Введение в недокументированное применение протокола Spanning Tree

О.К. Артемьев В.В. Мяснянкин

8 июня 2002 г.

Лицензионное соглашение

Данное исследование является интеллектуальной собственностью Олега Артемьева и Владислава Мяснянкина (здесь и далее - авторов). Оно может свободно использоваться для ссылок, но его текст - как полностью, так и частично - не может быть переведен на какой либо язык или включен в любую другую статью, книгу, журнал, а также иные электронные или бумажные издания без письменного разрешения обоих авторов. Кроме того, при ссылке на результаты этого исследования, или их использовании необходимо полностью приводить название, имена авторов и текст лицензии. Материалы исследования могут свободно распространяться в электронном виде, если и только если соблюдены следующие условия:

1. лицензионное соглашение и текст самого исследования оставлены без каких либо изменений, включая PGP подпись; причем какое либо переформатирование текста не допускается;

2. распространение материалов этого исследования не противоречит данной лицензии.

Материалы исследования содержат информацию, свободное распространение которой не противоречит законодательству большинства стран. Если же ваше локальное законодательство содержит ограничения на распространение информации об уязвимостях информационных систем (подобие DMCA в США), то публикация данных материалов в этой стране противоречит данной лицензии. Более того, будучи гражданином такой страны и читая эту статью, вы нарушаете и законодательство своей страны, и данную лицензию.

Авторы не несут ответственности за любые последствия от использования (или не использования) изложенной в исследовании информации, включая повреждение данных, недополучение прибыли, а также любые другие косвенные или прямые потери.

Авторы декларируют материалы этого исследования как исключительно образовательные. Вы не должны знакомиться с ними, если не согласны использовать их иначе как в образовательных целях.

Данное лицензионное соглашение может быть изменено без предупреждения по согласию обоих авторов.

Гарантии: это исследование распространяется без каких либо гарантий и ответственности.

...
They say that is technological breakdown...
...or no - this is the road to hell.
...
Chris Rea.

...
Стой, опасная зона - работа мозга.
...
В. Цой.

Замечания

Этот документ основан на анализе стандартов IEEE, относящихся к протоколу Spanning Tree [1], а так же на анализе стандарта RFC 2878 [5], и будет часто цитировать и ссылаться на них. Все копирайты и зарегистрированные торговые марки, упомянутые в нем, принадлежат их собственникам. Протокол и алгоритм Spanning Tree определены в главе 8 стандарта «MEDIA ACCESS CONTROL (MAC) BRIDGES ANSI/IEEE Std 802.1D» [1].

На момент публикации первичных данных часть разделов требовала доводки - внесения изменений, дополнительных подробностей. Фрагменты, которые будут доработаны в следующей версии, помечены в тексте знаком $\otimes$.

Этот документ предназначен для тех, кто, как минимум, знаком с моделью OSI [2]. Для тех, кто не имел возможности прочесть содержимое стандарта [1], написана краткое введение в работу протокола STP, которое, однако, не может заменить чтения этого стандарта, поскольку рассматривает подробно только те особенности его работы, которые необходимы для описания его недокументированного применения. Также предполагается, что читатель имеет хорошее представление о построении ЛВС и используемых в этой области сокращениях. В целях наиболее точной передачи смысла многие цитаты из оригинальной документации и текстов стандартов оставлены на английском языке. В связи с этим предполагается, что читатель владеет английским языком в степени, достаточной для понимания технического текста. Далее в тексте при описании атак, до тех пор, пока не сказано иного, рассматриваются атаки

на устройства с использованием Spanning Tree Protocol[1].

Идея и первичный текст принадлежат Олегу Артемьеву[2], первичная концептуальная реализация в коде принадлежит Владиславу Мяснянкину[3]. Для связи с владельцем любого из указанных в статье email'ов надо убрать из него слово NOSPAM - для предотвращения авто-спама. Хотелось бы сказать несколько слов о причинах написания этой статьи авторами:

Олег Артемьев: Хочется сказать себе и всему миру - я не скрипткидди. Ж;) Надеюсь, теперь я могу сказать это с полным правом. :) Хочется добавить строчку в резюме - разбор стандарта по косточкам с выявлением его недостатков - это, как ни странно, под силу не каждому. Это интересно - замечательная зарядка для интеллекта. Это полезно для всего мира и моей страны в частности - в результате этой публикации, я надеюсь, будет дан толчок в развитии стандартов ЛВС и обслуживающих ЛВС устройств в сторону обеспечения реальной, или, хотя бы, просто большей безопасности, чем та, которая существует в данный момент. Даже если люди, ответственные за разработку новых версий сетевых протоколов, проигнорируют эту статью им придется принять во внимание ее содержимое после того, как оно станет доступным всему миру. А еще, я надеюсь, что люди, ответственные в нашей стране за выбор используемых технологий в областях, относящихся к государственной тайне, будут предупреждены таким образом о потенциальной опасности технологий современных ЛВС разработанных вне России.

Владислав Мяснянкин: Мне сложно однозначно сформулировать причину. Это и естественный интерес к одному из аспектов сетевой безопасности, это и проявление чувства долга - ведь в своей повседневной деятельности я пользуюсь результатами чужих исследований (в частности, информацией об обнаруженных уязвимостях и соответствующими заплатками).

[1] На данный момент рассматривается только IEEE реализация STP (см. раздел 4, стр. 25).

[2] olli@olli.digger.orgNOSPAM.ru

[3] vlad@cybervlad.port5.NOSPAMcom

Ключевые слова

Уязвимости протокола Spanning Tree, уязвимости алгоритма Spanning Tree; безопасность 2-го уровня OSI; исследование безопасности стандарта IEEE 802.1D; безопасность мостов, маршрутизаторов и коммутаторов на 2-м уровне OSI; cisco, avaya, 3com, compex, cnet (и т.д. - все, кто делает stp-совместимые устройства); уязвимости интеллектуальных коммутаторов, поддерживающих протокол Spanning Tree; реализация отказа в обслуживании и навязывание ложного маршрута с использованием протокола Spanning Tree; STP design flaw; exploiting STP holes.

Содержание

1 Введение в Spanning Tree протокол 7

2 STP & VLANs 16
2.1 Краткое введение в технологию VLAN 16
2.2 Миф о раздельном дереве STP в каждом VLAN 18
2.3 Миф о физической раздельности VLAN'ов 18

3 STP в гетерогенных средах 19

4 Замечания, вытекающие из анализа RFC 2878 20

5 Комментарии к написанию кода, генерирующего STP пакеты 27

6 Возможные схемы атак 29
6.1 BPDU spooffing 30
6.2 Provocation Aging 33
6.3 BPDU filter 33
6.4 Отказ в обслуживании (DoS) 34
6.4.1 «Вечные выборы» 34
6.4.2 «Исчезновение корня» 35
6.4.3 Алгоритм случайного совпадения 36
6.4.4 Другие возможные алгоритмы 36
6.4.5 Частичный отказ в обслуживании 37
6.4.6 Параметры, приводящие к DoS 41
6.4.7 Какие паузы эффективны для DoS 42
6.5 Человек посередине (MiTM) 43
6.5.1 Пакеты для MiTM 45
6.6 Провокационный сниффинг 46
6.6.1 Что такое провокационный сниффинг 46
6.6.2 Организация провокационного сниффинга 47
6.6.3 Сравнение arp-poisoning и provocation sniffing 50
6.6.4 Комментарии к написанию кода для provocation sniffing 50

7 Получение дополнительной информации о сети при помощи STP 51

8 Особенности реализации у различных производителей 51
8.1 Cisco 51
8.2 Avaya (бывш. Lucent) 52
8.3 Intel 53
8.4 HP 53

8.5 3Com 53
8.5.1 Установка параметров STP для порта 55

9 Некоторые замечания по поводу Linux bridging project 57

10 Некоторые замечания по поводу GARP и GVRP 57

11 Обзор атак на 2 уровне OSI 57

12 Уязвимые продукты 58

13 Примеры уязвимых сетей 59

14 Как администраторы сетей могут противостоять атакам 61

15 Как IDS могут обнаружить STP атаки 63
15.1 Сложности обнаружения STP-атак и их причины 63
15.2 Варианты обнаружения атак 63
15.2.1 Вариант обнаружения по наличию STP пакетов . . 63
15.2.2 Вариант обнаружения по «чужому» Bridge ID . . . 64
15.2.3 Вариант обнаружения по длительности 64
15.2.4 Вариант обнаружения по интенсивности 65
15.2.5 Вариант обнаружения по монотонности 65
15.2.6 Вариант обнаружения по цикличности 65
15.2.7 Вариант обнаружения по потере производительности 66
15.2.8 Вариант обнаружения по изменению интервалов между ST событиями 66
15.2.9 Невозможность обнаружить атаку. Пример 67

16 Корни проблемы, или «откуда ноги растут» 67

17 Что делать производителям 68

18 Эпилог 71
18.1 Кратко: что можно сделать с помощью нецелевого применения STP? 71
18.2 Кратко: чего нельзя сделать с помощью нецелевого применения STP? 71

19 Благодарности 72
19.1 Олег Артемьев 72
19.2 Владислав Мяснянкин 73

20 Ссылки **74**
20.1 Ссылки, относящиеся к Spanning Tree 74
20.2 Ссылки, относящиеся к теме «bridging» 74
20.3 Ссылки по конструированию фреймов 74
20.4 Open Systems Interconnection refernce model 74
20.5 Другие интересные ссылки 75

A Программа формирования ST пакетов **78**

B Сценарий для запуска программы формирования ST пакетов **85**

Список иллюстраций

1 Принцип работы STP 8
2 Структура C-BPDU на языке C 13
3 Структура TCN-BPDU на языке C 13
4 STP в действии 15
5 Пример логической структуры сети 20
6 Пример физической структуры сети 21
7 Частичный DoS (пример 1) 38
8 Частичный DoS (пример 2) 39
9 Частичный DoS (пример 3) 40
10 Человек посередине 43

Список таблиц

1 Структура C-BPDU 12
2 Структура TCN-BPDU 12

1 Введение в Spanning Tree протокол

Этой статьей авторы хотели бы не столько объяснить, что такое Spanning Tree Protocol, сколько рассказать о недокументированных возможностях его применения, которые, возможно, не принимались во внимание разработчиками протокола. Поэтому, несмотря на то, что в статье в общих чертах приводится описание протокола Spanning Tree и его назначения, она не претендует на статус материала, который следует использовать для изучения этого протокола с нуля - предполагается, что вы уже читали какие-либо материалы на эту тему, а это введение лишь освежит вашу память. Если же вы вообще не знакомы с этим протоколом и (или) это введение показалось вам недостаточно вразумительным - рекомендуем обратиться к многочисленным описаниям STP в литературе, на сайтах ведущих компаний производителей интеллектуальных LAN-устройств (например, поискать материалы на http://www.cisco.com) и, разумеется, к IEEE стандарту, описывающему этот протокол [1]. Также в разделе 20 («ссылки») есть, возможно, достаточный для составления полной картины набор url.

Основное назначение протокола Spanning Tree - построение топологии ЛВС без избыточного дублирования соединений или закольцовывания, недопустимых в силу логики построения ЛВС. STP позволяет организовать в сети, построенной при помощи мостов (bridges), отказоустойчивую архитектуру[4].

Используя STP, вы можете построить сеть, в которой существует несколько параллельных путей, и гарантировать при этом, что:

- резервные пути прохождения трафика при нормальном функционировании основного пути заблокированы;
- один из резервных путей активизируется при нарушении основного пути.

Например, на рисунке 1 показана сеть, построенная на четырех коммутаторах, поддерживающих протокол STP. Это обстоятельство позволяет соединить их избыточным количеством линков, которые в обычных условиях создали бы петли и привели к неработоспособности сети. Однако, при инициализации сети STP обнаруживает дублирующие пути

[4]На данный момент поддержка Spaning Tree Protocol встречается не только в переключательном оборудовании (коммутаторы, они же свитчи или, в частном случае, мосты), но и в маршрутизирующем оборудовании, которое снабжено возможностью коммутировать пакеты между интерфейсами. Так, если рассматривать, например, Cisco, то в результате поддержки маршрутизатором Bridging Virtual Interface (например, модель 3640 и др.) становится логичной поддержка STP.

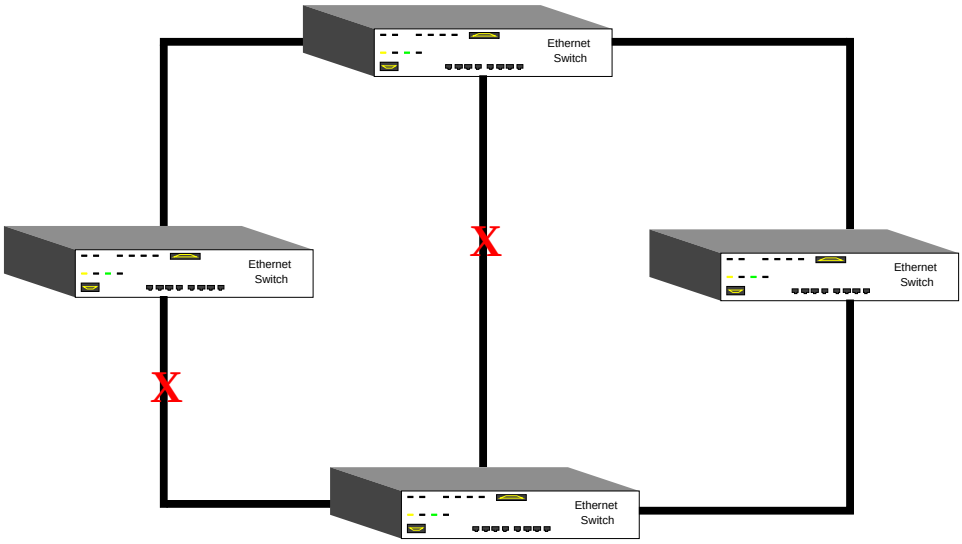

Рис. 1: Принцип работы STP

и оставляет на каждое направление только один, переводя остальные задействованные в линках порты в состояние «blocking» (на рисунке соответствующие линки зачеркнуты). При нарушении основного канала передачи (например, в следствие отключения одного из коммутаторов), STP обнаруживает этот факт и задействует запасной маршрут.

Протокол разработан в начале 90-х годов XX века (ANSI/IEEE 802.1D 1993 Edition) и дорабатывался в 1996 и 1998 годах. В качестве модели используется граф в виде дерева. Построение дерева сети начинается с корня (root) – устройства, выигравшего выборы корневого (designated root). Designated root – это просто отправная точка для построения активной незакольцованной топологии, в которой физические кольца не дают логических колец (см. главы 8 и 9 [1]). Иными словами, это логическое понятие.

Важно заметить, что Spanning Tree Protocol определен для различных сред передачи данных (MAC, Media Access Control). Далее в тексте рассматриваются примеры сетей в основном на базе Ethernet, как наиболее распространенной.

BPDU - это именно пакеты, а не фреймы. Это ясно из [1], 8.3.2, стр.61. BPDU пакеты инкапсулируются в используемый данной топологией тип фрейма с мультикастным MAC адресом в поле назначения, что обуславливает передачу этих пакетов через неинтеллектуальное оборудование,

не знающее о существовании Spanning Tree.

Работа STP подразумевает выполнение следующих условий:

1. Возможность передачи информации между мостами, что осуществляется путем передачи каждым STP-совместимым устройством специальных блоков данных - Bridge Protocol Data Units (BPDU). Эти блоки данных передаются в пакетах с определенным групповым адресом назначения (multicast address), зарезервированным в [7].

2. Один из мостов функционирует как «ведущее» устройство, называемое Designated Root Bridge.

Designated root присутствует всегда, даже когда топология не содержит физических колец. Если в сети только одно STP-совместимое устройство - до тех пор, пока STP не выключен, оно будет анонсировать себя. Каждое STP-совместимое устройство[5] начинает работу, считая себя designated root, так что в сети с единственным STP-совместимым устройством оно и является Designated Root Bridge.

Если же в сети присутствует более устройства, поддерживающего STP, то Designated Root Bridge выбирается путем голосования (выборов) на основе значения параметра Bridge Identifier, обычно являющегося комбинацией уникального MAC-адреса моста и устанавливаемого для моста приоритета (см. [1], главы 8 и 9). На роль Designated Root Bridge назначается мост с наименьшим значением Bridge Identifier.

Для всех остальных мостов в сети определяется Root Port, т.е. порт моста, ближайший к Root Bridge. От других портов, соединенных с root bridge непосредственно или через другие мосты, он отличается своим идентификатором, который содержит его номер и «вес», который может быть задан администратором.

Другой величиной, влияющей на процесс выборов, является Root Path Cost. Она состоит из стоимости пути до Root Port данного моста плюс стоимость путей до Root Port мостов по всему маршруту до Root Bridge.

[5] В данном случае авторы позволяют себе некоторую вольность, поскольку [1] рассматривает исключительно мосты (или свитчи, коммутаторы, поскольку коммутатор – это многопортовый мост). Начиная от этого момента и далее, речь идет об абстрактном «устройстве», которое, в частном случае, может быть как коммутатором, так и маршрутизатором, а равно и другим интеллектуальным устройством.

