandle
MVS NFS mount	100044	mvsnfsprog
remote user file operations	100045	rem_fileop_user_prog
batched ypupdate	100046	batch_ypupdateprog
network execution mgr	100047	nem_prog
raytrace/mandelbrot remote daemon	100048	raytrace_rd_prog
raytrace/mandelbrot local daemon	100049	raytrace_ld_prog
remote group file operations	100050	rem_fileop_group_prog
remote system file operations	100051	rem_fileop_system_prog
remote system role operations	100052	rem_system_role_prog
gpd lego fb simulator	100053	[неизвестно]

gpd simulator interface	100054	[неизвестно]
ioadmd	100055	ioadmd
filemerge	100056	filemerge_prog
Name Binding Program	100057	namebind_prog
sunlink NJE	100058	njeprog
MVSNFS get attribute service	100059	mvsattrprog
SunAccess/SunLink resource manager	100060	rmgrprog
UID allocation service	100061	uidallocprog
license broker	100062	lbserverprog
NETlicense client binder	100063	lbbinderprog
GID allocation service	100064	gidallocprog
SunIsam	100065	sunisamprog
Remote Debug Server	100066	rdbsrvprog
Network Directory Daemon	100067	[неизвестно]
Network Calendar Program	100068	cmsd cm
ypxfrd	100069	ypxfrd
rpc.timed	100070	timedprog
bugtraqd	100071	bugtraqd
	100072	[неизвестно]
Connectathon Billboard - NFS	100073	[неизвестно]
Connectathon Billboard - X	100074	[неизвестно]
Sun tool for scheduling rooms	100075	schedroom
Authentication Negotiation	100076	authnegotiate_prog
Database manipulation	100077	attribute_prog
Kerberos authentication daemon	100078	kerbprog
Internal testing product (no name)	100079	[неизвестно]
Sun Consulting Special	100080	autodump_prog
Event protocol	100081	event_svc
bugtraq_qd	100082	bugtraq_qd
ToolTalk and Link Service Project	100083	database service
Consulting Services	100084	[неизвестно]
Consulting Services	100085	[неизвестно]
Consulting Services	100086	[неизвестно]
Jupiter Administration	100087	adm_agent admin
Dual Disk support	100090	libdsd/dsd
DocViewer 1.1	100091	[неизвестно]
ToolTalk	100092	remote_activation_svc
Consulting Services	100093	host_checking
SNA peer-to-peer	100094	[неизвестно]
Roger Riggs	100095	searchit
Robert Allen	100096	msgtool
SNA	100097	[неизвестно]
SISU	100098	networked version of CS5
NFS Automount File System	100099	autofs
	100100	msgboard
event dispatching agent [eventd]	100101	netmgt_eventd_prog
statistics/event logger [netlogd]	100102	netmgt_netlogd_prog
topology display manager [topology]	100103	netmgt_topology_prog
syncstat agent [syncstatd]	100104	netmgt_syncstatd_prog
ip packet stats agent [ippktd]	100105	netmgt_ippktd_prog
netmgt config agent [configd]	100106	netmgt_configd_prog
restat agent [restatd]	100107	netmgt_restatd_prog
lpq agent [lprstatd]	100108	netmgt_lprstatd_prog
netmgt activity agent [mgtlogd]	100109	netmgt_mgtlogd_prog
proxy DECnet NCP agent [proxydni]	100110	netmgt_proxydni_prog
topology mapper agent [mapperd]	100111	netmgt_mapperd_prog
netstat agent [netstatd]	100112	netmgt_netstatd_prog
sample netmgt agent [sampled]	100113	netmgt_sampled_prog
X.25 statistics agent [vcstatd]	100114	netmgt_vcstatd_prog
Frame Relay	100128	[неизвестно]
PPP agent	100129	[неизвестно]
localhad	100130	rpc.localhad
layers2	100131	na.layers2
token ring agent	100132	na.tr
related to lockd and statd	100133	nsm_addr
Kerberos project	100134	kwarn
etherif2	100135	na.etherif2
hostmem2	100136	na.hostmem2
iostat2	100137	na.iostat2
snmpv2	100138	na.snmpv2
Cooperative Console	100139	cc_sender
na.cputat	100140	na.cputat
Sun Cluster SC3.0	100141	rgmd_receptionist
	100142	fed
Network Storage	100143	rdc
Sun Cluster products	100144	nafo
SunCluster 3.0	100145	scadmd
ASN.1	100146	amiserv
	100147	amiaux # BER и DER, кодирование и декодирование
Delegate Management Server	100148	dm
	100149	rkstat
	100150	ocfserv
	100151	sccheckd
	100152	autoclientd
	100153	sunvts

	100154	ssmond
	100155	smserverd
	100156	test1
	100157	test2
	100158	test3
	100159	test4
	100160	test5
	100161	test6
	100162	test7
	100163	test8
	100164	test9
	100165	test10
	100166	nfsmapid
	100167	SUN_WBEM_C_CIMON_HANDLE
	100168	sacmmd
	100169	fmd_adm
	100170	fmd_api
	100171	[неизвестно]
	100172	idmapd
не выделены	100173 - 100174	
snmptrap	100175	na.snmptrap
не выделены	100176-100199	
не выделены	100200	
MVS/NFS Memory usage stats server	100201	[неизвестно]
Netapp	100202-100207	
не выделены	100208-100210	
8.0 SunLink SNA RJE	100211	[неизвестно]
8.0 SunLink SNA RJE	100212	[неизвестно]
	100213	ShowMe
	100214	[неизвестно]
	100215	[неизвестно]
AUTH_RSA Key service	100216	keyrsa
SunSelect PC license service	100217	[неизвестно]
WWCS (Corporate)	100218	sunsolve
	100219	cstatd
X/Open Federated Naming	100220	xfn_server_prog
Kodak Color Management System	100221	kcs_network_io kcs
HA-DBMS	100222	ha_dbms_serv
	100223-100225	[неизвестно]
	100226	hafaultd
NFS ACL Service	100227	nfs_acl
distributed lock manager	100228	dlmd
	100229	metad
	100230	metamhd
	100231	nfsauth
	100232	sadmind
	100233	ufsd
	100234	grpservd
	100235	cachefs
	100236	msmprog Media_Server
	100237	ihnamed
	100238	ihneta
	100239	ihsecured
	100240	ihclassmgrd
	100241	ihrepositoryd
	100242	metamedd rpc.metamedd
	100243	contentmanager cm
	100244	symon
	100245	pld genesil
	100246	ctid
		cluster_transport_interface
	100247	ccd
		cluster_configuration_db
	100248	pmfd
	100249	dmi2_client
	100250	mfs_admin
	100251	ndshared_unlink
	100252	ndshared_touch
	100253	ndshared_slink
	100254	cbs control_board_server
	100255	skiserv
	100256	nfsxa nfsxattr
	100257	ndshared_disable
	100258	ndshared_enable
	100259	sms_account_admin
	100260	sms_modem_admin
	100261	sms_r_login
	100262	sms_r_subaccount_mgt
	100263	sms_service_admin
	100264	session_admin
	100265	canci_ancs_program
	100266	canci_sms_program
	100267	msmp
	100268	halck
	100269	halogmsg