Помимо Designated Root Bridge в STP вводится логическое понятие Designated Bridge. Владелец этого статуса считается главным в обслуживании данного сегмента ЛВС. Статус Designated Bridge также является выборным и может переходить от одного устройства к другому.

Аналогичным образом вводится выборное логическое понятие Designated Port - порта, который обслуживает данный сегмент сети. Для Designated Port вводится понятие Designated cost, описывающее стоимость пути ([1], 8.5.5.5).

В зависимости от содержимого получаемых от соседних устройств конфигурационных пакетов то или иное устройство перестает оспаривать статус Designated Root Bridge и начинает анонсировать устройство, выигравшее этот статус в процессе выборов. Так происходит до тех пор, пока ситуация не придет к завершающей стадии, которая характеризуется следующим образом:

1. В сети только одно устройство, считающее себя корнем, а остальные устройства периодически анонсируют его как корень, что поддерживает статус кво, обновляя таймеры на всех STP-совместимых устройствах.

2. Root Bridge периодически посылает во все свои порты пакеты с BPDU. Интервал времени, через который происходит посылка, называется Hello Time.

3. В каждом сегменте сети имеется единственный Designated Bridge Port - порт, через который проходит обмен трафиком с Root Bridge. Этот порт имеет наименьшее значение Root Path Cost по сравнению с другими портами в сегменте, либо меньший bridge ID.

4. BPDU принимаются и отправляются STP-совместимым устройством на всех его портах, даже на тех, которые были «выключены» работой STP. Однако BPDU не принимаются на портах, которые были «выключены» администратором.

5. Каждый мост осуществляет пересылку (forwarding) пакетов только между Root Port и портами, которые являются Designated Bridge Port для соответствующего сегмента. Все остальные порты находятся в состоянии «Blocking».

Как уже говорилось, корень не несет на себе иных обязанностей, кроме анонса своих параметров согласно спецификации Spanning Tree протокола, а это, в частности, значит, что в сети, не содержащей физических

колец или избыточных соединений, от изменения владельца статуса Designated Root Bridge пути пакетов, пересылаемых между двумя произвольными станциями не изменится.

Спецификация протокола [1] не накладывает ограничений на время, в которое могут начаться выборы того или иного параметра, а лишь описывает необходимые для этого условия.

К числу ситуаций, в которых возникают выборы как минимум одного из параметров, относятся:

- подключение в сеть нового STP-совместимого устройства;
- устаревание имеющихся данных (например, в результате выхода из строя или отключения одного из устройств, участвовавших в создании дерева);
- административная акция, изменяющая топологию сети.

В качестве параметров при выборах используются следующие величины:

1. Постоянные величины:

   root path cost - стоимость пути к корневому устройству. Чем меньше значение, тем выше приоритет ([1], 8.5.1.2);

   bridge identifier - идентификатор устройства. Чем меньше значение, тем больше приоритет ([1], 8.5.1.3);

   port identifier - идентификатор порта. Чем меньше значение, тем выше приоритет ([1], 8.5.1.4).

2. Выборные величины:

   designated port - назначенный порт ([1], 8.5.5.7);

   designated root - назначенный корень ([1], 8.5.5.4);

   designated cost - назначенная стоимость ([1], 8.5.5.5).

Величины bridge identifier и port identifier являются составными и образуются из поля приоритета (может устанавливаться администратором) и поля, присваиваемого производителем (некоторые устройства могут поддерживать установку этих идентификаторов целиком).

Согласно спецификации выборные величины, в зависимости от контекста, могут принимать участие в различных выборах, например, designated cost может принимать участие как в выборах назначенного моста, так и в выборах корневого моста.

Смещение	Название	Размер
1	Protocol Identifier	2 bytes
	Protocol Version Identifier	1 byte
	BPDU type	1 byte
	Flags	1 byte
	Root Identifier	8 bytes
	Root Path Cost	4 bytes
	Bridge Identifier	8 bytes
	Port Identifier	2 bytes
	Message Age	2 bytes
	Max Age	2 bytes
	Hello Time	2 bytes
35	Forward Delay	2 bytes

Таблица 1: Структура C-BPDU

Название	Размер
Protocol Identifier	2 bytes
Protocol Version Identifier	1 byte
BPDU type	1 byte

Таблица 2: Структура TCN-BPDU

Во время работы устройства анонсируют себя и параметры своих портов через Configuration BPDU (далее называемые c-bpdu). Формат c-bpdu приведен в таблице 1 (согласно [1], 9.3.1, рисунок 9-1) и на рисунке 2 (согласно [1], 8.9.1).

Для сообщения об изменениях в топологии используются topology change notification BPDUs, формат которых приведен в таблице 2 и на рисунке 3 (согласно [1], параграф 8.3.5, стр. 63, последний абзац).

Собственно информации передается совсем немного - только статус. По факту прихода такого BPDU любой коммутатор уменьшает время жизни записей в своей таблице коммутации до минимума, определенного стандартом [1].

В момент получения конфигурационного bpdu STP-совместимый мост может определить, что пришедшая информация должна обновить имеющуюся у него информацию. В этом случае устройство изменяет содержимое анонсируемых им bpdu так, чтобы анонсировать выигравшее

```
typedef struct {
  Bpdu_type    type;
  Identifier   root_id;
  Cost         root_path_cost;
  Identifier   bridge_id;
  Port_id      port_id;
  Time         message_age;
  Time         max_age;
  Time         hello_time;
  Time         forward_delay;
  Flag         topology_change_acknowledgement;
  Flag         topology_change;
} Config_bpdu;
```

Рис. 2: Структура C-BPDU на языке C

```
typedef struct {
  Bpdu_type    type;
} Tcn_bpdu;
```

Рис. 3: Структура TCN-BPDU на языке C

устройство, а не себя. Обратите внимание, что BPDU не проходят сквозь устройства, которые поддерживают STP - они лишь, при определенных условиях, инициируют на «промежуточном» устройстве посылку собственных BPDU ([1], 8.3.2) и/или изменение содержимого BPDU посылаемых «промежуточным» устройством.

В момент обновления конфигурации STP-совместимое устройство изменяет состояние своих портов - они поочередно принимают одно из следующих возможных значений ([1], 8.4):

блокирован (Blocking) Заблокирован. Однако, фреймы содержащие STP пакеты (bpdu), принимаются и обрабатываются, в отличие от пользовательских фреймов ([1], 8.4.1);

слушает (Listening) Первый этап подготовки к состоянию Forwarding. Фреймы, содержащие STP пакеты (bpdu), принимаются и обрабатываются, в отличие от пользовательских фреймов ([1], 8.4.2);

обучается (Learning) Второй этап подготовки к состоянию Forwarding. Фреймы, содержащие STP пакеты (bpdu), принимаются и обрабатываются, в отличие от пользовательских фреймов ([1], 8.4.3);

передает (forwarding) Рабочее состояние портов устройства. Передаются как фреймы, содержащие STP пакеты, так и фреймы пользо-

вательских протоколов ([1], 8.4.4).

Время, на которое порт попадает в то или иное состояние, определяется значениями пауз, которые передаются посредством C-BPDU вместе с остальными параметрами (см. раздел 6.4.7 на стр. 42).

Согласно стандарту, во время нахождения в состояниях Blocking, Listening и Learning пересылка не-STP пакетов не разрешена ([1], 8.4.1-8.4.3). В то же время, bpdu-пакеты не обрабатываются только в случае если порт выключен администратором ([1], 8.4.5). *Это говорит о том, что если порты находятся в режимах Blocking, Listening или Learning - сеть работает только на STP протокол, но не на пользователя.* Этим можно воспользоваться для организации атаки «отказ в обслуживании» (Denial of Service, DoS), в терминах [3].

Собственно, причина, по которой протокол работает именно таким образом ([1], 8.3.4), проста - все так организовано для того, чтобы избежать хаотичного дублирования и/или размножения пакетов в различные сегменты сети в момент перестройки ее топологии. Дублирование и/или размножение пакетов возможно, например, за счет возникновения временных колец, возникающих в процессе переключения состояния того или иного линка. Описание DoS атаки, а также величины параметров, необходимые для максимальной ее эффективности, см. в разделе 6 на стр. 29.

В силу того, что топология сети определяется при участии протокола STP возможна атака по схеме ложный объект PBC с навязыванием ложного пути (в терминологии [3]). Подробности см. в разделе 6.5, стр. 43.

Следует отметить, что стандарт не уточняет термин LAN-сегмент в данном контексте, что еще раз указывает, что это вещь весьма относительная, так что состав соответствующих сетей будет меняться в зависимости от выбранной точки подключения.

Вообще говоря, описываемые на примере Ethernet уязвимости протокола относятся (часть может быть применена практически без изменений алгоритма) к любым топологиям, на которых может работать STP. Как следствие, протокол и рассматриваемые атаки, требующие включенной поддержки STP, могут быть применены на любых сетевых топологиях, не допускающих логических колец). В [1] специально обращено внимание на то, что STP может ходить не только поверх Ethernet, но и поверх других сетей с другим принципом MAC.

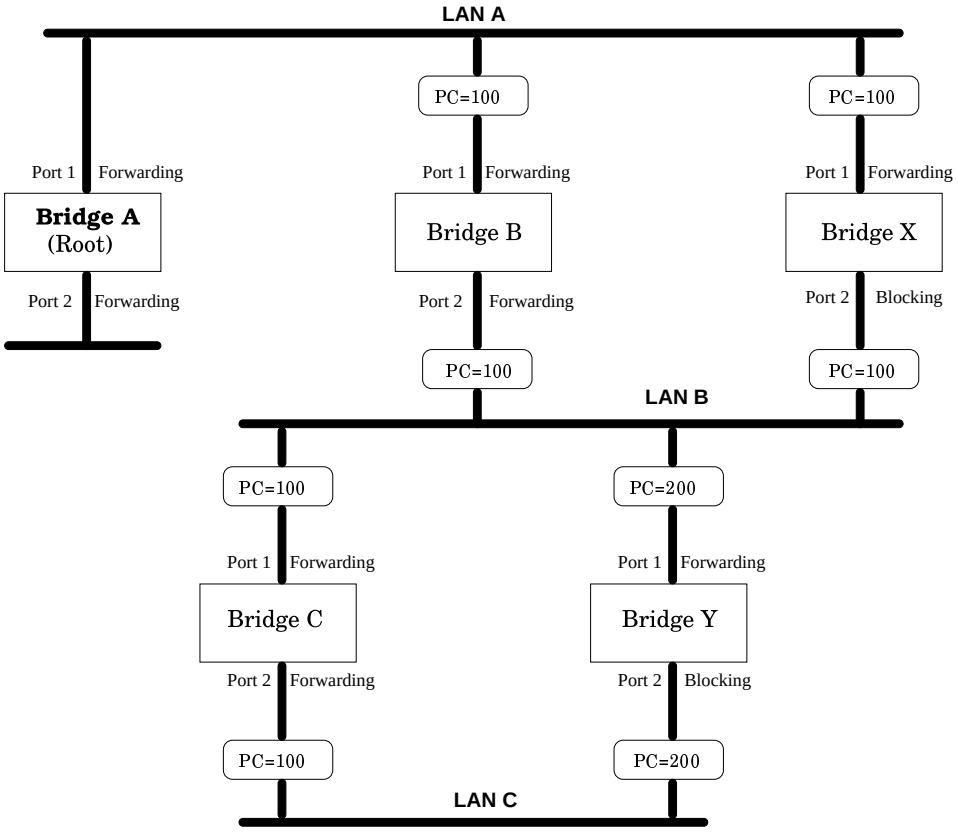

Рис. 4: STP в действии

Для большей наглядности и лучшего понимания работы STP авторы включили примеры, изображенные на рисунках 1 и 4. Комментарии к рисунку 1 расположены в самом начале этой главы. Рассмотрим теперь работу STP более подробно, для чего обратимся к рисунку 4.

Каждый порт каждого моста имеет свою величину Path Cost, обозначенную как PC=XXX. Мост A является Root Bridge, т.к. имеет наименьшее значение Bridge Identifier. Для сети A (LAN A) Designated Bridge Port является порт 1 моста A. Один из портов каждого из четырех оставшихся мостов является Root Port (это порт, ближайший к Root Bridge).

Мосты X и B предоставляют доступ к сети B с одинаковым параметром Path Cost. Однако, в данном случае, порт моста B выбран в качестве Designated Bridge Port, т.к. этот мост имеет меньшее значение Bridge Identifier.

Порт моста C выбран в качестве Designated Bridge Port для сети C, т.к. он предоставляет более «дешевый» доступ к этой сети (стоимость пути через мосты C и B равна 200, а через мосты Y и B - 300).

В стабильном состоянии все мосты ожидают периодической посылки Root Bridge специальных пакетов - Hello BPDU. Если в течение промежутка времени, определяемого значением Max Age Time, таких пакетов от Root Bridge не поступает, мост считает, что либо между ним и Root Bridge нарушена связь, либо последний отключен. В этом случае мост инициирует реконфигурацию топологии сети. Путем установки соответствующих параметров можно регулировать, насколько быстро мосты будут обнаруживать изменения в топологии и задействовать запасные маршруты. На практике с регулировкой значений STP приходится сталкиваться в основном в двух случаях - если в силу обстоятельств STP переводит в режим «запасного» более быстрый линк и если сеть имеет плохую сходимость на значениях по умолчанию из-за большого количества устройств.

2 STP & VLANs

2.1 Краткое введение в технологию VLAN

Что есть VLAN? Virtual LAN, некоторое средство логического деления сети на части. Основная идея заключается в том, чтобы выделить некую часть сети в независимую группу. В современной практике очень часто применяются для разделения доступа к той или иной части сети. Среди людей занимающихся администрированием ЛВС встречаются люди, считающие, что выделенные в VLAN сегменты сети полностью отделены друг от друга, чуть ли не на физическом уровне. Так ли это? Давайте посмотрим. Во первых, какие бывают VLAN'ы: VLAN'ы бывают MAC-based, port based и «тегированные» (определяются стандартом [4]).

MAC based VLAN – это группа рабочих станций, объединенных на основе физических адресов их сетевых адаптеров, в случае Ethernet - на основе MAC-адреса Ethernet адаптера компьютера (адреса его NIC).

Port based VLAN'ы – это VLAN'ы, создаваемые из портов коммутатора путем объединения их в группы.

Наконец, 802.1q VLAN'ы (dot 1 q), или tag-based VLAN'ы появляются в средах передачи данных с модифицированными фреймами. Модификатор фрейма представляет собой некоторую добавку к стандартному

типу фрейма. Согласно содержимому этой «добавки» устройство, обслуживающее сеть, может разделять трафик, в том числе по параметру «VLAN ID». В типичном случае, такие VLAN «ходят» внутри транковых (тегированных) каналов, или, перефразируя, между тегируемыми интерфейсами. В терминологии Cisco Systems такой порт называется «транковым» (trunk).

Особенности .1q VLANов:

- Существует некий default VLAN, тег которого определяет его VLAN id=0.
- Могут одновременно существовать нетегируемые и тегируемые порты, и те и другие могут иметь при этом членство в неких VLAN.
- При входе в не тегируемый порт, являющийся членом VLAN отличного от default, трафик «оборачивается» в некий внутренний формат, позволяющий определить какому VLAN принадлежит этот пакет.
- VLAN'ы могут соединяться между собой через нетегируемые порты. Например, порт 1 и порт 2 состоят в vlan1 и vlan2 соответственно, при этом они нетегируемые (не-транковые). Так вот, если соединить эти порты самым обычным кроссом получится объединение vlan1 и vlan2, при этом на линке в момент прохода пакта между этими портами происходит конвертация номеров VLAN, то есть трафик из порта 1 в порт 2 с VLAN ID =1 выходит из порта 1 уже с vlan id 0 (default, идентично «не обернутому» трафику), а в порту 2 оборачивается и получает VLAN ID=2. При этом, по формату фреймов, вошедших в порт 2 из порта 1 уже никак не понять, что они были когда-то в vlan1.

С точки зрения функционирования некоторого моста, сеть, в которой появляются VLAN, должна обслуживаться с иной таблицей коммутации, чем та, которая применяется для сети без VLAN. В сетях с использованием VLAN появляется еще один параметр в таблице коммутации - VLAN id и правила коммутации привязываются к этому параметру так, чтобы коммутация фреймов могла происходить только между портами, являющимися членами одной и той же VLAN. В случае, если часть портов коммутатора в одном VLAN'е, а другая часть - в другом VLAN, то тогда, с точки зрения нормального функционирования сети, можно рассматривать эту ситуацию как два независимых коммутатора. Это утверждение действительно справедливо для пользовательского трафика, однако оно не совсем верно для ситуации с STP и BPDU, обеспечивающих функциональность, требуемую алгоритмом Spanning Tree.

2.2 Миф о раздельном дереве STP в каждом VLAN

Прежде всего, следует заострить внимание на функционировании STP в системах с поддержкой VLAN. При этом надо отметить, что наиболее «продвинутые» устройства, поддерживающие VLAN (но не все), поддерживают отдельное STP дерево на каждый VLAN. Таким образом все STP атаки на такие устройства, на первый взгляд, будут эффективны только в пределах данного VLAN (по поводу VLAN см. [4]). С точки зрения STP внутри vlan мы имеем стандартную схему функционирования портов - каждый порт, согласно протоколу, получает и отправляет BPDU и, в зависимости от их содержимого, изменяет свое состояние.

Тем не менее, часть устройств, поддерживающих VLAN, имеет единственное Spanning Tree дерево на весь коммутатор (например,модели Intel 460T), что делает всю сеть, построенную на таких устройствах, уязвимой ко всем STP атакам.

С точки зрения обсуждаемых проблем, устройства второго типа с единственным на все устройство STP-деревом по части функционирования STP атак ничем не отличаются от рассматриваемого ниже случая без поддержки VLAN, так что мы не будем рассматривать отдельно этот вариант. Со вторым типом устройств, поддерживающих собственное STP-дерево на каждый VLAN, дела обстоят немного по-другому. Чтобы подойти поближе к заголовку этой главы, давайте обсудим пару практических примеров, которые могут встретиться в нормальной (рабочей) ситуации, а также к тому, как коммутатор может обеспечить работу именно по такому варианту.⊗

2.3 Миф о физической раздельности VLAN'ов

Итак, как уже указывалось ранее, port based VLAN'ы можно объединить, просто воткнув патч-корд в соседние порты, подключенные в разные VLAN. После того, как это произойдет, порты, объединенные в VLAN'ы, с точки зрения коммутатора по-прежнему будут состоять в разных VLAN - до тех пор, пока не используется GVRP (см. раздел 10, стр. 57), принадлежность порта тому или иному vlan определяется исключительно администратором (и правилом по умолчанию, если администратор не соизволил определить данный порт в тот или иной VLAN). Однако для корректного функционирования STP описанная ситуация должна отслеживаться особенным образом, то есть, с точки зрения STP, если два изначально «независимых» дерева вдруг соединились, то появился повод для STP-выборов. Действительно, аналогичная ситуация возникала на практике и алгоритм STP срабатывал. С учетом возможности конвертации VLAN ID становится понятно, что на самом

деле независимость STP дерева в пределах VLAN – это миф. Возникает идея: что если сгенерировать STP BPDU с параметрами, аналогичными тем BPDU, которые распространяются в соседнем VLAN, можно заставить STP устройство устроить перевыборы с участием интерфейса, находящегося в соседнем VLAN. Правда есть еще один вариант передачи BPDU, который позволил бы отслеживать ситуацию с VLAN соединенными, через нетегируемый порт, а именно - передавать BPDU всегда в тегируемом фрейме и игнорировать нетегируемые порты.

На момент написания этой главы авторы еще не завершили исследование стандарта [4], так что описание используемой в большинстве устройств реализации «отдельного» STP дерева на каждый VLAN пока за рамками данного исследования и будет включено лишь через некоторое, надеемся, небольшое время[6]. Впрочем, с точки зрения возможных атак на STP дерево соседнего VLAN, нет принципиальной разницы в каком фрейме передаются STP пакеты, поскольку генерация тегированных фреймов не составляет проблемы и давно реализована во многих UNIX-подобных ОС. Например в OpenBSD поддержка VLAN идет «из коробки». Разумеется поддержка VLAN есть и для Linux, однако в некоторых дистрибутивах она потребует установки «заплатки» (patch) на ядро.⊗

3 Особенности STP в гетерогенных средах на примере ATM + Ethernet

Хочется остановиться на особенностях функционирования STP в сложных сетях, а особенно – гетерогенных сетях. Расчет последствий работы атаки в таких сетях становится особенно сложным. Возьмем в качестве примера сеть, построенную с применением Ethernet технологий и ATM LANE. На рисунке 5 изображена логическая структура сети, а на рисунке 6 – ее физическая структура. До тех пор, пока мы смотрим на физическую топологию, все кажется достаточно простым. Однако, после рассмотрения логики картина значительно усложняется. Дело в том, что распространение информации о VLAN может идти и внутри одного тегируемого Ethernet линка (trunk), и внутри ATM через LANE (LAN Emulation), а маршрутизатор может играть роль моста, если на нем включена поддержка BVI (Bridging Virtual Interface). Это, кстати, является основной причиной, по которой имеет смысл поддержка STP на маршрутизаторах. Вся эта «каша» управляется, в том числе, по STP, причем количество возможных колец может исчисляться десятками за счет колец, которые могли бы образовать различные логические сущ-

[6]Этот и предыдущий подразделы требуют существенной доработки.

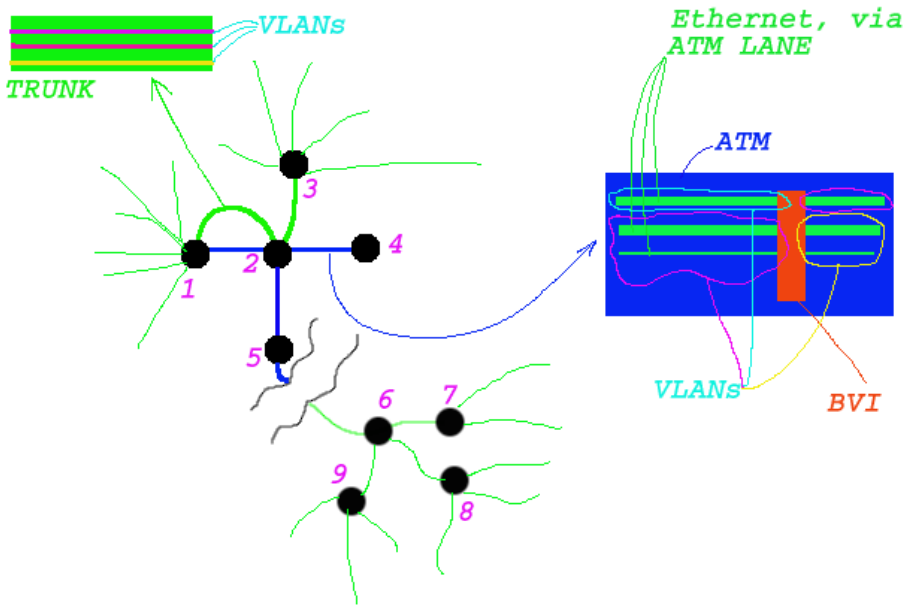

Рис. 5: Пример логической структуры сети

ности. Основная сложность состоит в том, что STP может «разрубить» кольцо в любом месте, расчет которого в случае с сетями подобной сложности может оказаться делом нетривиальным. Очевидно, что если сети разделяются через BVI, то есть немалые шансы, что STP-DoS может распространяться поверх ATM. В любом случае ATM соединения с использованием LANE могут прозрачно передать атаку через LANE, поскольку LANE – это своего рода прозрачная инкапсуляция фреймов Ethernet в ячейки ATM[7].

4 Замечания, вытекающие из анализа RFC 2878

Согласно RFC 2878:

> Two basic algorithms are ambient in the industry for Bridging of Local Area Networks. The more common algorithm is called «Transparent Bridging», and has been standardized for Extended LAN configurations by IEEE 802.1. The other is called «Source Route Bridging», and is prevalent on IEEE 802.5 Token Ring

[7] Да простят нам это обобщение любители точных описаний ;)

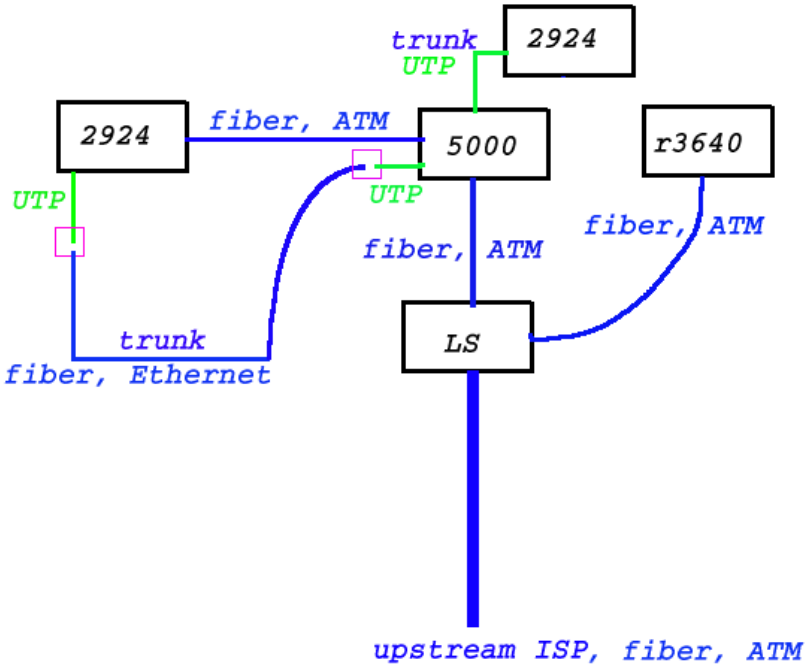

Рис. 6: Пример физической структуры сети

LANs.

Поскольку RFC доступны в виде ASCII-текстов, их цитирование гораздо удобнее, чем IEEE-стандартов, распространяемых в формате pdf (см. раздел 20). В этом разделе цитаты из RFC 2878 в связи с большим их количеством будут просто браться в кавычки и выделяться в отдельный абзац, без специального упоминания о том, какой именно цитируется документ - достаточно одного раза.

> The implementor may model the line either as a component within a single MAC Relay Entity, or as the LAN media between two remote bridges.
>
> The IEEE 802.1G Remote MAC Bridging committee has proposed a model of a Remote Bridge in which a set of two or more Remote Bridges that are interconnected via remote lines are termed a Remote Bridge Group. Within a Group, a Remote Bridge Cluster is dynamically formed through execution of the spanning tree as the set of bridges that may pass frames among each other.
>
> We allow for modelling the line either as a connection residing between two halves of a «split» Bridge (the split-bridge model), or as a LAN segment between two Bridges (the independent-bridge model). In the latter case, the line requires a LAN Segment ID. By default, PPP Source Route Bridges use the independent-bridge model.
>
> Given that source routing is configured on a line or set of lines, the specifics of the link state with respect to STE frames are defined by the Spanning Tree Protocol in use. Choice of the split - bridge or independent-bridge model does not affect spanning tree operation. In both cases, the spanning tree protocol is executed on the two systems independently[8].

Таким образом, для систем, использующих «Source Route Bridging», STP-атаки из удаленного сегмента во многих случаях не страшны. Внутри таких систем, соединенных WAN-линком, существуют независимые STP-деревья, так что максимум, чего мог бы добиться атакующий - опустить WAN-link, передавая STP пакеты от имени удаленного designated root. Однако даже это может оказаться невозможным - смысл в поддержке STP пакетов для линка между двумя различными STP-сущностями не очевиден.

[8] Здесь STE frame - Spanning Tree Explorer frame - специфичный для 802.5 MAC фрейм.

> The Bridging Control Protocol (BCP) is responsible for configuring, enabling and disabling the bridge protocol modules on both ends of the point-to-point link.
>
> BCP packets may not be exchanged until PPP has reached the Network-Layer Protocol phase.
>
> Before any Bridged LAN Traffic or BPDUs may be communicated, PPP MUST reach the Network-Layer Protocol phase, and the Bridging Control Protocol MUST reach the Opened state.

Из главы 4.1.4. [5] «Separation of Spanning Tree Domains»

> It is conceivable that a network manager might wish to inhibit the exchange of BPDUs on a link in order to logically divide two regions into separate Spanning Trees with different Roots (and potentially different Spanning Tree implementations or algorithms). In order to do that, he should configure both ends to not exchange BPDUs on a link. An implementation that does not support any spanning tree protocol MUST silently discard any received IEEE 802.1D BPDU packets.

То есть, во-первых, администратор может разделить STP в связанных WAN линком сетях посредством выключения STP на обоих концах. Во-вторых, если подцепить несовместимый с STP свитч, то согласно этому RFC он должен уничтожать STP-пакеты. Причем, после прочтения этой главы остается такое впечатление, что по умолчанию STP ходит и его надо специально выключать, чтобы это поведение отменить. Видимо в этом месте авторы RFC забыли обратить внимание на то, что это относится не к «Source Route Bridging» (иначе получается противоречие), а к «Transparent Bridging».

Стоит сказать пару слов об RFC 1638, в котором описан старый алгоритм работы протокола BCP и прочие подробности, в том числе пересмотренные в [5]. Основной смысл (с точки зрения тематики этой статьи) сводится к тому, что в старых версиях PPP BCP некоторые параметры не использовались, однако устройства, работающие по спецификации RFC 1638, также способны передавать STP-BPDU. С точки зрения STP атак не важно, какой RFC определяет их хождение - главное, что они ходят. Собственно, формат фреймов проходящих по PPP WAN линку определяется в [5] в главах 4.2 и 4.3, а немного ниже в [5], в главе 4.4, говорится, что для определения корректной топологии, состояния VLAN, регистрации участия в мультикастовых группах мосты (или свитчи, а с точки зрения этой статьи - просто некие устройства поддерживающие

данные протоколы) обмениваются BPDU по спецификации STP, GVRP, GMRP (и вообще GARP):

> To avoid network loops and improve redundancy, Bridges exchange a Spanning Tree Protocol data unit known as BPDU. Bridges also exchange a Generic Attributes Registration Protocol data unit to carry the GARP VLAN Registration Protocol (GVRP) data and GARP Multicast Registration Protocol (GMRP). GVRP allow the Bridges to create VLAN groups dynamically. GMRP allows bridges to filter Multicast data if the receiver is absent from the network. These Bridge protocols include Spanning Tree Protocol and GARP protocols data units are carried with a special destination address assigned by the IEEE.

Ну, и, наконец, в следующем абзаце определяется самое главное в контексте STP-атак:

> These bridge protocols data units and GARP protocol data units must be carried in the frame format shown in section 4.2 or 4.3.

С оговоркой, о совместимости со старыми устройствами, но тем не менее обязательной поддержкой этих протоколов, а значит и STP:

> But there is one exception to this rule: if the bridge is connected to an old BCP bridge [10][9] and can support backward compatibility, it MUST send the BPDU in the old format described in Appendix A.

Резюмируем: в принципе PPP WAN линки могут служить переносчиками STP атак.

Теперь о деталях. В BCP методика действий сходна с PPP LCP:

> BCP uses the same packet exchange mechanism as the Link Control Protocol.

LCP, грубо говоря, служит для определения того, какие потенциально возможные опции устанавливаемого линка будут применены в данном конкретном соединении. В PPP LCP используется система обмена запрос/ответ. Аналогичный подход и в BCP:

> If a Configuration Option is not included in a Configure-Request packet, the default value for that Configuration Option is assumed.

[9] Здесь ссылка [10] - ссылка на RFC 1638.

То есть могут использоваться какие-то значения по умолчанию. Какие же? Давайте посмотрим - ниже будут замечания по этим значениям, а пока стоит взглянуть на главу 5.6 в [5]. Эта глава описывает поведение устройств при работе с STP. Первым же абзацем идет определение совместимости со старым форматом (RFC 1638):

> The Spanning-Tree-Protocol Configuration enables a Bridge to remain compatible with older implementations of BCP [10][10].This configuration option is, however, incompatible with the Management-Inline option, which enables a bridge to implement the many protocols that IEEE now expects a bridge to be able to use.

Далее определяется возможное поведение системы, в случае, если на другом конце стоит устройство, работающее по RFC 1638, и при этом выбирается совместимый с ним алгоритм работы:

> If both bridges support a spanning tree protocol, they MUST agree on the protocol to be supported. The old BPDU described in Appendix A MUST be used rather than the format shown in section 4.2 or 4.3. When the two disagree, the lower-numbered of the two spanning tree protocols should be used. To resolve the conflict, the system with the lower-numbered protocol SHOULD Configure-Nak the option, suggesting its own protocol for use. If a spanning tree protocol is not agreed upon, except for the case in which one system does not support any spanning tree protocol, the Bridging Control Protocol MUST NOT enter the Opened state.

Здесь стоит сделать следующий комментарий - согласно RFC 2878 Spanning Tree протоколов несколько, каждый из которых пронумерован согласно RFC 1700 [7]:

0 Null (no Spanning Tree protocol supported)
1 IEEE 802.1D spanning tree
2 IEEE 802.1G extended spanning tree protocol
3 IBM Source Route Spanning tree protocol
4 DEC LANbridge 100 Spanning tree protocol

Очевидно, все эти протоколы имеют некие общие свойства, позволяющие называть их Spanning Tree протоколами, что, кстати, наталкивает на мысль, что все они могут быть подвержены, как минимум части описываемых здесь атак на Spanning Tree Protocol[11], пронумерованный в [7]

[10]Здесь ссылка [10] - ссылка на RFC 1638.