	100270	nfs_id_map
	100271	ncall
	100272	hmip
	100273	repl_mig
	100274	repl_mig_cb
NIS+	100300	nisplus
NIS+	100301	nis_cachemgr
NIS+ call back protocol	100302	[неизвестно]
NIS+ Password Update Daemon	100303	nispasswd
FNS context update in NIS	100304	fnsypd
	100305	[неизвестно]
	100306	[неизвестно]
	100307	[неизвестно]
	100308	[неизвестно]
	100309	[неизвестно]
не выделены	100310 - 100398	
nfscsum	100399	nfscsum
network utilization agent	100400	netmgt_netu_prog
network rpc ping agent	100401	netmgt_rping_prog
	100402	na.shell
picsprint	100403	na.picslp
	100404	traps
	100405 - 100409	[неизвестно]
	100410	jdsagent
	100411	na.haconfig
	100412	na.halhost
	100413	na.hadtsrv
	100414	na.hamdstat
	100415	na.neoadmin
	100416	ex1048prog
rdmaconfig	100417	rpc.rdmaconfig
IETF NFSv4 Working Group - FedFS	100418 - 100421	
	100422	mdcomm
	100423	kiprop krb5_iprop
	100424	stsf
не выделены	100425 - 100499	
Sun Microsystems	100500 - 100531	[неизвестно]
	100532	ucmmstate
	100533	scrcmd
не выделены	100534 - 100999	
nse link daemon	101002	nselinktool
nse link application	101003	nselinkapp
unassigned	101004 - 101900	
	101901	[неизвестно]
не выделены	101902 - 101999	
AssetLite	102000	[неизвестно]
PagerTool	102001	[неизвестно]
Discover	102002	[неизвестно]
не выделены	102003 - 105000	
ShowMe	105001	sharedapp
Registry	105002	REGISTRY_PROG
Print-server	105003	print-server
Proto-server	105004	proto-server
Notification-server	105005	notification-server
Transfer-agent-server	105006	transfer-agent-server
не выделены	105007 - 110000	
	110001	tsolrpcb
	110002	tsolpeerinfo
	110003	tsolboot
	120001	cmip na.cmip
	120002	na.osidiscover
	120003	cmiptrap
не выделены	120004 - 120099	
	120100	eserver
	120101	repserver
	120102	swserver
	120103	dmd
	120104	ca
не выделены	120105 - 120125	
	120126	nf_fddi
	120127	nf_fddismt7_2
не выделены	120128 - 150000	
pc passwd authorization	150001	pcnfsdprog
TOPS name mapping	150002	[неизвестно]
TOPS external attribute storage	150003	[неизвестно]
TOPS hierarchical file system	150004	[неизвестно]
TOPS NFS transparency extensions	150005	[неизвестно]
PC NFS License	150006	pcnfslicense
RDA	150007	rdaprog
WabiServer	150008	wsprog
WabiServer	150009	wsrlprog
не выделены	150010 - 160000	
	160001	nihon-cm
	160002	nihon-ce
не выделены	160003 - 170099	

	170100	domf_daemon0
	170101	domf_daemon1
	170102	domf_daemon2
	170103	domf_daemon3
	170104	domf_daemon4
	170105	domf_daemon5
не выделены	170106 - 179999	
	180000	cecprog
	180001	cecsysprog
	180002	cec2cecprog
	180003	cesprog
	180004	ces2cesprog
	180005	cet2cetprog
	180006	cet2cetdoneprog
	180007	cetcompprog
	180008	cetsysprog
	180009	cghapresenceprog
	180010	cgdmnsyncprog
	180011	cgdmcnsccliprog
	180012	cgdmcrsccliprog
	180013	cgdmcrssvcproG
	180014	chmprog
	180015	chmsysprog
	180016	crsapiprog
	180017	ckptmprog
	180018	crimcomponentprog
	180019	crimqueryprog
	180020	crimsecondaryprog
	180021	crimservicesprog
	180022	crimsyscomponentprog
	180023	crimsysservicesprog
	180024	csmagtapiproG
	180025	csmagtcallbackprog
	180026	csmreplicaproG
	180027	csmsrvprog
	180028	cssccltprog
	180029	csscsvrprog
	180030	csscopresultprog
не выделены	180031 - 199999	
	200000	pyramid_nfs
	200001	pyramid_reserved
	200002	cadds_image
	200003	stellar_name_prog
	200004	[неизвестно]
	200005	[неизвестно]
	200006	pacl
	200007	lookupids
	200008	ax_statd_prog
	200009	ax_statd2_prog
	200010	edm
	200011	dtedirwd
	200012	[неизвестно]
	200013	[неизвестно]
	200014	[неизвестно]
	200015	[неизвестно]
	200016	easerpcd
	200017	rlxnfs
	200018	sascuiddprog
	200019	knfsd
	200020	ftnfsd ftnfsd_program
	200021	ftsyncd ftsyncd_program
	200022	ftstatd ftstatd_program
	200023	exportmap
	200024	nfs_metadata
не выделены	200025 - 200200	
	200201	ecoad
	200202	eamon
	200203	ecolic
	200204	cs_printstatus_svr
	200205	ecodisc
не выделены	200206 - 300000	
	300001	adt_rflockprog
	300002	columbinel
	300003	system33_prog
	300004	frame_prog1
	300005	uimxprog
	300006	rvd
	300007	entombing daemon
	300008	account mgmt system
	300009	frame_prog2
	300010	beeper access
	300011	dptuprog
	300012	mx-bcp
	300013	instrument-file-access
	300014	file-system-statistics
	300015	unify-database-server