[11]В данной статье пока рассматривается только IEEE реализация STP, однако в дальнейшем мы планируем включить анализ и остальных реализации ST-алгоритма.

как 1. Поскольку часть сказанного в данной статье относится к Spanning Tree алгоритму вообще, то слабость этих протоколов очевидна.

Приложение A в [5] содержит аналогичную таблицу для значений передаваемых в поле идентификатора STP протокола, согласно [7]:

Value (in hex)	**Protocol**
0201	IEEE 802.1 (either 802.1D or 802.1G)
0203	IBM Source Route Bridge
0205	DEC LANbridge 100

Далее декларируется возможность PPP WAN линков работать сразу с несколькими типами STP:

> Most systems will only participate in a single spanning tree protocol. If a system wishes to participate simultaneously in more than one spanning tree protocol, it MAY include all of the appropriate protocol types in a single Spanning-Tree-Protocol Configuration Option.

А вот еще одно важное замечание: если устройства на обоих концах не могут прийти к согласию по поводу того, какой STP протокол использовать, то выигрывает протокол с меньшим номером, то есть, IEEE реализация, если STP вообще поддерживается, конечно:

> When the two disagree, the lower-numbered of the two spanning tree protocols should be used. To resolve the conflict, the system with the lower-numbered protocol SHOULD Configure-Nak the option, suggesting its own protocol for use. If a spanning tree protocol is not agreed upon, except for the case in which one system does not support any spanning tree protocol, the Bridging Control Protocol MUST NOT enter the Opened state.

Выделенное заглавными буквами весьма важно - если с обоих сторон поддерживается STP, то он должен обязательно быть использован. Далее сказано, что по умолчанию либо STP поддерживает хотя бы IEEE реализацию, либо не поддерживается вовсе:

> By default, an implementation MUST either support the IEEE 802.1D spanning tree or support no spanning tree protocol.

Теперь рассмотрим эту ситуацию в следующем теоретическом случае, очень близком к повсеместной практике. Допустим, мы имеем ISP с dialup доступом, построенном на маршрутизаторе доступа. Если этот маршрутизатор доступа поддерживает STP (например, многие модели Cisco), то, вероятно, PPP линки, которые организуются между ISP и его клиентом, будут поддерживать STP со стороны ISP. В нормальной ситуации PPP клиенты не запрашивают поддержку STP просто за ненадобностью, однако, если вместо обычного пользователя PPP соединение с провайдером установит вредитель, он, вероятно, сможет, модифицировав процедуру установления PPP соединения, реализовать часть STP-атак на сеть провайдера.

Стоит заметить, что согласно главе 5.7 [5], по WAN линкам может также передаваться информация о VLAN (и прочих опциях определяемых тегированными фреймами 802.1q [4]). При этом определено, что если удаленное устройство не заявило о поддержке 802.1q, то тегированные фреймы не должны ему посылаться. Однако ситуация, при которой злоумышленник будет посылать со своей стороны такие фреймы не определена, что позволяет предположить различную реакцию со стороны устройств различных производителей:

> By default, IEEE 802 Tagged Frame is not supported. A system which does not negotiate, or negotiates this option to be disabled, should never receive a IEEE 802 Tagged Frame.

Не следует забывать, что помимо PPP существуют еще и другие WAN-линки, например, ADSL based, на которые есть свои стандарты. Они также могут допускать свободное хождение STP по умолчанию - это дотошный читатель может выяснить самостоятельно.

5 Комментарии к написанию кода, генерирующего STP пакеты

Глава 8.3.2 в [1] содержит следующую фразу: «A MAC frame conveyng a BPDU». Это значит, что BPDU это *пакет*, а не фрейм. Таким образом, нет необходимости конструировать фреймы самостоятельно, поскольку они стандартны для данной физической среды - мы можем конструировать BPDU пакеты выше, чем самый нижний подуровень 2-го уровня OSI. Иначе говоря, можно отвязать программу конструктор BPDU от конструктора фреймов, оставив функцию инкапсуляции соответствующему драйверу. Однако, авторы выбрали для своих исследований создание программного модуля, конструирующего BPDU напрямую. Исходный текст модуля на языке C и сценарий на языке bash для более

удобного манипулирования параметрами командной строки приведены в приложениях[12].

Чем хорошо создание фреймов напрямую:

1. Можно ставить произвольный MAC-адрес источника, что значительно осложняет определение реального источника атаки. До тех пор покуда не применяется насильственный line-jamming (заявка коллизий в их отсутствие в CSMA/CD сетях [6]), сегмент, из которого осуществляется атака, ничем не отличается от любого другого сегмента, поскольку данная атака нарушает исключительно работу коммутаторов, то есть межсегментное соединение, но не работу в пределах сегмента.

2. Если конструировать фреймы самостоятельно, можно добиться ситуации при которой атака идет как бы параллельно с нормальной работой, маскируя таким образом атакующего за счет передачи пакетов с «фальшивого» MAC адреса одновременно с нормальной работой в сети со «своего» MAC адреса.

3. По мнению авторов, пока не существует стандартного (одинакового для всех систем) способа формировать BPDU иначе, чем конструируя фреймы полностью. Это говорит о том, что реализации алгоритмов таким методом для FreeBSD, OpenBSD, Linux и других OS будут существенно различны, в результате чего лучшим способом будет все же посылка BPDU в полностью самостоятельно сконструированных фреймах.

Чем плохо создание фреймов, а не пакетов Spanning Tree:

1. Поскольку Spanning Tree протокол определен для различных типов MAC[13], код получится непереносимым между ними.

Ethernet был выбран авторами из-за наибольшей распространенности, по сравнению со всеми остальными типами сетей.

[12] Авторы долго не могли прийти к единому мнению: стоит ли делать эти тексты общедоступными. Без понимания принципов работы STP и выбора параметров эти тексты ничем не помогут личностям с деструктивными наклонностями, а человек, который разобрался в сути уязвимости, может воспользоваться любым конструктором пакетов. В связи с этим, тексты решено было включить в открытую публикацию.

[13] Media Access Control (управление доступом к среде передачи данных). Например, Ethernet, Token Ring и т.п.

Помимо написания собственных программ есть возможность использовать готовые программные средства, обернув их в скрипты на любом подходящем языке, например, perl или языках shell - bash, zsh, ksh, csh (см. раздел 9), что значительно упрощает работу с параметрами Spanning Tree. Однако при этом появляется дополнительное условие - Linux с ядром, настроенным для поддержки bridging плюс соответствующий комплект утилит. Еще одним отрицательным моментом выбора такого пути будет необходимость накладывания патчей на исходный код Linux bridging project в случае, если потребуется сконструировать атаку с модификацией алгоритма работы, выходящей за рамки стандарта, например, изменить значения таймеров на меньшие или большие, чем описано в стандарте.

Стоит также сказать несколько слов на тему особенностей программирования под разные операционные системы и выбора языка:

Windows-подобные системы. В этих системах, исключая Windows XP, нет поддержки raw sockets, что не позволяет без специального драйвера (VxD) конструировать фреймы напрямую - как обычно гибкость windows систем «из коробки» оставляет желать лучшего. Поэтому под windows системы и unix системы это будут совсем разные реализации. Ну, разве что, писать на perl, который, однако установлен не на каждой windows системе. Еще вариант - использовать java, приложения на которой можно запускать автономно при помощи jre.

Unix-подобные системы. В качестве языка лучше всего выбрать C-компиляторы - они есть во всех unix-подобных системах практически под любую платформу. Впрочем, можно было выбрать и perl - там тоже есть raw-sockets. Возможны проблемы портирования на другие unix-подобные системы с linux, т.к. отличаются названия интерфейсов. Но это можно легко решить при помощи Makefile. А вот если различаются API – придется потрудиться побольше.

6 Возможные схемы атак

Идея этих атак лежит на самой поверхности. Поразительно, что потратив 3 часа на поиски я так и не наткнулся на полное описание атак возможных при помощи протокола Spanning Tree - одни лишь намеки и частные случаи. Возможно из-за лени администраторов? :-) Для того чтобы устроить Отказ в Обслуживании можно воспользоваться тем, что STP-совместимые устройства в момент реконфигурации работают не на пользователя, а лишь на создание Spanning Tree дерева. Поскольку реконфигурация может быть вызвана, в том числе, появлением

нового STP-совместимого устройства, можно имитировать периодически появление нового устройства с параметрами, которые будут лучше установившихся, что вызовет реконфигурацию (перевыборы) одного из выборных параметров. Некоторые (глупые) устройства будут пытаться пересматривать состояние всех портов при появлении любого нового устройства даже с худшими (с точки зрения выигрыша в выборах) чем у них параметрами. На более умное железо эта атака может действовать исключительно в случае появления устройства с лучшими (с точки зрения STP) параметрами - в худшем случае может пройти состояние реконфигурации только один порт. Однако даже такая ситуация будет противоречить спецификации протокола [1], в котором не предусмотрена ситуация переконфигурации устройства, к которому подключают другое с худшими (с точки зрения STP) параметрами. Поскольку предметом атаки могут быть любые «выборные» значения, мы можем выделить несколько типов атак по используемым методам их проведения:

1. Инициация перевыборов назначенного корня для всей сети.
2. Инициация перевыборов назначенного моста для сегмента сети.
3. Инициализация перевыборов назначенного порта для сегмента сети.

Как будет показано ниже в разделе 6.4.5 случай 3) часто сопутствует случаю 2).

Здесь и далее под «лучшим» понимается такое значение параметра, содержащегося в BPDU, которое позволяет добиться рассматриваемой цели. Чаще всего под «лучшим» будет пониматься такое значение параметра, которое обеспечит победу в STP-выборах.

В качестве побочных эффектов STP-атак могут образовываться «пакетные штормы» внутри сети (в большинстве случаев - недолговременные).

6.1 BPDU spooffing

Эта часть статьи была добавлена уже примерно на середине разработки. Как вы сможете заметить далее, эта методика нужна для части описываемых здесь алгоритмов и изначально мы обошли ее вниманием, считая, что в ней нет ничего такого, что могло бы его заслужить. Тема с общим названием «spoofing» разжевана во многих источниках настолько хорошо, что, казалось, ничего нового сказать нельзя. Однако в применении к STP и с учетом уровня OSI, на котором работает Spanning Tree, у этой методики есть несколько принципиальных особенностей. Говоря

кратко, spooffing – это подмена. Наиболее типична подмена адреса отправителя - случай, в котором потребуется подменить адрес получателя, придумать трудно, хотя и можно, к тому же, почти все они будут относится исключительно к локальному для данной машины стеку TCP/IP, за исключением разве что методик отлова недостаточно хорошо спрятанных promiscuous интерфейсов. Типичные способы борьбы с этой атакой в Internet заключаются в:

1. введении граничных условий;
2. введении дополнительных требований в протокол обмена между source и destination;
3. введении требования установления соединения между участниками обмена трафиком (в принципе это частный случай 2).

Особенностями STP, важными с точки зрения детального рассмотрения BPDU spooffing, является следующее:

1. Это протокол 2-го уровня OSI, который не маршрутизируем.
2. C-BPDU пакеты этого протокола ходят исключительно между двумя STP-совместимыми устройствами. Любое STP-совместимое устройство, получив такой пакет, генерирует в качестве реакции свой (то есть другой) пакет, ограничивая таким образом зону хождения C-BPDU.
3. Этот протокол не предполагает каких-либо административных ограничений на его использование.
4. Этот протокол не предусматривает входной фильтр хождения BPDU по интерфейсу с которого оно получено, поскольку сам факт получения BPDU на том или ином интерфейсе является событием протокола[14].
5. Этот протокол предусматривает динамическое конфигурирование. Причем, ограничений на факт и время появления в структуре нового устройства не предусмотрено.
6. Это протокол без установки соединения (connection-less protocol).

[14]Исключения возможны при поддержке расширений, например Cisco BPDU Root Guard, см. раздел 8, стр. 51.

Из 1 и 4 следует, что граничные условия фильтрации ложных пакетов к нему не применимы[15].

Из 2 следует, что фильтрация пакетов по пути неким STP-совместимым устройством не имеет смысла, поскольку путь STP-пакета не может содержать промежуточное STP устройство. 3 говорит сам за себя. 6 указывает на то, что подделка соединения не потребуется, а между тем она затрудняет, например, TCP/IP spooffing. Впрочем, введение требования установки соединения в данном случае поможетмало, поскольку нельзя ограничивать возможность установки такового.

В результате ситуация с BPDU spooffing такова: при правильно поставленном BPDU-spooffing принципиально невозможно отличить «легальные» пакеты от поддельных. Под правильно поставленным BPDU-spooffing понимается подделка не только параметров STP пакета, но и адресов в MAC фрейме. При этом, благодаря 5, даже если не использовать собственно spoofing, можно все равно добиться интересных результатов.

Играя с времяобразующими параметрами Max Age, forward delay, hello time в фальсифицированных C-BPDUs мы могли бы управлять, по крайней мере, близлежащими STP-устройствами посредством постоянной и очень быстрой посылки таких C-BPDU. Под «фальсифицированными bpdu» в данном случае понимаются c-bpdu с такими же параметрами, как у текущего designated bridge или designated root bridge, и с тем же, что и у него source MAC, но с другими значениями параметров, определяющих различные паузы в рамках протокола, что, правда, может вызвать один из случаев описанных в этой статье - переконфигурацию STP-дерева с последующим частичным DoS. Для того чтобы этого избежать фальсифицированные пакеты не должны вызывать реконфигурацию дерева. Этого можно добиться, выставив в них port id в большее значение (согласно протоколу при этом должен выключиться этот интерфейс), или же выставить худший path-cost - в этом случае STP выключит этот порт, но BPDU тем не менее он будет принимать. Однако само по себе это мало что дает - все, что мы получим, меняя эти параметры, это незначительная «дезориентация» ближайшего свитча и свитчей, который получают информацию от уже «обманутого» (те, что находятся ниже в STP-дереве). То есть максимум, что это может дать - увеличение

[15]По крайней мере до тех пор, пока нет уверенности, что на определенных интерфейсах получение BPDU принципиально невозможно (или не приемлемо по каким либо соображениям). Такую уверенность может иметь администратор, однако сам по себе стандарт в данном случае не предусматривает возможности отключения STP поинтерфейсно - в таком поведении стандарта есть смысл только до тех пор, пока безопасность имеет меньший приоритет, чем удобство.

времени жизни некорректной структуры дерева после того, как мы перестали передавать пакеты по этому атакующему алгоритму. В случае возникновения реальной STP-переконфигурации, возможны проблемы из-за появления data loops. По сравнению с уже затронутыми вариантами DoS это просто не серьезно. Однако использование Max Age может привести к тому, что DoS по одному из описанных выше алгоритмов будет функционировать дольше.

6.2 Provocation Aging

Так же следует заметить, что возможна подделка вообще любых bpdu: не только configuration-bpdu (c-bpdu), но и notification bpdu (n-bpdu, tcn-bpdu). Например, согласно последнему абзацу главы 7.9.2 в [1], используя tcn-bpdu, можно реализовать еще одну атаку - «provocation aging» - управление величиной времени хранения информации о положении в сети MAC-адресов, динамически формируемой в процессе работы устройства. Согласно [1] стр. 45, таблица 7.4, величина Aging Time может быть сброшена не ранее, чем через 10 секунд[16].

Сама по себе эта атака малофункциональна, но, теоретически, может быть использована для других атак, при которых одним из требований является внесение ложных записей в таблицу коммутации в качестве катализатора, обеспечивающего повышение скорости срабатывания такой атаки. Возможно, использование этой методики для помощи в provocation sniffing (см. раздел 6.6, стр. 46) поможет дать боле-менее существенные результаты. Кроме того, эта атака может быть использована в случае поддержки STP-совместимым устройством расширений STP, например STP port fast (Cisco). Как будет показано, ниже в таких случаях aging time может быть сброшено до нулевой величины. В таком случае атака provocation aging станет синонимом provocation sniffing. Впрочем, об этом ниже.

6.3 BPDU filter

Очевидно, что осуществить DoS в рамках одного коммутатора очень легко. Для этого необходимо организовать data-loop, который послужит причиной лавинного размножения и саморегенерации пакетов. Данная атака, разумеется, эффективна в пределах одного vlan, однако, она затронет и остальные за счет снижения производительности всего устройства. Это происходит из-за повышения накладных расходов на обработку постоянного большого паразитного трафика, объем которого может вырасти до суммарной пропускной способности образованных между

[16] Однако и это тоже неплохо бы проверить ;)

устройствами линков. Идея этой атаки крайне проста: создается дополнительный link между двумя STP-совместимыми устройствами, в середину которого ставится фильтр BPDU, который просто «тупо» вырезает BPDU с любыми source & destination, не влияя на остальной траффик, либо, дополнительно создавая его. В силу своей неэлегантности, и в силу того, что для реализации обязательно требуется доступ к включенным портам хотя бы одного из коммутаторов этот метод авторы считают мало интересным с практической точки зрения.

6.4 Отказ в обслуживании (DoS)

Авторы предполагают, что аудитория достаточно хорошо понимает сам термин. Если же этот термин не знаком, советуем отложить чтение этой статьи и внимательно прочесть [3], либо другую обзорную литературу содержащую необходимый теоретический минимум. Тем не менее, авторы приведут одно из многочисленных (неакадемических) определений DoS:

> Атака Denial Of Service лишает объект атаки возможности работать по назначению, иными словами лишает пользователя возможности получить от объекта, подвергшегося этой атаке, некую функциональность, предоставлять которую он должен в «нормальных» условиях.

Отказ в обслуживании – это довольно распространенная атака. Однако до настоящего момента ее применение в основном ограничивалось глобальными сетями, а также локальными сетями, использующими протоколы глобальных сетей (наиболее распространены DoS-атаки связанные с реализацией или принципами работы стека протоколов TCP/IP). Если классифицировать атаки по протоколам, то история атак на уровнях выше 2-го OSI гораздо богаче. Этот документ добавит описание еще нескольких экземпляров в немногочисленную коллекцию атак, возможных на 2-м уровне OSI. С использованием STP возможны две схемы реализации DoS.

6.4.1 STP DoS: «вечные выборы» или постоянный перебор

Наиболее эффективный метод: подождать появления STP пакета с текущим STP-root, затем по очереди перебирать значения bridge id, посылая bpdu с все меньшими значениями ($id = id - 1$), до тех пор пока не будет достигнуто предельное значение, вызывая таким образом перевыборы designated root каждым посланным пакетом (для большей уверенности - посылать одинаковые c-bpdu несколько раз подряд). Когда же будет достигнуто минимально возможное значение - подождать,

пока это значение не устареет из-за паузы, и начать сначала. С учетом того, что все параметры, включая время устаревания, устанавливаются в посылаемых назначенным корнем конфигурационных bpdu, можно получить ситуацию, при которой порты никогда не войдут в состояние «forwarding» пока происходит генерация фреймов, обеспечивающих отказ в обслуживании. Более того, в силу особенностей протокола состояние отказа в обслуживании будет продолжаться еще некоторое время, равное параметру Max Age, который можно выставить согласно стандарту[17] до 40 секунд (см. [1], стр. 108, таблица 8.3). Поскольку помимо 65535 возможных приоритетов bridge id включает в себя еще и MAC адрес, то количество времени, которое потребуется чтобы перебрать все возможные значения весьма велико и составит (в шестнадцатеричной системе счисления):
$(CurrentVictimBridgePriority - 1 + VictimBridgeMAC - 1) *$
$(ListeningTime + LearningTime) =$
$(CurrentVictimBridgePriority + VictimBridgeMAC - 2) *$
$(ListeningTime + LearningTime) =$
$(CurrentVictimBridgePriority + VictimBridgeMAC - 2) *$
$2 * ForwardDelay$ секунд.

Значение по умолчанию для ForwardDelay составляет 15 секунд и может доходить до 30. На самом деле при зацикливании алгоритма состояние DoS может продолжаться сколь угодно долго, пока посылаются пакеты с фальшивыми BPDU. При этом, как видите, вариаций алгоритма можно найти довольно много.

6.4.2 STP DoS: алгоритм «исчезновения корня»

Начать посылку BPDU с минимально возможным bridge id, то есть с максимально возможным приоритетом, периодически переставая передавать конфигурационные bpdu, чтобы это наше значение назначенного корня устарело и так по кругу. На первый взгляд это не настолько эффективно, поскольку может существовать небольшой промежуток времени, когда сеть работает, тем не менее, в силу того, что ограничения, накладываемые спецификацией протокола, позволяют нам устанавливать время устаревания значений в очень широких пределах (см. раздел 6.4.7), этим методом можно достигнуть точно такой же эффективности. Более того, этот метод наиболее прост в реализации, поскольку не требует от нас ни знания текущего идентификатора назначенного корня, ни каких-либо предположений относительно его величины (в отличие от предыдущего случая).

[17] Это если программисты не забыли поставить знак неравенства. А если забыли - вплоть до максимальной величины, вмещающейся в отведенную под этот параметр область памяти. Опять же, разные производители зачастую совершенно по-своему соблюдают совместимость с тем или иным IEEE стандартом. ;)

Следует заметить, что если в некоторой ЛВС текущий идентификатор designated root не является идентификатором наивысшего возможного приоритета (имеется ввиду нулевые и Bridge Priority и MAC address), то эта ЛВС, скорее всего, подвержена обоим типам DoS в полной мере. В случае же, если на одном из STP-совместимых устройств установлен bridge id равный 0, то DoS, тем не менее, может быть реализована, но только на части устройств, а именно на тех, к которым подключен атакующий. Дело в том, что ситуация, когда мы имеем два STP-совместимых устройства с одним и тем же bridge id может быть воспринята как кольцо, и STP отключит либо порт атакующего, либо другой порт, в зависимости от соотношения номера порта, на котором производится атака, и номера порта к которому подключена атакуемая часть сети (см. раздел 1).

6.4.3 STP DoS: алгоритм случайного совпадения

Этот подраздел касается случаев сетей с использованием технологии port-based VLAN. В случае такой организации сети описанные до этого момента DoS атаки могут сработать исключительно в пределах одного VLAN. Однако и в этом случае злоумышленник может наделать неприятностей всей сети - дело в том, что, как рассматривалось выше, STP может выделяться в отдельное дерево посредством фильтрования по bridge ID[18]. Поэтому, если в рамках данного VLAN отправлять BPDU от имени имеющегося в соседнем VLAN designated root или просто designated bridge, коммутаторы можно убедить в том, что в некотором месте VLAN'ы стали вдруг как-то соединены. Это вызовет перевыборы designated bridge, в том числе и в сегменте, который атакуется и принадлежит VLAN, отличному от VLAN атакующего. $\otimes$

Этот алгоритм назван алгоритмом случайного совпадения потому, что атакующему придется подбирать designated root bridge ID именно такой, который совпадет с имеющимся в соседнем VLAN. Разумеется наиболее точно эта атака воспроизводится в случае, если атакующий может получить dump работы сети за время порядка Hello Time (по умолчанию - 2 секунды).

6.4.4 STP DoS: другие возможные алгоритмы

Возможны также произвольные комбинации вышеописанных методик. Например, можно сделать вариант 2 не с максимально весомым bridge identifier, а просто с текущим $designatedroot+1$ и периодически его анонсировать, по мере устаревания. Однако вариации на тему пройденного,

[18]Это всего лишь один из возможных вариантов, пока требующий подтверждения.

разумеется не представляют интереса, поэтому в дополнение к сказанному отдельно следует рассмотреть атаки, которые могут произвести отказ обслуживания в пределах некоторого сегмента ЛВС, тем самым создав ситуацию «частичного отказа в обслуживании».

6.4.5 STP DoS: Частичный отказ в обслуживании

В случае, если атакующему каким-то образом, стало известно значение bridge identifier одного или более устройств в ЛВС, он может вызвать выборы designated bridge для того или иного участка ЛВС, послав соответствующее c-bpdu для оспаривания статуса назначенного моста для соответствующего сегмента ЛВС. Воздействуя исключительно на ближайшее к атакующему устройство, он может объявить себя лучшим возможным путем к соседнему с ним устройству, например, заявив, что его порт является самым дешевым по стоимости путем к нему, в результате чего «более дорогой» порт будет переведен в состояние Blocking, а поскольку реальной связи нет, все оконечные устройства, подключенные к атакованному устройству, потеряют возможность работать с отключенным сегментом ЛВС[19].

В описанном на рисунке 7 случае в примере B), псевдо-бридж-система атакующего, анонсируя лучшие параметры, выиграла выборы назначенного моста (designated bridge), не являясь таковым. Одновременно порт, к которому подключен атакующий, стал назначенным портом для атакуемой части сети (статус перешел от порта перешедшего в состояние Blocking). Следует отметить, что при переходе статуса назначенного моста для сегмента в результате STP атаки, в случае, если оспаривавшие этот статус устройства были подключены к разным портам, изменяется и назначенный порт. С точки зрения пользователя сети сегмент «Victim LAN» перестает «видеть» остальную сеть (все, что подключено через Bridge 1), при этом атакующий (attacker) продолжает видеть все сегменты сети, кроме сегмента «Victim LAN». Объектом атаки является «Bridge 1», которому посылаются BPDU от имени «Bridge 2», но с «лучшими» (с точки зрения выигрыша STP-выборов) параметрами.

Пример A) отличается от B) тем, что будучи подключенным к тому же порту «Bridge 1» атакующий (attacker) также, как и жертва («Victim LAN») перестает «видеть» остальную сеть. Объектом атаки является «Bridge 1», которому возвращаются BPDU, полученные от него же и измененные так, чтобы сэмулировать закольцовывание между этим и другим интерфейсом. При этом параметры таких BPDU должны быть

[19] Данная атака может оказаться невозможной без физического подключения к соответствующему сегменту. Авторы намерены провести дополнительные эксперименты.

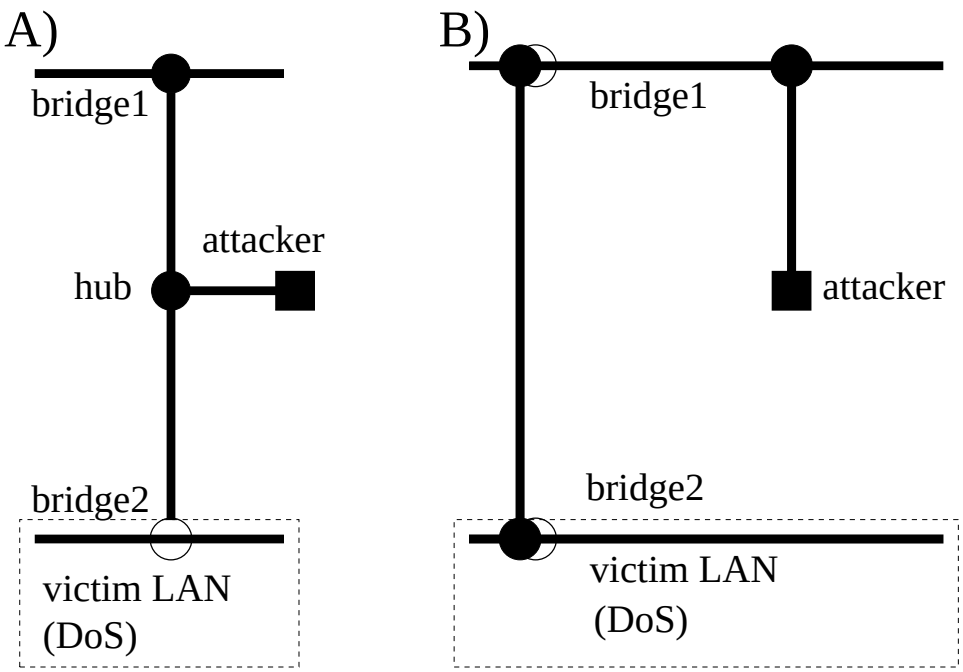

Рис. 7: Частичный DoS (пример 1)

подобраны именно так, чтобы выключился именно этот, а не соседний интерфейс.

Вариант A) может быть также реализован в случае если атакующий (attacker) подключен не к «Bridge 1» через концентратор, а к «Bridge 2» - в этом случае объектом атаки будет «Bridge 1», которому для реализации частичной DoS по этой схеме необходимо присылать BPDU от имени «Bridge 1» с параметрами «лучшими», чем имеющееся подключение между «Bridge 1» и «Bridge 2». Собственно, по этой схеме можно добиться DoS для любого порта на том же STP-совместимом устройстве. Эти схемы проиллюстрированы рисунком 8. При разумном подборе параметров BPDU это становится возможным также и для портов на соседних STP-совместимых устройствах.

Более типичной будет схема показанная на рисунке 9. В реальных условиях у злоумышленника редко есть возможность расцепить линк между коммутаторами, «врезать» туда концентратор и вставлять в поток свои пакеты. Скорее возможен такой сценарий: сервера включены в один коммутатор, клиенты включены в концентраторы, подсоединенные к одному или нескольким коммутаторам. По аналогии атаки Митника на Шимомуру, атакующему надо «выключить» одного из участников со-

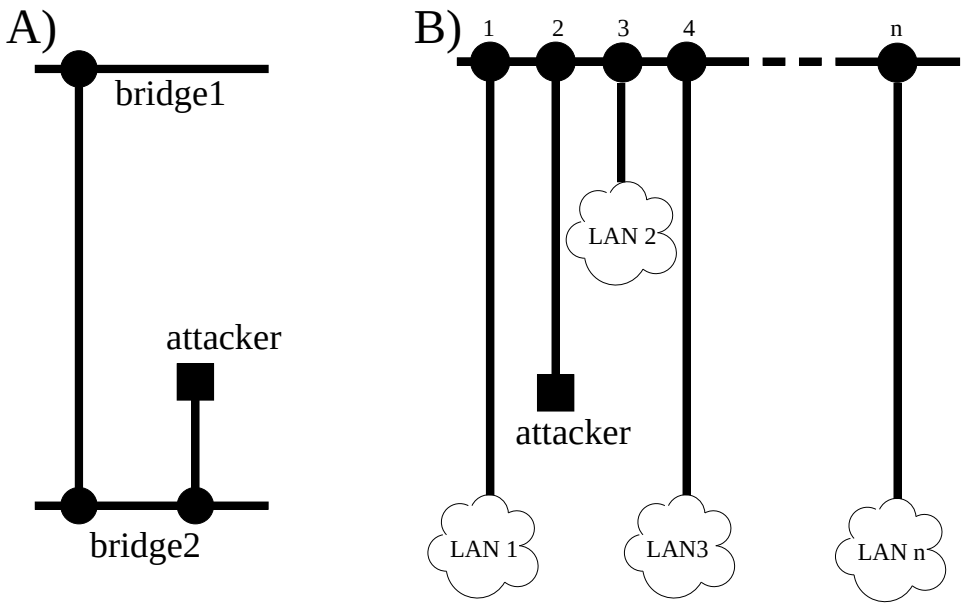

Рис. 8: Частичный DoS (пример 2)

единения, а именно, сервер. Для этого он должен убедить коммутатор, к которому подключен концентратор с интересующим его клиентом, что он имеет лучший путь до второго коммутатора, к которому подключен сервер. В результате, линк между коммутаторами рвется, и атакующий может прикидываться сервером или просто злорадствовать от факта недоступности сервера клиенту (см. рис. 9). Разумеется, концентратор здесь не принципиален, хотя можно строить атаку, подобную этой, на всю сеть, не задумываясь над подсчетом величин для данного сегмента сети. Если конечной целью является компьютер, подключенный к концентратору - он не «упадет» от этой атаки, ибо слишком «глуп».

Рисунок 8, часть a), обращает внимание на следующую особенность: если считать, что нумерация портов идет слева направо, то атакующий, пытаясь устроить частичный DoS для линка между коммутаторами, может сделать его исключительно себе, если не подобрать параметры специальным образом. Дело в том, что, при прочих равных условиях, включенным останется порт с меньшим ID. Поэтому атакующий должен подобрать для фальсификации такой номер порта соседнего коммутатора, который заставит коммутатор «Bridge 2» выключить именно необходимый атакующему линк, а не тот, с которого идет атака. Для случая же с одним единственным STP-совместимым устройством (рис. 8, часть b)) атака легко реализуема для всех LAN, подключенных к портам, боль-

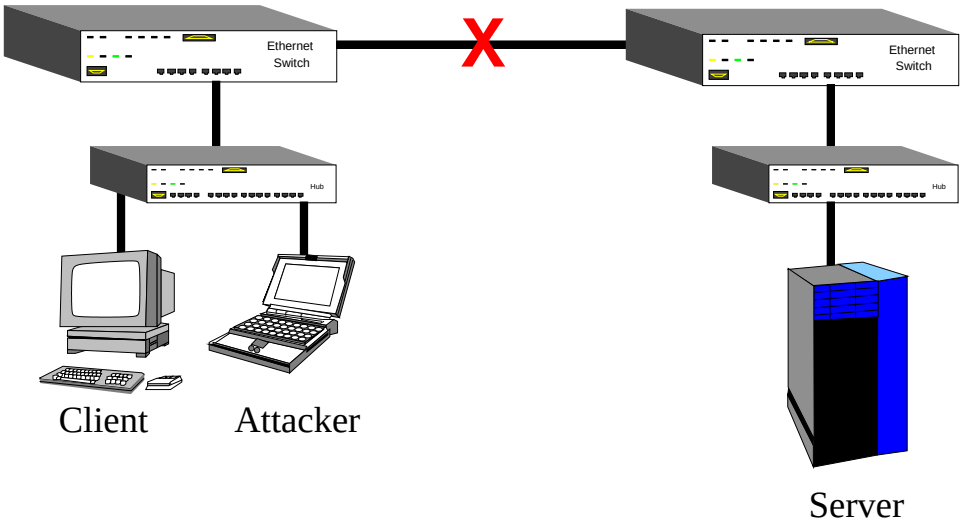

Рис. 9: Частичный DoS (пример 3)

шим по номеру. Однако при попытке реализовать ее по отношению к первому порту ничего не выйдет: получив BPDU на большем по номеру порту от имени порта с меньшим номером, устройство выключит порт с большим номером.