300016	tmd_msg
300017	[неизвестно]
300018	[неизвестно]
300019	automounter access
300020	lock server
300021	[неизвестно]
300022	office-automation-1
300023	office-automation-2
300024	office-automation-3
300025	office-automation-4
300026	office-automation-5
300027	office-automation-6
300028	office-automation-7
300029	local-data-manager
300030	chide
300031	csi_program
300032	[неизвестно]
300033	online-help
300034	case-tool
300035	delta
300036	rgi
300037	instrument-config-server
300038	[неизвестно]
300039	[неизвестно]
300040	dtia-rpc-server
300041	cms
300042	viewer
300043	aqm
300044	exclaim
300045	masterplan
300046	fig_tool
300047	[неизвестно]
300048	[неизвестно]
300049	[неизвестно]
300050	remote-lock-manager
300051	[неизвестно]
300052	gdebug
300053	ldebug
300054	rscanner
300055	[неизвестно]
300056	[неизвестно]
300057	[неизвестно]
300058	[неизвестно]
300059	[неизвестно]
300060	[неизвестно]
300061	[неизвестно]
300062	[неизвестно]
300063	[неизвестно]
300064	[неизвестно]
300065	[неизвестно]
300066	nSERVER
300067	[неизвестно]
300068	[неизвестно]
300069	[неизвестно]
300070	[неизвестно]
300071	BioStation
300072	[неизвестно]
300073	NetProb
300074	Logging
300075	Logging
300076	[неизвестно]
300077	[неизвестно]
300078	[неизвестно]
300079	[неизвестно]
300080	[неизвестно]
300081	[неизвестно]
300082	sw_twin
300083	remote_get_login
300084	odcprog
300085	[неизвестно]
300086	[неизвестно]
300087	[неизвестно]
300088	[неизвестно]
300089	[неизвестно]
300090	[неизвестно]
300091	smartdoc
300092	superping
300093	distributed-chembench
300094	uacman/alfil-uacman
300095	ait_rcagent_prog
300096	ait_rcagent_appl_prog
300097	smart
300098	ecoprogram
300099	leonardo
300100	[неизвестно]
300101	[неизвестно]

300102	[неизвестно]
300103	[неизвестно]
300104	[неизвестно]
300105	[неизвестно]
300106	[неизвестно]
300107	[неизвестно]
300108	wingz
300109	teidan
300110	[неизвестно]
300111	[неизвестно]
300112	[неизвестно]
300113	[неизвестно]
300114	[неизвестно]
300115	[неизвестно]
300116	cadc_fhlockprog
300117	highscan
300118	[неизвестно]
300119	[неизвестно]
300120	[неизвестно]
300121	opennavigator
300122	aarpcxfer
300123	[неизвестно]
300124	[неизвестно]
300125	[неизвестно]
300126	groggs
300127	licsrv
300128	issdemon
300129	[неизвестно]
300130	maximize
300131	cgm_server
300132	[неизвестно]
300133	agent_rpc
300134	docmaker
300135	docmaker
300136	[неизвестно]
300137	[неизвестно]
300138	[неизвестно]
300139	iesx
300140	[неизвестно]
300141	[неизвестно]
300142	[неизвестно]
300143	[неизвестно]
300144	smart-mbs
300145	[неизвестно]
300146	[неизвестно]
300147	docimage
300148	[неизвестно]
300149	dmc-interface
300150	[неизвестно]
300151	jss
300152	[неизвестно]
300153	arimage
300154	xdb-workbench
300155	frontdesk
300156	dmc
300157	expressight-6000
300158	graph service program
300159	[неизвестно]
300160	[неизвестно]
300161	[неизвестно]
300162	[неизвестно]
300163	[неизвестно]
300164	[неизвестно]
300165	[неизвестно]
300166	[неизвестно]
300167	[неизвестно]
300168	[неизвестно]
300169	[неизвестно]
300170	[неизвестно]
300171	[неизвестно]
300172	[неизвестно]
300173	[неизвестно]
300174	[неизвестно]
300175	[неизвестно]
300176	rlpr
300177	nx_hostdprog
300178	netuser-x
300179	rmntprog
300180	[неизвестно]
300181	mipe
300182	[неизвестно]
300183	collectorprog
300184	uslookup_PROG
300185	viewstation
300186	iate
300187	[неизвестно]

300188	[неизвестно]
300189	[неизвестно]
300190	imsvtprog
300191	[неизвестно]
300192	[неизвестно]
300193	[неизвестно]
300194	pmdb
300195	pmda
300196	[неизвестно]
300197	[неизвестно]
300198	trend_idbd
300199	rres
300200	sd.masterd
300201	sd.executiond
300202	sd.listend
300203	sd.reserve1
300204	sd.reserve2
300205	msbd
300206	stagedprog
300207	mountprog
300208	watchdprog
300209	pms
300210	[неизвестно]
300211	session_server_program
300212	session_program
300213	debug_serverprog
300214	[неизвестно]
300215	[неизвестно]
300216	pacprog
300217	[неизвестно]
300218	mbus
300219	aframes2ps
300220	npartprog
300221	cmlserver
300222	cmlbridge
300223	sailfrogfaxprog
300224	sailfrogphoneprog
300225	sailfrogvmailprog
300226	wserviceprog arcstorm
300227	hld
300228	alive
300229	radsp
300230	radavx
300231	radview
300232	rsys_prog
300233	rsys_prog
300234	fm_rpc_prog
300235	aries
300236	uapman
300237	ddman
300238	top
300239	[неизвестно]
300240	trendlink
300241	licenseprog
300242	statuslicenseprog
300243	oema_rmpf_svc
300244	oema_smpf_svc
300245	oema_rmsg_svc
300246	grapes-sd
300247	ds_master
300248	ds_transfer
300249	ds_logger
300250	ds_query
300251	[неизвестно]
300252	[неизвестно]
300253	nsd_prog
300254	browser
300255	epoch
300256	floorplanner
300257	reach
300258	tactic
300259	cachescientific1
300260	cachescientific2
300261	desksrc_prog
300262	photo3d1
300263	photo3d2
300264	[неизвестно]
300265	soundmgr
300266	s6k
300267	aims_referenced_ text_processor
300268	xess
300269	ds_queue
300270	[неизвестно]
300271	orionscanplus
300272	openlink-xx