Изменяя Max Age, forward delay, hello time в ложных c-bpdu, мы можем управлять как минимум ближайшими коммутаторами, изменяя их представление о топологии сети, а также (что принципиально при организации частичной DoS) - изменять тайм-ауты протокола, что позволяет нам выставить конфликтующие с остальной сетью значения.

Следует заметить, что в силу таймаутов реконфигурации в рамках протокола и того, что в момент реконфигурации пользовательские фреймы не передаются, применение данной методики для *временного* получения пакетов (например, части пакетов используемых при установлении соединения с помощью протоколов верхних уровней, например, IP) нецелесообразно, поскольку в момента изменения топологии эти пакеты блокируются[20].

Предметом атаки может быть маршрут между любыми двумя STP-совместимыми устройствами сети. Следует также отметить, что при организации части описываемых здесь атак придется угадывать bridge ID

[20] Тем не менее, при определенных настройках с использованием расширенной спецификации STP-протокола это все-таки имеет смысл (см. раздел 8.1)

части устройств, однако эта задача зачастую облегчается возможностью получить часть идентификатора или диапазон возможных значений косвенным образом - через протоколы более высокого уровня (например, используя стек TCP/IP, можно получить MAC коммутатора, пропинговав его и посмотрев arp таблицу), используя таблицы зарезервированных за тем или иным производителем диапазонов MAC адресов и др. Имея один из MAC адресов многопортового активного устройства, легко вычислить остальные, поскольку производители обычно присваивают MAC адреса портам подряд, а их изменение, даже если оно доступно администратору, практически не требуется для типичных задач администрирования. Важно заметить, что реализация частичной DoS не требует использования перевыборов STP-root - достаточно перевыборов designated bridge, что, как следствие, позволяет, при определенных условиях, провести эту атаку незаметно для всей остальной сети[21].

6.4.6 STP DoS: Величины, которые должны быть установлены в STP пакетах, организующих DoS

Глава 8.6 (конкретнее, раздел 8.6.1.3) в [1] содержит подробное описание всех параметров BDPU. Разумеется, в зависимости от алгоритма атаки величины используются разные. Для DoS атаки посредством постоянной инициации выборов STP root bridge важны следующие параметры.

Root bridge устанваливает следующие значения:

$$RootPathCost = 0 \quad ([1],\ 8.5.3.2)$$
$$RootPort = 0 \quad ([1],\ 8.5.3.3)$$
$$Bridge's rootport = 0 \quad ([1],\ 8.8.1.a)$$

Message Age надо устанавливать минимально возможным, так, чтобы пакет считался молодым. В любом случае этот параметр должен быть намного меньше Max-age. Впрочем установка этого параметра также зависит от выбранного алгоритма.

[21] Данная атака может оказаться нереализуемой без физического подключения к атакуемому сегменту. Авторы планируют уточнить это дополнительными экспериментами. После обсуждения с Dmitry Goloubev (адрес см. в разделе 19) опубликованной авторами статьи, являющейся выжимкой данной работы, у нас появились сомнения в порядке работы STP-совместимых устройсв с BPDU. Существует две возможности: 1) коммутатор изменяет состояние порта в случае прихода BPDU на любой порт и 2) коммутатор изменяет состояние порта только по приходе BPDU именно на него. Разумеется во втором случае часть описаных атак сужает область своего применения *(но только часть!)*.

Как и Message Age, Max Age надо устанавливать в зависимости от выбранного алгоритма. Так, для атак, в которых важно максимально долго сохранить навязанную структуру STP дерева, Max Age должен быть установлен в максимум, 40 секунд (см. [1], стр. 108, таблица 8.3), а для тех случаев, когда необходимо инициировать как можно больше перевыборов - минимальным, т.е. 6 секунд.

Bridge ID должен устанавливаться в соответствии с алгоритмом из раздела 6. Выборы выигрывают меньшие значения.

Port ID - в определенных случаях участвует в выборах, поэтому должно быть выбрано значение с максимальным приоритетом - 0.

6.4.7 STP DoS: Наиболее эффективные величины пауз при организации отказа в обслуживании

Max Age ([1], 8.5.1.6, стр. 68; [1], 8.5.3.4, стр. 69) - должен быть максимально возможным.

Forward Delay ([1], 8.5.1.8) - должен быть максимально возможным (чтобы максимально затормозить переключение в forwarding state). Время которое порт не передает фреймы $t = 2 * fowarddelay$ (определяется [1], 8.7.5, шаг 2)).

Hello Time ([1], 8.5.3.5) - должен быть минимальным.

Bridge ID - рассматривается в другой главе.

На страницах 108 и 109 в [1] приведены таблицы значений, в том числе значений таймеров. Для максимально эффективной DoS нужно, чтобы

$$2 * Forward\_Delay > Max\_Age.$$

Выбирая разрешенные стандартом значения из таблицы 8-3 из [1], 8.10.2, стр.107 видим, что неравенство

$$2 * 30 > 40$$

выполняется с хорошим запасом[22].

[22] В сложных сетях эти параметры могут иметь другие значения. Это связано с тем, что в некоторых случаях администратору необходимо подбирать величины параметров под размеры сети для обеспечения сходимости.

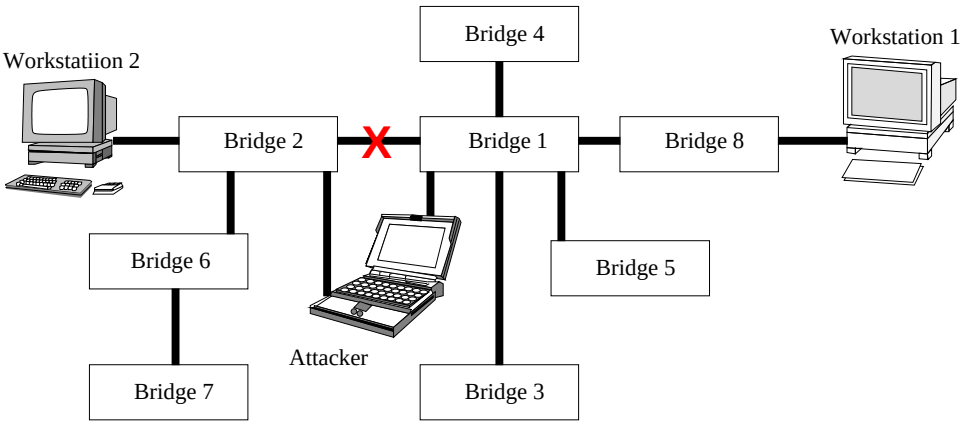

Рис. 10: Человек посередине

Определенные значения из таблиц интересны только при отдельных типах атаки, например, path cost имеет смысл устанавливать только при нежелании оспаривать статус «designated root».

6.5 Навязывание ложного маршрута (человек посередине), или «перетяни кольцо на себя».

Как уже упоминалось, в силу того, что топология сети определяется при участии протокола STP, возможна атака по схеме «ложный объект PBC с навязыванием ложного пути» (в терминологии [3]), известная по англоязычным публикациям как MitM (Man-in-the-Middle). Однако реализовать такую атаку можно только при определенных условиях конфигурации сети, а именно в тех случаях, когда жертвы, трафик между которыми надо перехватить (то есть, направить по каналу, который можно прослушивать), находятся в сети с двумя (как минимум) STP-совместимыми коммутаторами и подключены к разным коммутаторам. Причем, для реализации атаки в полной мере требуется два сетевых интерфейса, подключение которых необходимо осуществить так, чтобы образовать потенциальный дубликат имеющегося пути между источниками.

Рассмотрим рис. 10. На нем изображена сложная сеть, построенная на интеллектуальных коммутаторах (bridge), поддерживающих протокол STP. Сеть показана в «устоявшемся» состоянии, т.е. формирование топологии завершено, резервные линки переведены в состояние «blocking». В задачу злоумышленника входит осуществить перехват трафика между устройствами, подключенными к коммутаторам Bridge2 и Bridge8 (на рисунке изображены как Workstation1 и Workstation2). Результатом

атаки будет перевод имеющихся соединений между Bridge1 и Bridge2 в состояние «blocking» (на рисунке отмечено крестиком) и перенаправление всего трафика между этими коммутаторами (а значит, и трафика между обеими точками, траффик которых атакующему необходимо подслушать) на интерфейсы атакующего. В принципе, задача определения мест включения в сеть, при которых будет возможна атака STP-MitM, должна поддаваться описанию через теорию графов. Для того, чтобы получить именно MitM, а не DoS, атакующий хост должен поддерживать работу обоих своих интерфейсов в режиме моста, что легко реализуемо, поскольку существует несколько свободно распространяемых (в том числе по GNU GPL лицензии) реализаций bridging'а, например, для Linux.

Заметим, что для этого даже не требуется поддержка STP ОС, работающей как мост. Более того, такая полнофункциональная поддержка только повредит, поскольку принцип MitM атаки при использовании двух сетевых интерфейсов базируется на том, что атакующий может посылать BPDU пакеты от чужого имени (например, в Bridge1 от имени Bridge2) чтобы представить себя как наилучший путь хождения пакетов. При такой логике атаки полноценная поддержка STP только помешает атакующему, поэтому для реализации MitM по схеме на рис. 10 необходимо отключать поддержку Spanning Tree, либо использовать OS+software, в которых нет поддержки STP, например, реализовать bridge на OpenBSD, либо модифицировать исходный код поддержки STP в данной OS. Таким образом, машина, реализующая MitM, на самом деле не нуждается в полноценной поддержке bridging'а, а может функционировать на манер концентратора, с единственным исключением - транзитный STP-траффик должен отбрасываться и вместо него генерироваться собственный, анонсирующий себя как наиболее выгодный путь для пакетов.

Хочется акцентировать внимание читателя на том, что на самом деле эта атака более общего свойства и предметом атаки может быть маршрут между любыми двумя STP-совместимыми устройствами сети (см. раздел 6.4.5).

Простота этого протокола делает возможной достаточно дешевую реализацию описываемой атаки «в железе» - за менее чем 5-10 тысяч долларов (в основном на оплату интеллектуального труда разработчиков) за срок порядка полугода возможно построить достаточно универсальный, если требуется и программируемый, «деструктор» сети или же реализо-

вать «в железе» вариант MitM-атаки, описываемой в этом разделе[23]. К сожалению такие разработки могут заинтересовать разве что военных, а тратить на разработку экспериментального образца собственные деньги авторы не считают возможным[24]. Требование наличия как минимум двух коммутаторов связано с тем, что в случае подключения к одному коммутатору анонсирование себя, как лучшего пути создаст тупиковый путь, то есть это случай, относящийся к разделу 6.4.5. К тому же для этого случая гораздо проще воспользоваться arp-poisoning.

Важно заметить, что все так однозначно и просто только в случае, если атакующий подключен к соседним коммутаторам. В случае же, если он подключен к коммутаторам непосредственно не соединенным, придется подбирать значения root id, возможно последовательным перебором, поскольку bpdu не ходят (в неизменном виде) дальше первого встреченного STP-совместимого устройства. Поэтому такая атака кажется не слишком часто применимой. Помимо этого у данной атаки есть еще одно неудобство - в случае реализации данной атаки на обычной персоналке, сразу же упадет пропускная способность сети между точками, соединенными через интерфейсы атакующего, за счет переноса соединения на более медленные сетевые интерфейсы ПК. В принципе, злоумышленнику не обязательно иметь подключение к коммутаторам являющимся частью разорванного STP-кольца - он может «растянуть» кольцо на себя, если в сети есть другие дублируемые соединения. Просто в случае подключения к коммутаторам, образующим кольцо, задачу организации MitM решить легче.

Описанная атака работает не на всех конфигурациях. Однако в случае, когда злоумышленник захватил контроль над компьютером, который по какой-либо причине подключен к двум или более коммутаторам различными интерфейсами - он имеет возможность ее реализовать.

6.5.1 Заметки по конструированию пакетов для реализации атаки Man in the Middle

В общем случае величины используемые при конструировании пакетов, должны зависеть от параметров, полученных от устройств, в которые включен атакующий, либо могут подбираться, что может вызвать частичный/временный DoS.

[23]Причем, не только интеллектуальной сети - «глупые» сети, построенные на концентраторах, можно вывести из строя, заставив NIC постоянно передавать сигнал jam, можно совместить по очереди jam'ing и описываемый здесь STP-redirect.

[24]Более того, об этом есть смысл говорить только применительно к портативному устройству с автономным питанием, которое можно незаметно пронести в здание, т.к. специалист сможет построить такой «деструктор» на базе старенькой ПЭВМ с двумя сетевыми платами, потратив на оборудование менее 100 долларов.

Отправной точкой к значениям, используемым в этой атаке, являются величины, используемые при выборах и анонсируемые точками подключения MitM-члена сети. В частном случае, чтобы выиграть выборы designated root для данного сегмента ЛВС, будет достаточно подстановки меньшего значения PathCost (при прочих равных условиях). Однозначно не стоит трогать глобальные параметры, поскольку они чреваты DoS всей сети.

6.6 Провокационный сниффинг

Провокационный сниффинг получил свое название по аналогии с провокационным спуфингом. На момент написания статьи авторам не встречалось описания аналогичных методик в области сниффинга. Идея провокационного сниффинга возникает после внимательного прочтения стандарта [1], а точнее той его части, которая описывает поведение STP-совместимого устройства в случае изменения STP-дерева (например, сразу после перевыборов Designated Bridge). Дело в том, что после такого события, согласно стандарту, коммутатор должен обнулить свою таблицу коммутации (кроме статически заданных администратором значений), а в случае нулевой таблицы коммутации коммутатор (первое время, до того, как запомнит на каком порту какой MAC) начинает вести себя как концентратор.

6.6.1 Что такое провокационный сниффинг

Провокационный сниффинг – это алгоритм атаки, который может быть использован при определенных настройках активного сетевого оборудования для «отупления» этого оборудования с целью подслушивания трафика, проходящего через его порты. Грубо говоря посредством провокационного сниффинга коммутатор превращается в концентратор посредством «провокации» - фальсификации события «произошло изменение топологии сети», после которого все динамические записи в таблице коммутации очищаются. Однако использование такой провокации с толком возможно только при одном условии: коммутатор не должен успеть обучиться. Под «обучением» в данном случае понимается формирование таблицы соответствия портов и MAC адресов подключенных к ним устройств. Поскольку обучение коммутатора происходит практически сразу же по получении им пакета (фрейма) от подключенного устройства, реализация полноценного сниффинга становится нетривиальной задачей, сложность которой пропорциональна скорости трафика в данной сети. Отсюда следует, что существуют условия при которых эта атака в принципе не реализуема в полном объеме. Например, если трафик между станциями, обмен которых необходимо «подслушать» (станциями «жертвами»), использует более 49% полосы пропускания. Для 100% эф-

фективности данная атака предполагает бомбардировку «отупляемого» коммутатора STP пакетами хотя бы вдвое чаще, чем проходят пакеты между жертвами, для того, чтобы каждый новый принятый пакет приходил на коммутатор с только что очищенной таблицей коммутации. С учетом того, что от порта коммутатора начинается коллизионный домен, полноценный сниффинг кажется еще более затрудненным, что, правда, не касается full-duplex портов. Для полноценного сниффинга без потери пакетов за счет обучения коммутатора суммарный траффик, проходящий через коммутатор в единицу времени, не должен превышать 48-49% пропускной способности канала атакующего.

Такая атака в нормальных условиях, следуя букве стандарта, не реализуема, поскольку сброс таблицы коммутации не осуществляется мгновенно по приходу C-BPDU с отличной от текущей конфигурацией. Однако многие устройства поддерживают расширения стандарта, в которые заложена возможность мгновенного перехода из blocking в forwarding. Как следствие, время жизни таблицы коммутации в таких устройствах в случае возникновения изменений топологии можно сбросить в 0 (см. стр.49). В этом случае, необходимо подобрать такие условия работы устройства, чтобы оно, не прерывая работы (т.е. без DoS), тем не менее постоянно сбрасывало свою таблицу коммутации.

Как и MitM-атака, атака provocation sniffing может быть использована, например, для получения доступа к управлению STP-совместимым устройством. Многие устройства, работающие преимущественно на 2-м уровне модели OSI, не поддерживают терминального доступа иначе, чем при помощи telnet, а пароли к сессии, открытой по этому протоколу передаются, как известно, в открытом виде. Помимо этого, атаки на втором уровне OSI могут служить подготовкой к атакам на более высоких уровнях. Например, имея возможность «слушать» трафик, можно эффективно угадывать sequence numbers для «вклинивания» в tcp-соединения.

6.6.2 Использование STP-выборов для реализации провокационного сниффинга

Основная сложность в случае стандартной поддержки STP заключается в том, что необходимо выбрать такой алгоритм, при котором постоянно происходят перевыборы чего либо «неважного». В такой ситуации помочь могут перевыборы, например, designated bridge или designated port для данного сегмента сети, что позволяет не вызвать DoS для сети и, в частности, для сегментов ЛВС, подключенных к данному коммутатору. В случае если при проведении STP-перевыборов в одном из сегментов остальные продолжают работать – это удача.

Задача атакующего облегчается, если коммутатор поддерживает расширенную спецификацию STP, а именно аналог cisco «spanning-tree portfast». В этом случае, по крайней мере, не будет пауз в процессе передачи фреймов «Frame Relaying» (см. определение Relayng Entity в [1] 7.2.2) даже при затрагивании более существенных параметров, чем designated port или bridge. Помимо этого условия важно избрать для оспаривания такую выборную величину, выборы которой не вызовут перехода в состояние Blocked порта, с которого производится атака. В процессе перевыборов можно играть практически всеми выборными параметрами STP, но так, чтобы не занять статус designated root, поскольку, если коммутаторов более двух, то, проводя перевыборы корня, мы автоматически затрагиваем все коммутаторы, а это может вызвать DoS, если на остальных коммутаторах не включена поддержка «spanning-tree portfast». В простейшем случае, если у нас есть всего один свитч, на котором установлено «STP portfast», мы можем смело устраивать перевыборы корня дерева по наиболее агрессивному из DoS алгоритмов, причем стараясь послать как можно больше пакетов, поскольку перевыборы должны стартовать чаще прихода пакетов от клиента, которого надо «прослушать», на коммутаторе. Следует также иметь ввиду, что в следствие того, что STP BPDU не проходят сквозь STP-совместимые устройства, а лишь вызывают посылку собственных, невозможно будет добиться полного «охабливания» всех коммутаторов, а не только ближайшего. В отличие от атакующей машины, коммутатор связан стандартами и, скорее всего, не будет посылать собственные BPDU чаще, чем через минимально возможный таймаут, определенный стандартом, поэтому часть коммутаторов успеет за время паузы «обучиться», пусть и частично. Впрочем, можно чередовать эту атаку с атаками provocation spooffing и Man in the Middle, что позволит при грамотном подборе параметров получить больше информации, чем при чистой «provocation sniffing» атаке.

В процессе изучения стандарта мы обратили внимание на то, что возможна также реализация provocation sniffing исключительно на C-BPDU, не вызывающих перевыборы. Такую атаку (без перевыборов) можно реализовать, используя особенность, определенную в [1] 8.3.5, «Notifying topology changes», в части, описывающей поведение моста по получении C-BPDU с установленным «topology change flag». Здесь имеет смысл вспомнить как работает сам алгоритм ST: когда топология сети меняется, мосты нуждаются в информации об этом чтобы переобучиться. Единственный доступный механизм быстрого переобучения - сброс информации о MAC адресах в таблице коммутации и ее повторное построение (Learning). Стандарт определяет следующую последовательность действий: обнаружив изменения в сети, некий мост передает TCN-BPDU в сторо-

ну designated root устройства до тех пор, пока не получит подтверждения от «designated bridge» для этой сети посредством C-BPDU. Соответствующий TCN-BPDU генерируется designated bridge этой сети и передается дальше в сторону корневого устройства. Когда корневое устройство (designated root) получает TCN-BPDU, генерация им C-BPDU изменяется, и в течении времени определяемого Topology Change Timer ([1], 8.5.3.13), они содержат флаг «topology change» установленным. По получении C-BPDU с установленным флагом topology change, мост обязан временно сбросить время жизни MAC адресов в таблице коммутации на минимум, равный значению Forward Delay. В случае использования расширений STP, позволяющих сразу перейти из состояния blocking в Forwarding и наоборот, параметр Forward Delay, очевидно, равен 0, что, при наличии C-BPDU flood заставит коммутатор вести себя как концентратор, если конечно он не переконфигурирован специальным образом. Данная атака не действенна в случае, если MAC-адреса станцийы прописан на коммутаторе администратором, поскольку, согласно стандарту [1] (глава 8.3.5), Aging Time можно сбросить только для записей, набираемых динамически в процессе самообучения коммутатора. Впрочем, и здесь есть ограничения - посылка C-BPDU вызовет хотя бы разовые выборы того или иного параметра, однако, в этом случае, если возможности послать C-BPDU от имени «верхнего к данному» коммутатора (обязательно через тот же порт) нет, то DoS для части сети практически неизбежен. Этого можно избежать, если образовать дополнительную петлю и скомбинировать эту атаку с методом, описанным в разделе 6.5, но уже во избежание DoS, чтобы заменить собой designated bridge для этой сети.

Для того, чтобы иметь возможность произвести атаку без последовательных перевыборов, надо подделать «keep-alive» C-BPDU, которые распространяет designated bridge этого сегмента сети (то есть тот, который ближе к designated root, чем атакуемый), что возможно без DoS не во всех случаях. Кстати, согласно той же главе [1] восприимчивость к topology changes может быть выключена администратором. В то же время, согласно стандарту [1] (8.5.5.10), реакция на TCN-BPDU и C-BPDU с установленным флагом topology change должна быть включена и возможность выключения этой способности не обязательна.

Для полноты картины имеет смысл прочитать главы 8.6.14 и 8.6.15 из [1].

Кроме того, на возможность реализации данной атаки может влиять поддержка свитчем и включение администратором таких дополнений к IEEE STP, как аналоги Cisco «BPDU Guard» и «BPDU root guard». Об этом будет рассказано ниже в разделе 8.

6.6.3 Сравнение arp-poisoning и provocation sniffing

Существенной разницей между arp-poisoning и провокационным снифингом является то, что arp-poisoning позволяет перехватывать трафик только между определенными узлами, работающими по IP, provocation sniffing же позволяет перевести коммутаторы в режим концентратора, в результате можно перехватить *весь* трафик, *всех* узлов[25], по *всем* протоколам (включая ipx, netbeui и т.п.), как 3го и выше, так и второго уровня OSI.

6.6.4 Комментарии к написанию кода для provocation sniffing

Прежде всего, чтобы коммутатор не успевал обучиться «надолго» его нужно постоянно бомбардировать C-BPDU с установленным topology change flag, по приходу которых Max Age динамических записей в таблице коммутации становится минимально возможным (около 4 секунд). Разумеется, это полезно для атакующего, поскольку часть пакетов в результате будет распространена во все порты из-за устаревания информации о нахождении MAC-адреса, которому данный пакет предназначен. И это без всяких перевыборов и при выключенном «spanning-tree portfast»![26] Следует помнить, что «перевыбирать» можно только те параметры, которые не заденут функциональность собственного порта, иначе можно легко организовать DoS самому себе. В качестве одного из возможных вариантов можно привести использование выборов designated port для K из N ($K < N$) сегментов сети, подключенных к атакуемому коммутатору. Почему не одного? Можно и одного, но если задействовать несколько сегментов, то можно чередовать выборы, устраивая их по очереди, чтобы процесс перевыборов происходил как можно чаще. Следующим после перебора designated port некоторого сегмента ЛВС, будет перебор designated bridge. Однако есть еще лучший путь - диапазон значений для некоторого порта содержит также определенный диапазон для изменения приоритета, причем не такой уж и маленький, таким образом можно свести побочный эффект атаки к DoS всего лишь одного порта, а при неполном заполнении портов - обойтись без DoS, устраивая перевыборы designated port для незанятого ничем порта. В случае если поддерживается аналог «spanning-tree portfast» (Cisco), мы имеем самый простой случай - можно атаковать коммутатор, используя как C-BPDU, так и TCN-BPDU, что даст почти одинаковый эффект. Однако TCN-BPDU предпочтительнее, так как не вызывают DoS.

[25] Практически только в пределах коммутаторов, к которым атакующий имеет прямое подключение

[26] Здесь не стоит забывать, что практически сведение aging time таблицы коммутации устройства-жертвы к 4 секундам будет давать крохи.

7 Получение дополнительной информации о сети при помощи STP

Помимо описанных выше фундаментальных атак на топологию сети существует еще одна - не страшная, но тоже неприятная особенность: благодаря STP возможно получение дополнительных данных о структуре сети. При этом такое получение дополнительных знаний может быть как пассивным (прослушивание среды передачи данных), так и активным - выиграв на короткое время статус STP корня мы автоматически становимся целью отправки STP TCN-BPDU (см. раздел 1), которые сами по себе несут некоторый объем познавательной информации. Например, проанализировав source MAC адрес, можно понять устройство какой фирмы отправило этот пакет, а это тоже атака, хоть и пассивная (см. [3]).

8 Особенности реализации Spaning Tree в некоторых устройствах различных производителей

Здесь и далее уделяется внимание поддержке LAG'ов - Link Aggregation Group - одного из вариантов, который, в некоторых случаях, может стать приемлемой заменой STP.

8.1 Cisco

Довольно «продвинутые» возможности настройки. Часть может быть использована для защиты от DoS атак. Поддерживается фильтрация STP для эмуляции раздельного STP-дерева на каждый vlan. Эта фирма не зря занимает лидирующие позиции на мировом рынке, однако это не мешает следующим недостаткам: STP во всех устройствах его поддерживающих по умолчанию включен. Имеющиеся наработки на самом деле, скорее всего, не предназначались для защиты от атак. Нет возможности выключить STP на отдельном порту.

Очевидным необходимым правилом для устойчивой к DoS настройке каждого порта будет следование следующему примеру (взято из конфигурации cisco 2924XL):

```
------------cut-------------
interface FastEthernet0/2
description Any Port Sample
spanning-tree portfast
------------cut-------------
```

Также заслуживают внимания следующие возможности IOS:

BPDU Guard
В общих чертах эта «фича» позволяет отключить порт, когда на нем появляется устройство, мнящее себя designated root. Предназначено для запрета изменения конфигурации STP Tree без ведома администратора. Спасает только от атаки, включающей фазу перевыборов designated root. Перевыборы же Designated Bridge для некоторого сегмента LAN не отслеживает (в общем случае сложно отличить атаку от нормальной работы, иногда невозможно, см. раздел 15.1). В рамках тематики данной статьи, как и spanning-tree portfast, считаем очень полезной. ⊗

Поддерживаются LAGи.

8.2 Avaya (бывш. Lucent)

В серии коммутаторов cajun (cajun P120, P330(R), P220, P550(R), P880) заявлена поддержка 802.1d, причем на моделях P550/P880 заявлена поддержка ST per VLAN, при этом в учебнике курса «Cajun Campus Training» отдельно отмечено, что поддержка ST per vlan не совместима с реализацией ST у других производителей. Интересно чем? :) Однако далее пишется, что не смотря на эту «несовместимость» P550/P880 все-таки работают с 802.1d и 802.1q устройствами других производителей. :) Чудеса в решете. ;)

Впрочем, поддержку STP в Avaya стоит рассмотреть отдельно. Наиболее слабые модели (P120 и ей подобные) поддерживают STP только в рамках спецификации 802.1d, более «продвинутые» (P330 и выше) поддерживают не только спецификацию STP из 802.1d, но и из 802.1q и, самые продвинутые (серии P580 & P882) поддерживают еще одну реализацию Spaning Tree, разработанную avaya, и имеющуюся только на их оборудовании - «Spaning Tree Dual Layer». Грубо говоря, это STP 802.1q + STP 802.1d, одновременно связанные так, что STP-деревья, работающие по алгоритму 802.1d терминируются на тех портах, к которым подключены устройства, не поддерживающие 802.1q. Вообще, у avaya очень большое внимание уделяется поддержке гетерогенных, с точки зрения наличия оборудования разных поставщикови просто возможностей тех или иных «железок», сетей, особенно в моделях 580/582. Собственно Spaning Tree Dual Layer предназначена именно для случая, когда часть коммутаторов не поддерживает STP per VLAN, а выбрасывать их не с руки. Spaning Tree Dual Layer организовано таким образом, что внутренности P580/P882 отслеживают возможные проблемы с STP, работающем по обоим стандартам. Хотя STP Dual Layer является закрытым алгоритмом, который до сих пор (Nov 29 2001) не опубликован avaya,

это не освобождает его от всех принципиальных уязвимостей, свойственных STP алгоритму. Наоборот, в силу усложнения протокола без отхода от его принципов, количество потенциальных уязвимостей может быть больше. Также важно, что, начиная с версии ПО 3.2.8, функционирование STP можно отключить на любом порту. Это очень удобно: если вам необходимо предоставлять подключение клиенту поверх LAN, и вы уверены, что STP-колец образоваться не может, то вы можете выключить поддержку STP на границах вашей части LAN и не боятся каких-либо STP-атак извне, по-прежнему пользуясь преимуществами STP внутри доверенной среды своей организации. В общем, не смотря на то, что, например, настроек вроде cisco «bpdu guard» у них нет, Avaya производит в этом отношении довольно приятное впечатление.

Поддерживаются LAGи. ⊗

8.3 Intel

Поскольку, в отличие от cisco и avaya, продукция других компаний знакома авторам, в основном, по документации, далее будет идти речь о конкретных моделях (что не отменяет ни одного слова, сказанного о глобальной уязвимости протокола и не делает устройства других компаний менее уязвимыми).

Модель 460T поддерживает STP, однако не поддерживает STP-фильтров для эмуляции отдельного ST-дерева на vlan, что является очевидным минусом и упрощает STP-атаки, не говоря уже просто о неудобствах для обыденной работы администратора сети.

Поддерживаются LAGи.

8.4 HP

Модели HP212M/214M приятно удивили своей резистивностью к STP атакам в конфигурации с настройками по умолчанию - STP disabled. Но на этом приятные сюрпризы закончились - по факту включения STP, коммутатор становится точно так же уязвим, как и все остальные. К тому же он не поддерживает VLAN. Впрочем, низкая стоимость этих моделей подразумевает это[27].

8.5 3Com

STP поддерживают многие, если не все, интеллектуальные модели.

[27] Если я не ошибаюсь, в этой модели HP есть и еще одна весьма полезная функция – возможность выключить STP per port.

Далее речь о коммутаторах 3Com SuperStack и им подобных. Ниже приводится выжимка из документации, доступной по url 5 в разделе 20.1 на стр. 74

Коммутатоph имеет полностью раздельные системы STP для каждого специфицированного вами VLAN. Каждый VLAN имеет свои Root Bridge, Root Ports и BPDU[28]. Помимо этого в 3COM используется резервирование определенных VLAN для нужд управления коммутатором, что накладывает определенные ограничения на функционирование STP в VLAN с определенным номером. Вот перевод одного из абзацев в документации:

> Вы не можете использовать VLAN 16 с STP. Также, если вы используете режим AutoSelect VLAN, вы не можете использовать VLAN 15. В этом случае эти VLAN используются для внутренних нужд коммутатора и недоступны.

В числе параметров STP в этих моделях 3COM доступны для просмотра следующие:

Topology Changes - показывает, сколько раз происходило изменение топологии в текущем vlan.

Max Age (6..40) - время (в секундах), которое коммутатор ждет перед началом реконфигурации сети. Если он не получает BPDU в течение этого времени, он пытается начать реконфигурацию.

Designated Root - Bridge Identifier designated Root Bridge.

Hello TIme (1..10) - интервал (в секундах) между передачами BPDU коммутатором.

Root Cost - стоимость пути от свитча до Root Bridge.

Forward Delay (4..30) - время (в секундах), которое порты коммутатора проводят в режимах Listening и Learning (см. подраздел 8.5.1).

Root Port - read only - идентификатор Root Port этого коммутатора (когда он не является designated root).

Hold Time - Минимально-допустимый временной интервал (в секундах) между передачами BPDU.

[28]Как показано ранее, производители очень часто выдают желаемое за действительное - поддержка полностью отдельного дерева STP на каждом VLAN - миф.

Time Since Topology Change - read only. Время с момента последней смены топологии.

Bridge Priority (0..65535). Это поле позволяет установить приоритет коммутатора. Меняя это значение, вы меняете шанс коммутатора выиграть выборы Root Bridge. Уменьшение величины увеличивает шанс. Значение по умолчанию 65535.

Bridge Max Age (6..40). Позволяет установить время (в секундах), которое коммутатор ждет, перед тем как начать реконфигурацию, если он является Root Bridge. Если он не получит BPDU в течение этого времени, он попытается изменить топологию STP. По умолчанию - 20 секунд. Это время должно быть больше или равно 2*(Hello Time + 1) и меньше или равно 2*(Forward Delay - 1).

Bridge Hello Time (1..10). Позволяет задать временную задержку (в секундах) между передачами BPDU этим коммутатором, когда он является Root Bridge.

Bridge Forward Delay (4..30). Позволяет задать время (в секундах), которое порты коммутатора проводят в режимах Listening и Learning, когда коммутатор является Root Bridge. По умолчанию 15 секунд (см. подраздел 8.5.1).

8.5.1 Установка параметров STP для порта

Состояния портов (read only):

Disabled - порт в этом состоянии не пересылает пакеты и не участвует в stp.

Listening - порт в этом состоянии готовится к пересылке пакетов, но временно заблокирован, чтобы не создавать петель. В этом состоянии BPDU передаются, принимаются и обрабатываются.