300273	kbmsprog
300274	[неизвестно]
300275	futuresource
300276	the_xprt
300277	cmg_srvprog
300278	[неизвестно]
300279	[неизвестно]
300280	front
300281	[неизвестно]
300282	[неизвестно]
300283	[неизвестно]
300284	conmanprog
300285	jincv2
300286	isls
300287	systemstatprog
300288	fxpsprog
300289	callpath
300290	axess
300291	armor_rpcd
300292	armor_dictionary_rpcd
300293	armor_miscd
300294	filetransfer_prog
300295	bl_swda
300296	bl_hwda
300297	[неизвестно]
300298	[неизвестно]
300299	[неизвестно]
300300	filemon
300301	acunetprog
300302	rbuild
300303	assistprog
300304	tog
300305	[неизвестно]
300306	sns7000
300307	igprog
300308	tgprog
300309	plc
300310	pxman pxlsprog
300311	hde_server hdeserver
300312	tsslicenseprog
300313	rpc.explorerd
300314	chrd
300315	tbisam
300316	tbis
300317	adsprog
300318	sponsorprog
300319	querycmprog
300320	[неизвестно]
300321	[неизвестно]
300322	mobill
300323	sld
	service_locator_daemon
300324	linkprog
300325	codexdaemonprog
300326	drprog
300327	ressys_commands
300328	stamp
300329	matlab
300330	schedld
300331	upcprog
300332	xferbkch
300333	xfer
300334	qbthd
300335	qbabort
300336	lsd
300337	geomgrd
300338	generic_fts
300339	ft_ack
300340	lymb
300341	vantage
300342	cltstd clooptstdprog
300343	clui clui_prog
300344	testerd tstdprog
300345	extsim
300346	cmd_dispatch maxx_ems
300347	callpath_receive_program
300348	x3270prog
300349	sbc_lag
300350	sbc_frfa
300351	sbc_frs
300352	atommgr
300353	geostrat
300354	dbvialu6.2
300355	[неизвестно]
300356	fxncprog
300357	infopolic

	300358	[неизвестно]
	300359	aagms
	300360	aagms
	300361	[неизвестно]
	300362	clariion_mgr
	300363	setcimrpc
	300364	virtual_protocol_adapter
	300365	unibart
	300366	uniarch
	300367	unifile
	300368	unisrex
	300369	uniscmd
	300370	rsc
	300371	set
	300372	desaf-ws/key
	300373	reelddb
	300374	nl
	300375	rmd
	300376	agcd
	300377	rsynd
	300378	rcnlib
	300379	rcnlib_attach
	300380	evergreen_mgmt_agent
	300381	fx104prog
	300382	rui
		remote_user_interface
	300383	ovomd
	300384	[неизвестно]
	300385	[неизвестно]
	300386	system_server
	300387	pipecs_cs_pipeprog
		ppktrpc
	300388	uv-net univision
	300389	auexe
	300390	audip
	300391	mqi
	300392	eva
	300393	eeee_reserved_1
	300394	eeee_reserved_2
	300395	eeee_reserved_3
	300396	eeee_reserved_4
	300397	eeee_reserved_5
	300398	eeee_reserved_6
	300399	eeee_reserved_7
	300400	eeee_reserved_8
	300401	cprrlm
	300402	wg_idms_manager
	300403	timequota
	300404	spiff
	300405-300414	ov_oem_svc
	300415	ov_msg_ctlg_svc
	300416	ov_advt_reg_svc
	300417-300424	showkron
	300425	daatd
	300426	swiftnet
	300427	ovomdel
	300428	ovomreq
	300429	msg_dispatcher
	300430	pcshare server
	300431	rcvs
	300432	fdfserver
	300433	bssd
	300434	drdd
	300435	mif_gutsprog
	300436	mif_guiprog
	300437	twolfd
	300438	twscd
	300439	nwsbumv
	300440	dgux_mgr
	300441	pfxd
	300442	tds
	300443	ovomadmind
	300444	ovomgate
	300445	omadmind
	300446	nps
	300447	npd
	300448	tsa
	300449	cdaimc
не выделены	300450-300452	
	300453	ckt_implementation
	300454	mda-tactical
не выделены	300455-300458	
	300459	attrun
	300460	RoadRunner
	300461	nas
	300462	undelete

300463	ovacadd
300464	tbdesmai
300465	arguslm
300466	dmd
300467	drd
300468	fm_help
300469	ftransrpc_prog
300470	finrisk
300471	dg_pc_idisched
300472	dg_pc_idiserv
300473	apd
300474	ap_sspd
300475	callpatheventrecorder
300476	flc
300477	dg_osm
300478	dspnamed
300479	iqddsrv
300480	iqjobsrv
300481	tacosxx
300482	wheeldbmg
300483	cnxmgr_nm_prog
300484	cnxmgr_cfg_prog
300485	3dsmapper
300486	ids
300487	imagine_rpc_svc
300488	lfn
300489	salesnet
300490	defaxo
300491	dbqtsd
300492	kms
300493	rpc.iced
300494	calc2s
300495	ptoidprog
300496	docs1s
300497	new
300498	collagebdg
300499	ars_server
300500	ars_client
300501	vr_catalog
300502	vr_tdb
300503	ama
300504	evama
300505	conama
300506	service_process
300507	reuse_proxy
300508	mars_ctrl
300509	mars_db
300510	mars_com
300511	mars_admch
300512	tbpicip
300513	top_acs_svc
300514	inout_svc
300515	csoft_wp
300516	mcfs
300517	eventprog
300518	dg_pc_idimsg
300519	dg_pc_idiaux
300520	atsr_gc
300521	alarm alarm_prog
300522	fts_prog
300523	dcs_prog
300524	ihb_prog
300525	[неизвестно]
300526	[неизвестно]
300527	clu_info_prog
300528	rmfm
300529	c2sdocd
300530	interahelp
300531	callpathasynmsgshandler
300532	optix_arc
300533	optix_ts
300534	optix_wf
300535	maxopenc
300536	cev cev_server
300537	sitewideprog
300538	drs
300539	drsdm
300540	dasgate
300541	dcdbd
300542	dcpsd
300543	supportlink_prog
300544	broker
300545	listner
300546	multiaccess
300547	spai_interface
300548	spai_adaption

300549 chimera_ci
chimera_clientinterface
300550 chimera_pi
chimera_processinvoker
300551 teamware_fl
teamware_foundationlevel
300552 teamware_sl
teamware_systemlevel
300553 teamware_ui
teamware_userinterface
300554 lprm
300555 mpsprog
Mensuration_Proxy_Server
300556 mo_syndis
300557 retsideprog
300558 slp
300559 slm-api
300560 im_rpc teamconference
300561 license_prog license
300562 stuple stuple_prog
300563 upasswd_prog
300564 gentranmentorsecurity
300565 gentranmentorprovider
300566 latituded
latitude_license_server
300567 gentranmentorreq1
300568 gentranmentorreq2
300569 gentranmentorreq3
300570 rj_server
300571 gws-rdb
300572 gws-mpmd
300573 gws-spmd
300574 vwcalcd
300575 vworad
300576 vwsybd
300577 vwave
300578 online_assistant
300579 internet_assistant
300580 spawnnd
300581 procmgrg
300582 cfgdbd
300583 logutild
300584 ibis
300585 ibisaux
300586 aapi
300587 rstrt
300588 hbeat
300589 pcspu
300590 empress
300591 sched_server
LiveScheduler
300592 path_server
LiveScheduler
300593 c2sdmd
300594 c2scf
300595 btsas
300596 sdtas
300597 appie
300598 dmi
300599 pscd
panther software corp daemon
300600 sisd
300601 cpwebserver
300602 wwcommo
300603 mx-mie
300604 mx-mie-debug
300605 idmn
300606 ssrv
300607 vpnserver
300608 samserver
300609 sams_server
300610 chrysalis
300611 ddm
300612 ddm-is
300613 mx-bcp-debug
300614 upmrd
300615 upmdsd
300616 res
300617 colortron
300618 zrs
300619 afpsrv
300620 apxft
300621 nrp
300622 hpid
300623 mailwatch
300624 fos bc_fcrb_receiver

300625	cs_sysadmin_svr
300626	cs_controller_svr
300627	nokia_nms_eai
300628	dbg
300629	remex
300630	cs_bind
300631	idm
300632	prpasswd
300633	iw-pw
300634	starrb
300635	Impress_Server
300636	colorstar
300637	gwugui
300638	gwsGui
300639	dai_command_proxy
300640	dai_alarm_server
300641	dai_fui_proxy
300642	spai_command_proxy
300643	spai_alarm_server
300644	iris
300645	hcxttp
300646	updatedb rsched
300647	urnd urn
300648	iqwpsrv
300649	dskutild
300650	online
300651	nlserv
300652	acsm
300653	dg_clar_sormsg
300654	wwpollerrpc
300655	wwmodelrpc
300656	nsprofd
300657	nsdstd
300658	recollect
300659	lssexecd lss_res
300660	lssagend lss_rea
300661	cdinfo
300662	sninsr_addon
300663	mm-sap
300664	ks
300665	psched
300666	tekdvfs
300667	storxll
300668	nisse
300669	lbadvise
300670	atcinstaller
300671	atntstarter
300672	NetML
300673	tdmesmge
300674	tdmesmgd
300675	tdmesmgt
300676	olm
300677	mediamanagement
300678	rdbprog fieldowsrv
300679	rpwdprog rpwd
300680	sapi-trace
300681	sapi-master-daemon
300682	omdcuprog om-dcu
300683	wwprocmon
300684	tndidprog
300685	rkey_setsecretprog
300686	asdu_server_prog
300687	pwrctrl
300688	siunixd
300689	wmapi
300690	cross_reference_ole
300691	rtc
300692	disp
300693	sql_compilation_agent
300694	tnsysprog
300695	ius-sapimd
300696	apteam-dx
300697	rmsrpc
300698	seismic_system
300699	remote
300700	ttl_ts_event nokia_nms
300701	fxrs
300702	onlicense
300703	vxkey
300704	dinis
300705	sched2d schedule-2
300706	sched3d schedule-3
300707	sched4d schedule-4
300708	sched5d schedule-5
300709	sched6d schedule-6
300710	sched7d schedule-7

	300711	sched8d schedule-8
	300712	sched9d schedule-9
	300713	adtsqry
	300714	adserv
	300715	adreperv
	300716	[неизвестно]
	300717	caad
	300718	caau
	300719	cescda
	300720	vcapiadmin
	300721	vcapi20
	300722	tcfs
	300723	csed
	300724	nohand
	300725	hacb
	300726	nfauth
	300727	imlm
	300728	bestcomm
	300729	lprpasswd
	300730	rprpasswd
	300731	proplistd
	300732	mikomomc
	300733	arepa-cas
	300734	[неизвестно]
	300735	[неизвестно]
	300736	ando_ts
	300737	intermezzo
	300738	ftel-sdh-request
	300739	ftel-sdh-response
	300740	[неизвестно]
	300741	[неизвестно]
	300742	[неизвестно]
	300743	[неизвестно]
	300744	[неизвестно]
	300745	vrc_abb
	300746	vrc_comau
	300747	vrc_fanuc
	300748	vrc_kuka
	300749	vrc_reis
	300750	hp_sv6d
	300751	correntmgr01
	300752	correntike
	300753	[неизвестно]
	300754	[неизвестно]
	300755	intransa_location
	300756	intransa_management
	300757	intransa_federation
	300758	portprot
	300759	ipmiprot
	300760	aceapi
	300761	f6000pss
	300762	vsmapi_program
	300763	ubertuple
	300764	ctconcrpcif
	300765	mfuadmin
	300766	aiols
	300767	dsmrootd
	300768	htdl
	300769	caba
	300770	vrc_cosimir
	300771	cmhelmd
	300772	polynsm
	300773	[неизвестно]
	300774	[неизвестно]
	300775	[неизвестно]
	300776	[неизвестно]
	300777	[неизвестно]
	300778	[неизвестно]
	300779	[неизвестно]
	300780	[неизвестно]
	300781	dsmrecalld
	300782	[неизвестно]
	300783	[неизвестно]
	300784	twrgcontrol
	300785	twrled
	300786	twrcfgdb
BMC software	300787-300886	
не выделены	300887 - 300999	
Sun Microsystems	301000-302000 [2000 номеров]	
не выделены	302001-349999	
American Airlines	350000 - 350999	
Acucobol Inc.	351000 - 351099	
The Bristol Group	351100 - 351249	
Amteva Technologies	351250 - 351349	
	351350	wfmMgmtApp
	351351	wfmMgmtDataSrv

не выделены
Sterling Software ITD

351352	wfmMgmtFut1
351353	wfmMgmtFut1
351354	wfmAPM
351355	wfmIAMgr
351356	wfmECMgr
351357	wfmLookOut
351358	wfmAgentFut1
351359	wfmAgentFut2
351360 - 351406	
351407	csed
351360	sched10d
351361	sched11d
351362	sched12d
351363	sched13d
351364	sched14d
351365	sched15d
351366	sched16d
351367	sched17d
351368	sched18d
351369	sched19d
351370	sched20d
351371	sched21d
351372	sched22d
351373	sched23d
351374	sched24d
351375	sched25d
351376	sched26d
351377	sched27d
351378	sched28d
351379	sched29d
351380	sched30d
351381	sched31d
351382	sched32d
351383	sched33d
351384	sched34d
351385	sched35d
351386	sched36d
351387	sched37d
351388	sched38d
351389	sched39d
351390	consoleserver
351391	scheduleserver
351392	RDELIVER
351393	REVENTPROG
351394	RSENDEVENTPROG
351395	snapp
351396	snepad
351397	sdsodb
351398	sdsmain
351399	sdssrv
351400	sdscInt
351401	sdsreg
351402	fsbatch
351403	fsmonitor
351404	fsdisp
351405	fsession
351406	fslog
351407	svdpappserv
351408	gns
351409	[unkonwn]
351410	[unkonwn]
351411	[unkonwn]
351412	axi
351413	rpcxfr
351414	slm
351415	smbpasswdd
351416	tbdbserv
351417	tbprojserv
351418	genericserver
351419	dynarc_ds
351420	dnscmdr
351421	ipcmdr
351422	faild
351423	failmon
351424	faildebug
351425	[неизвестно]
351426	[неизвестно]
351427	siemens_srs
351428	bsproxy
351429	ifsrpc
351430	CesPvcSm
351431	FrPvcSm
351432	AtmPvcSm
351433	radius
351434	auditor
351435	sft

	351436	voicemail
	351437	kis
	351438	SOFTSERV_NOTIFY
	351439	dynarpc
	351440	hc
	351441	iopas
	351442	iopcs
	351443	iopss
	351444	spcnfs
	351445	spcvss
	351446	matilda_sms
	351447	matilda_brs
	351448	matilda_dbs
	351449	matilda_sps
	351450	matilda_svs
	351451	matilda_sds
	351452	matilda_vvs
	351453	matilda_stats
	351454	xtrade
	351455	mapsvr
	351456	hp_graphicsd
	351457	berkeley_db
		berkeley_db_svc
	351458	io_server
	351459	rpc.niod
	351460	rpc.kill
	351461	hmdisproxy
	351462	smdisproxy
	351463	avatard
	351464	namu
	351465	BMCsess
	351466	FENS_Sport
	351467	EM_CONFIG
	351468	EM_CONFIG_RESP
	351469	lodge_proof
	351470	ARCserveIT-Queue
	351471	ARCserveIT-Device
	351472	ARCserveIT-Discover
	351473	ARCserveIT-Alert
	351474	ARCserveIT-Database
	351475	scand1
	351476	scand2
	351477	scand3
	351478	scand4
	351479	scand5
	351480	dscv
	351481	cb_svc
	351482	[неизвестно]
	351483	iprobe
	351484	omniconf
	351485	isan
BG Partners	351486 - 351500	
	351501	mond
	351502	iqllremote
	351503	iqllalarm
не выделены	351504 - 351599	
Orion Multisystems	351600-351855	
не выделены	351856 - 351899	
NSP lab	351900 - 351999	
не выделены	351999 - 352232	
	352233	asautostart
	352234	asmediad1
	352235	asmediad2
	352236	asmediad3
	352237	asmediad4
	352238	asmediad5
	352239	asmediad6
	352240	asmediad7
	352241	asmediad8
	352242	asmediad9
	352243	asmediad10
	352244	asmediad11
	352245	asmediad12
	352246	asmediad13
	352247	asmediad14
	352248	asmediad15
	352249	asmediad16
	352250	waruser
	352251	warlogd
	352252	warsvrmgr
	352253	warvfsysd
	352254	warftpd
	352255	warnfsd
	352256	bofproxyc0
	352257	bofproxys0
	352258	bofproxyc1

	352259	bofproxys1
	352260	bofproxyc2
	352261	bofproxys2
	352262	bofproxyc3
	352263	bofproxys3
	352264	bofproxyc4
	352265	bofproxys4
	352266	bofproxyc5
	352267	bofproxys5
	352268	bofproxyc6
	352269	bofproxys6
	352270	bofproxyc7
	352271	bofproxys7
	352272	bofproxyc8
	352273	bofproxys8
	352274	bofproxyc9
	352275	bofproxys9
	352276	bofproxysa
	352277	bofproxysa
	352278	bofproxycb
	352279	bofproxysb
	352280	bofproxysc
	352281	bofproxysc
	352282	bofproxycd
	352283	bofproxysd
	352284	bofproxysc
	352285	bofproxysc
	352286	bofproxysc
	352287	bofproxysf
	352288	bofproxyp0
	352289	bofproxyp1
	352290	bofproxyp2
	352291	bofproxyp3
	352292	bofproxyp4
не выделены	352293-370000	
	370001	[неизвестно]
	370002	[неизвестно]
	370003	[неизвестно]
	370004	[неизвестно]
	370005	[неизвестно]
	370006	[неизвестно]
	370007	[неизвестно]
	370008	[неизвестно]
	370009	[неизвестно]
	370010	[неизвестно]
	370011	[неизвестно]
	370012	[неизвестно]
	370013	[неизвестно]
	370014	[неизвестно]
	370015	[неизвестно]
	370016	[неизвестно]
	370017	[неизвестно]
	370018	[неизвестно]
	370019	[неизвестно]
	370020	[неизвестно]
	370021	[неизвестно]
	370022	[неизвестно]
	370023	[неизвестно]
	370024	[неизвестно]
	370025	[неизвестно]
	370026	[неизвестно]
	370027	[неизвестно]
не выделены	370028 - 379999	
	380000	opensna
	380001	probenet
	380002	[неизвестно]
	380003	license
	380004	na.3com-remote
	380005	na.ntp
	380006	probeutil
	380007	na.vlb
	380008	cds_mhs_agent
	380009	cds_x500_agent
	380010	cds_mailhub_agent
	380011	codex_6500_proxy
	380012	codex_6500_trapd
	380013	na.nm212
	380014	cds_mta_metrics_agent
	380015	[unknown]
	380016	na.cable
	380017	codexcapletrap
Swiss Re	380018-380028	
	380029	ncstat
	380030	ncnfsstat
	380031	ftams
	380032	na.isotp

	380033	na.rfc1006
не выделены	380034 - 389999	
Epoch Systems	390000 - 390049	
Quickturn Systems	390050 - 390065	
Team One Systems	390066 - 390075	
General Electric CRD	390076 - 390085	
TSIG NFS subcommittee	390086 - 390089	
SoftLab ab	390090 - 390099	
Legato Network Services	390100 - 390115	
	390116	cdsmonitor
	390117	cdslock
	390118	cdslicense
	390119	shm
	390120	rws
	390121	cdc
Data General	390122 - 390141	
Perfect Byte	390142 - 390171	
JTS Computer Systems	390172 - 390181	
Parametric Technology	390182 - 390191	
Voxem	390192 - 390199	
Effix Systems	390200 - 390299	
Motorola	390300 - 390309	
Mobile Data Intl.	390310 - 390325	
Physikalisches Institut	390326 - 390330	
Ergon Informatik AG	390331 - 390340	
Analog Devices Inc.	390341 - 390348	
Interphase Corporation	390349 - 390358	
NeWare	390359 - 390374	
Qualix Group	390375 - 390379	
Xerox Imaging Systems	390380 - 390389	
Noble Net	390390 - 390399	
Legato Network Services	390400 - 390499	
Client Server Tech.	390500 - 390511	
Atria	390512 - 390517	
GE NMR Instruments	390518 - 390525	
Harris Corp.	390526 - 390530	
Unisys	390531 - 390562	
Aggregate Computing	390563 - 390572	
Interactive Data	390573 - 390580	
OKG AB	390581 - 390589	
K2 Software	390591 - 390594	
Collier Jackson	390595 - 390599	
Remedy Corporation	390600 - 390699	
Mentor Graphics	390700 - 390799	
AT&T Bell Labs (Lucent)	390800 - 390899	
Xerox	390900 - 390999	
Silicon Graphics	391000 - 391063	
Data General	391064 - 391095	
Computer Support Corp.	391096 - 391099	
Quorum Software Systems	391100 - 391199	
InterLinear Technology	391200 - 391209	
Highland Software	391210 - 391229	
Boeing Comp. Svcs.	391230 - 391249	
IBM Sweden	391250 - 391259	
Signature Authority Svc	391260 - 391271	
ZUMTOBEL Licht GmbH	391272 - 391283	
NOAA/ERL	391284 - 391299	
NCR Corp.	391300 - 391399	
FTP Software	391400 - 391409	
Cadre Technologies	391410 - 391433	
Visionware Ltd (UK)	391434 - 391439	
IBR-Partner AG	391440 - 391449	
CAP Programator AB	391450 - 391459	
Reichle+De-Massari AG	391460 - 391474	
Swiss Bank Corp (London)	391475 - 391484	
Unisys Enterprise Svr	391485 - 391489	
Intel - Test Dev. Tech.	391490 - 391499	
Ampex	391500 - 391755	
	391756	naas-spare
	391757	naas-admin
	391758	isps
	391759	isps-admin
	391760	mars
	391761	mars-admin
	391762	attcis_spare0
	391763	attcis_spare1
	391764	mail-server
	391765	mail-server-spare
	391766	attcis_spare2
	391767	attcis_spare3
	391768	attcis_spare4
	391769	attcis_spare5
	391770	attcis_spare6
	391771	attcis_spare7
Integrated Systems, Inc.	391772 - 391779	
Parametric Tech., Inc.	391780 - 391789	

Ericsson Telecom AB	391790 - 391799
SLAC	391800 - 391849
	391850 qhrdata
	391851 qhrbackup
	391852 minutedata
	391853 prefecture
	391854 supc
	391855 suadmincrw
	391856 suadminotas
	391857 sumessage
	391858 sublock
	391859 sumotd
staffware dev. (uk)	391860 - 391869
Staffware Dev. (UK)	391870 - 391879
	391880 namesrvr
	391881 disksrvr
	391882 tapesrvr
	391883 migsrvr
	391884 pdmsrvr
	391885 pvrsvr
	391886 repacksrvr
	391887 [неизвестно]
Convex Computer Corp.	391888 - 391951
	391952 lookoutsrv
	391953 lookoutagnt
	391954 lookoutprxy
	391955 lookoutsnmp
	391956 lookoutrmon
	391957 lookoutfut1
	391958 lookoutfut2
windward	391959 - 391967
	391968 sra_legato
	391969 sra_legato_imgsvr
	391970 sra_legato_0
	391971 sra_legato_1
	391972 sra_legato_2
	391973 sra_legato_3
	391974 sra_legato_4
	391975 sra_legato_5
	391976 sra_legato_6
	391977 sra_legato_7
	391978 sra_legato_8
	391979 sra_legato_9
Brooktree Corp.	391980 - 391989
Cadence Design Systems	391990 - 391999
J. Frank & Associates	392000 - 392999
Cooperative Solutions	393000 - 393999
Xerox Corp.	394000 - 395023
	395024 odbc_sqlretriever
3M	395025 - 395091
Digital Zone Intl.	395092 - 395099
Software Professionals	395100 - 395159
Del Mar Solutions	395160 - 395164
	395165 ife-es
	395166 ife-resmgr
	395167 ife-aes
	395168 ife-bite
	395169 ife-loader
	395170 ife-satcom
	395171 ife-seat
	395172 ife-dbmgr
	395173 ife-testmgr
	395174 atrium_server
	395175 ase_director
	395176 ase_agent
	395177 ase_hsm
	395178 ase_mgr
	395179 ase_sim
Hewlett-Packard	395180 - 395194
XES, Inc.	395195 - 395199
Unitech Products	395200 - 395249
TransSys	395250 - 395505
Unisys Govt Systems	395506 - 395519
Bellcore	395520 - 395529
IBM	395530 - 395561
AT&T Network Services	395562 - 395571
Data General	395572 - 395577
Swiss Bank Corp	395578 - 395597
Swiss Bank Corp	395598 - 395637
Novell	395638 - 395643
Computer Associates	395644 - 395650
Omneon Video Networks	395651 - 395656
unassigned	395657 - 395908
UK Post Office	395909 - 395924
AEROSPATIALE	395925 - 395944
Result d.o.o.	395945 - 395964

DataTools, Inc.	395965	-	395980
CADIS, Inc.	395981	-	395990
Cummings Group, Inc.	395991	-	395994
Cadre Technologies	395995	-	395999
American Airlines	396000	-	396999
Ericsson Telecom TM Div	397000	-	398023
IBM	398024	-	398028
Toshiba OME Works	398029	-	398033
TUSC Computer Systems	398034	-	398289
AT&T	398290	-	398320
Ontario Hydro	398321	-	398346
Micrion Corporation	398347	-	398364
unassigned	398365	-	398591
Pegasystems, Inc.	398592	-	399616
Spectra Securities Soft	399617	-	399850
QualCom	399851	-	399866
unassigned	399867	-	399884
Altris Software Ltd.	399885	-	399899
ISO/IEC WG11	399900	-	399919
Parametric Technology	399920	-	399949
Dolby Laboratories	399950	-	399981
unassigned	399982	-	399991
Xerox PARC	399992	-	399999
#			
Next Inc.	200100000	-	200199999
Netwise (RPCtool)	200200000		
Concurrent Computer Corp	200200001	-	200200007
AIM Technology	200300000	-	200399999
TGV	200400000	-	200499999
#			
# Выделенные Sun номера для аутентификации			
#			
AUTH_NONE	0		/* без аутентификации, см. RFC 1831 */
			/* AUTH_NULL */
AUTH_SYS	1		/* unix style (uid+gids), RFC 1831 */
			/* AUTH_UNIX */
AUTH_SHORT	2		/* короткий стиль unix, см. RFC 1831 */
AUTH_DH	3		/* стиль des (шифрованная метка времени) */
			/* AUTH_DES, см. RFC 2695 */
AUTH_KERB	4		/* аутентификация kerberos, см. RFC 2695 */
AUTH_RSA	5		/* аутентификация RSA */
RPCSEC_GSS	6		/* защита RPC на основе GSS для аутентификации, целостности и приватности, RFC 5403 */
AUTH_NW	30001	NETWARE	
AUTH_SEC	200000	TSIG NFS subcommittee	
AUTH_ESV	200004	SVr4 ES	
AUTH_NQNF	300000	Univ. of Guelph - Not Quite NFS	
AUTH_GSSAPI	300001	OpenVision <john.linn@ov.com>	
AUTH_ILU_UGEN	300002	Xerox <janssen@parc.xerox.com>	
		- ILU Unsecured Generic Identity	
#			
# Небольшие блоки, выделенные из 39xxxx			
#			
AUTH_SPNEGO	390000		
	390000	-	390255 псевдо-варианты NFS для RPCSEC_GSS
	390003	-	аутентификация kerberos_v5, RFC 2623
	390004	-	kerberos_v5 с защитой целостности, RFC 2623
	390005	-	kerberos_v5 с защитой приватности, RFC 2623
	200000000	Reserved	
	200100000	NeXT Inc.	

Нормативные документы

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, [RFC 2119](#), March 1997.
- [RFC2203] Eisler, M., Chiu, A., and L. Ling, "RPCSEC_GSS Protocol Specification", RFC 2203, September 1997.
- [RFC4506] Eisler, M., Ed., "XDR: External Data Representation Standard", STD 67, [RFC 4506](#), May 2006.

Дополнительная литература

- [DH] Diffie & Hellman, "New Directions in Cryptography", IEEE Transactions on Information Theory IT-22, November 1976.
- [RFC0768] Postel, J., "User Datagram Protocol", STD 6, [RFC 768](#), August 1980.
- [RFC0793] Postel, J., "Transmission Control Protocol", STD 7, [RFC 793](#), September 1981.
- [RFC1094] Sun Microsystems, "NFS: Network File System Protocol specification", RFC 1094, March 1989.
- [RFC1813] Callaghan, B., Pawlowski, B., and P. Staubach, "NFS Version 3 Protocol Specification", RFC 1813, June 1995.
- [RFC1831] Srinivasan, R., "RPC: Remote Procedure Call Protocol Specification Version 2", [RFC 1831](#), August 1995.
- [RFC1833] Srinivasan, R., "Binding Protocols for ONC RPC Version 2", RFC 1833, August 1995.

- [RFC2623] Eisler, M., "NFS Version 2 and Version 3 Security Issues and the NFS Protocol's Use of RPCSEC_GSS and Kerberos V5", RFC 2623, June 1999.
- [RFC2695] Chiu, A., "Authentication Mechanisms for ONC RPC", RFC 2695, September 1999.
- [RFC2743] Linn, J., "Generic Security Service Application Program Interface Version 2, Update 1", RFC 2743, January 2000.
- [RFC3530] Shepler, S., Callaghan, B., Robinson, D., Thurlow, R., Beame, C., Eisler, M., and D. Noveck, "Network File System (NFS) version 4 Protocol", RFC 3530, April 2003.
- [RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, [RFC 5226](#), May 2008.
- [VMTP] Cheriton, D., "VMTP: Versatile Message Transaction Protocol", Preliminary Version 0.3, Stanford University, January 1987.
- [XRPC] Birrell, A. D. & B. J. Nelson, "Implementing Remote Procedure Calls", XEROX CSL-83-7, October 1983.

Адрес автора

Robert Thurlow
Sun Microsystems, Inc.
500 Eldorado Boulevard, UBRM05-171
Broomfield, CO 80021
Phone: 877-718-3419
EMail: robert.thurlow@sun.com

Перевод на русский язык

Николай Малых
nmalykh@protokols.ru