Blocking - в этом состоянии порт не передает пакеты, чтобы предотвратить появление дублирующих путей. Порт учитывается в вычислениях STP, BPDU могут передаваться, приниматься и обрабатываться.

Learning - порт в этом состоянии готовится к пересылке пакетов, но временно заблокирован, чтобы не создавать петель. В этом режиме порт изучает адреса всех не содержащих ошибок пакетов. Порт учитывается в вычислениях STP, BPDU могут передаваться, приниматься и обрабатываться.

Forwarding - порт передает пакеты. BPDU также могут приниматься и обрабатываться.

Designated Port - read only. Показывает Bridge Identifier Designated Bridge Port для портов текущего сегмента.

Designated Root - read only. Идентификатор Root Bridge.

Designated Cost - read only. Стоимость пути от Root Bridge до Designated Bridge Port для портов данного сегмента.

Designated Bridge - read only. Показывает Bridge Identifier Designated Bridge для портов данного сегмента.

Fwd Transition - read only. Сколько раз этот порт переходил из Learning в Forwarding.

Port Enable (Enable/Disable) - позволяет запретить или разрешить данный порт.

Priority (0..255). Позволяет задать приоритет порта. Изменяя значение, вы изменяете шанс этого порта стать Root Port. Меньшее значение увеличивает шанс. По умолчанию - 128.

Path Cost (0..65535). Позволяет задать path cost для порта. Коммутатор автоматически назначает значения path cost по приведенной ниже таблице. Если вы задаете другое значение, автоматическое назначение отключается и вернуть это обратно можно только переинициализацией коммутатора.

Port type	**Duplex**	**Cost**
100BASE-TX/100BASE-FX (VLT)	Full	5
	Half	12
10BASE-T (VLT)	Full	24
	Half	25
100BASE-TX/100BASE-FX	Full	150
	Half	300
10BASE-T	Full	650
	Half	700

Fast Start (Enable/Disable) Это поле задает, переходит ли порт сразу в состояние Forwarding, когда к нему присоединяется устройство. Ставьте Enabled, если к порту непосредственно присоединена конечная станция. По умолчанию - Enable.

ВНИМАНИЕ! Если вы поставите порту Fast Start Enable, когда к нему присоединено несколько конечных станций, в сети могут появиться петли.

Как видно из этих выжимок из документации обсуждаемая модель 3COM поддерживает аналог cisco «spanning-tree portfast», что важно в контексте этой статьи.

9 Некоторые замечания по поводу Linux bridging project

Ядро linux имеет поддержку бриджинга (см. раздел 20.2). На сайте Linux bridging project есть утилиты для управления параметрами бриджа. В качестве одного из вариантов конструирования атак, обсуждаемых в этой статье, можно просто сделать скриптик, который будет вызывать эти утилиты с параметрами «понизить bridge id» (т.е. стать designated root) и периодически менять различные параметры, например max age и т.п.

10 Некоторые замечания по поводу GARP и GVRP

GARP - Generic Attribute Registration Protocol. Протокол используемый для регистрации некоторых абстрактных свойств в рамках сети. Чем так интересен GARP? Тем, что STP это частный случай GARP, а значит сам по себе GARP стоит рассмотреть на предмет потенциальных дыр. GVRP – Generic VLAN Registration Protocol – протокол регистрации информации о VLAN. Позволяет разделять информацию о VLAN между мостами и производить администрирование распространения VLAN. Чем так интересен GVRP? Также, как и STP, является частным случаем GARP, а значит его стоит проверить на предмет потенциальных дыр. По крайней мере полное отсутствие механизмов обеспечения безопасности в STP наводит на мысль, что и с GARP вообще и с GVRP в частности ситуация будет весьма похожей. =) ⊗

11 Краткий обзор известных атак на втором уровне OSI, а также заметки о возможных атаках на втором уровне OSI

В этом разделе сделаны краткие заметки о других возможных типах атак на 2-м уровне OSI.

- jam distribution - DoS атака в CSMA/CD сети (например ethernet). В случае, если сеть не имеет коммутаторов, результатом является выход из строя на время атаки всей сети, если же есть коммутаторы - отказ работы одного сегмента сети, в котором стоит атакующая станция (в пределах коллизионного домена). Кстати, важно

понимать, что несмотря на то, что VLAN'ы призваны разделять броадкастные и коллизионные домены, к MAC-based VLAN'ам это не относится, поскольку они организуются поверх имеющихся коллизионных доменов.

- marker slowing - DoS атака на сети, построенные на принципах логического кольца с передачей маркера. Результатом является, как минимум, снижение пропускной способности сети, за счет задержек передачи маркера в рамках определенных стандартами таймаутов. Возможно есть шанс добиться DoS.

- arp poisoning - изменение таблицы соответствия MAC и IP-адресов на удаленном хосте путем посылки ему пакетов arp-reply с ложными данными о MAC-адресе, соответствующем интересующему IP-адресу. Эта атака очень близка к 2-му уровню OSI, однако ее скорее стоит позиционировать как находящуюся на стыке между 2-м и 3-м уровнями OSI.

- <protocol> overflow (flood) - использование потока блоков данных административно-информативных протоколов на максимально допускаемых носителем скоростях. К таким атакам относится в том числе cdp-flooding. Практическое обнаружение подобных атак затруднено необходимостью проводить массу тестирований на работу оборудования при нарушении спецификации, заданной стандартом. Например, как поведет себя коммутатор, если будет получать TCN/C-BPDU в количестве во много раз превышающем предусмотренную стандартом нагрузку[29]?

⊗

12 Устройства и программное обеспечение, подверженные описанным STP-атакам и устройства, индифферентные к таким атакам.

Устройства и комплектующие. Все устройства полностью совместимые с спецификацией Spanning Tree протокола [1] определенной в комитете стандартизации IEEE, или, другими словами, почти все интеллектуальное оборудование, обслуживающее локальные вычислительные сети и часть оборудования, используемого для создания WAN соединений. Это включает большинство достаточно интеллектуальных коммутаторов и некоторые маршрутизаторы. Не затронутыми данным типом атак будут все «глупые» устройства, например, концентраторы, поскольку они настолько глупы, что не поддерживают спецификацию ST

[29] Полученную, например, за счет несоблюдения пауз при посылке.

в [1]. Частично ситуация решается в устройствах, имеющих аналоги cisco BPDU-guard и BPDU-root-guard. Тем не менее, по умолчанию такие устройства также уязвимы, поскольку поддержка STP в них также включена по умолчанию, расширения STP выключены, а кроме того включение их никак не помогло бы покупателям этих устройств, поскольку настройка защиты от STP-атак требует знания топологии сети, и, самое главное, защититься можно отнюдь не от всех перечисленных атак.

Неполный список компаний, производящих так или иначе подверженные STP-атакам устройства включает: Cisco, Avaya(Lucent), 3Com, Intel, HP, Cabletron Systems и так далее - любой производитель, производящий интеллектуальные устройства с поддержкой спецификации [1] может «похвастаться» «дырявостью» этого оборудования. Особенно забавно читать слово secure в названии SFPS от Cabletron Systems в рекламном обзоре Ethernet Switching Bridge Media Interface Module от 1994 года - эта аббревиатура расшифровывается как «SecureFast Packet switching» - на первой же странице красуется поддержка 802.1d - security на небывалой высоте. :) Неважно, кто сделал то или иное STP-совместимое устройство - все они, поддерживая спецификацию Spaning Tree протокола, становятся уязвимыми, поскольку уязвимости эти заложены в самом протоколе.

Программное обеспечение. Проект реализации коммутатора на основе компьютера с OS Linux – Linux Bridge (см. секцию 20.2 в разделе 20). Как утверждается на основной странице разработчиков - проект полностью совместим с IEEE 802.1D спецификацией STP, поэтому коммутатор, построенный с использованием этого ПО на PC или промышленном ПК, будет также подвержен STP-атакам, как и любой другой коммутатор, совместимый с [1]. Поскольку функции «BPDU-guard» или «spanning-tree portfast» не заявлены, такой коммутатор не имеет никаких средств борьбы с STP-атаками. Мы настоятельно рекомендуем разработчикам включить поддержку этих возможностей в todo на ближайшее время, поскольку в этом случае проект будет иметь хотя бы какое-то подобие системы безопасности.

13 Примеры уязвимых сетей

Этот абзац статьи должен был быть посвящен изготовителям червей призванных устроить DoS врагу (например chineese и его бесславному Code Red, который так и не смог сделать DoS сайту www.white-сами-знаете.gov).

В случае применения материалов данного документа устройство DoS сводится к захвату контроля хотя бы над одним из компьютеров в атакуемой сети, подключенном к ЛВС атакуемого объекта. Правда дополнительным условием становится интеллектуальность оборудования, то есть поддержка STP-протокола. Однако на момент написания этих строк (Пт Ноя 2 03:27:47 2001) большая часть известного нам сетевого оборудования, поддерживающего STP, поставлялось с включенной по умолчанию («из коробки») поддержкой STP, что распространяет возможность использовать данную атаку на все сети с конфигурацией по умолчанию. Единственным приятным исключением были модели HP ProCurve 212M и 224M компании Hewlett Packard, но они не поддерживали VLANы, не говоря уже о раздельных STP деревьях в них.

Стоит также напомнить, что основное применение Spaning Tree – защита от образования колец в сети, например, за счет введения запасных каналов для повышения надежности. Такая структура сети наиболее вероятна в крупных правительственных учреждениях (например, сеть американских Белого Дома и Пентагона наверняка построена с использованием избыточности). Самое интересное, что в силу наличия дублирующих каналов, администратор в некоторых случаях не сможет отказаться от использования Spaning Tree. Надо заметить, что во многих известных мне устройствах, обслуживающих сеть, Spaning Tree включался и выключался для всего устройства, так что администратор просто не имеет возможности выключить работу Spaning Tree на одном из портов, не выключая этот порт. На сегодняшний день авторам не известны устройства, которые позволяли бы выключать поддержку STP на отдельном порту, хотя, при чтении документации к коммутатору intel 460T создается впечатление, что этот коммутатор умеет выключать поддержку STP индивидуально на каждом порту, но, скорее всего, имелся ввиду вариант с установкой порта в режим STP-blocking перманентно[30]. Однако устройства некоторых производителей (например, Cisco) имеют настройки, позволяющие защититься от *части* STP атак. Это расширения STP «spanning-tree portfast», BPDU guard и BPDU root guard.

К числу прочих организаций, которые могут пострадать от STP-DoS, относятся провайдеры Internet с последней милей на LAN-технологии (например Ethernet). Устройства у провайдеров с большой вероятностью должны быть интеллектуальными, и, в случае богатого провайдера (или

[30] Авторы будут благодарны, если кто-либо из людей, имевших дело с этими и другими активными устройствами на практике, развеет наши сомнения по отношению к данному прибору. Кстати, мы с удовольствием внесли бы в следующую переработанную публикацию ваши конструктивные отзывы в рамках тематики этой работы.

богатого клиента), они могут иметь дублирующие каналы[31].

Разумеется, в случае возникновения атаки в одной из двух ЛВС, соединенным неким устройством 2-го уровня OSI (bridge, hub), или даже маршрутизатором, который поддерживает spaning tree, DoS перекинется с одной ЛВС в другую, поскольку второй уровень прозрачен для STP протокола. Сущности, обменивающиеся STP-пакетами, используют мультикастовую адресацию, которая позволяет пакетам проходить в том числе и через оборудование, которое не поддерживает STP, например через концентраторы. Помимо этого подверженными STP-атакам будут хостинговые площадки провайдеров - стоит обратить внимание на то, что для организации DoS-атаки для некоего сервера не обязательно получать контроль над компьютером, предоставляющим место именно этому информационному ресурсу - достаточно будет получить контроль над компьютером, находящимся в том же сегменте ЛВС, что и основная жертва, так, чтобы между zombie host и victim host не было маршрутизатора.

Так же возможно распространение атаки через WAN-линки в случае, если оборудование настроено на частичный пропуск пакетов административных протоколов, включая STP. Сюда относится и обычное dialup-соединение. Для атаки поверх такого соединения злоумышленнику надо всего лишь изменить поведение своего модуля, осуществляющего PPP соединение, так, чтобы он запрашивал поддержку STP протокола при соединении. По умолчанию эта функциональность не запрашивается и атака невозможна, далее все зависит от настроек оборудования провайдера - если STP поднимется по запросу, DoS атака со стороны анонимного покупателя интернет карты может стать реальностью.

14 Как администраторы сетей могут противостоять атакам

Если обеспечение безопасности сети стоит не на последнем месте, лучше отключать поддержку Spaning Tree всегда, когда это возможно, и пытаться заменить STP другими технологиями там, где требуются решения с дублированием каналов, но все же можно обойтись без особенностей STP. Например, можно использовать технологию Link Agregation (поддерживается многими устройствами, в том числе произведенными Intel, Avaya и др.).

[31] Хотя для вывода оборудования из строя важна *всего лишь* поддержка STP, а никак не наличие дублируемых соединений. ;)

Если же безопасность важна, но выключить STP по тем или иным неутешительным причинам для всей сети неприемлемо, придется использовать расширения Spaning Tree доступные, в частности, в решениях, предоставляемых cisco[32], либо просто расслабиться и надеятся на то, что ваша сеть не попадет в поле интереса различных личностей с деструктивными наклонностями, например «кракеров»[33], или просто на то, что производители когда-нибудь исправят имеющуюся печальную ситуацию. Правда, здесь не совсем все так печально, как может показаться - помогут следующие административно-технические шаги:

1. Если у Вас менее 2-х WAN-линков между офисами, STP-домены следует разделить (то есть отключить поддержку STP на WAN-линке). См. раздел 4.

2. Если оборудование позволяет, следует выключить STP вовсе или оставить его только на тех портах, которые являются транковыми (тегированными) - такие порты смотрят обычно на другие сетевые устройства, но не на пользователей. Следовательно, если в рамках сети нет портов, на которых подключены рабочие станции и сервера, то это будет одним из вариантов достаточно устойчивой и (возможно, если я не пропустил чего ;)) безопасной картины. =)

3. Можно попробовать выставить минимально возможный bridge id и прочая (тут надо вписать подробности выбора корня), что сделает приоритет свитча максимально возможным, тогда он всегда будет root'ом и реконфигурация может разве что затронуть часть сети (ближайшие к атакующему свитчи, если атакующий использует подделку bpdu), за исключением установленного раз и навсегда STP-корня. Мы можем это сделать, поскольку STP корень выполняет всего лишь навсего роль точки отсчета, от которой строится топология без колец. Но вот незадача - последний параметр выбора (если все одинаковое) - MAC-адрес порта, что собственно считается гарантированно разным. Но если нужен DoS, то злоумышленник может создать некорректную с точки зрения стандарта ситуацию, когда MACи совершенно идентичны. При этом вопрос «как поведет себя коммутатор?» остается открытым. Результат требует проверки, поскольку нигде не документировано (насколько нам известно) как ведет себя мост в случае получения сообщения с собственным MAC.

[32]Которые, однако, не решают всех проблем.

[33]Наиболее удачное описание различий терминов «кракер» и «хакер» см. в [3]

15 Как IDS могут обнаружить STP атаки

В каждой организации, всерьез заботящейся о безопасности своих данных, должна стоять система обнаружения атак - IDS (Intrusion Detection System). В этом разделе рассматриваются особенности STP-атак, которые можно использовать для их обнаружения и трассировки атакующей стороны. К сожалению, большинство возможных способов настройки «сенсоров» IDS на STP-атаки требуют индивидуальной настройки на данную сеть, то есть потребуют от настраивающего как минимум хорошего представления о функционировании STP и знания всех особенностей данной сети в контексте функционирования STP протокола.

15.1 Сложности обнаружения STP-атак и их причины

Основная сложность обнаружения состоит в том, что для атаки используют вполне стандартные для протокола пакеты - BPDU. То есть присутствие STP-пакетов в сети однозначно означает STP-атаку лишь в одном из возможных случаев.

Вторая сложность (скорее даже особенность): большинство рассматриваемых возможностей обнаружения таких атак связаны с внесением в базу IDS некоторых данных о топологии сети ее администратором.

Следующая сложность связана с тем, что поскольку атака идет на топологию и работоспособность сети, IDS должна иметь собственный независимый канал для передачи сообщений ответственному за безопасность. Это может быть, например, передача сообщений через модем, через подсоединенный к IDS мобильный телефон (в частности, есть решения по соединению с PC телефонов Nokia через RS-232) или выделенное соединение точка-точка (например ethernet-кросс) между IDS и рабочей станцией ответственного за безопасность. Впрочем, это условие весьма типично и для других DoS атак.

15.2 Варианты обнаружения атак

15.2.1 Вариант обнаружения по наличию STP пакетов

Дано: В сети нет устройств, поддерживающих STP, либо поддержка STP отключена на всех устройствах сети, сеть не подсоединена к другим сетям, в которых есть STP-совместимые устройства.

Событие: обнаружены STP пакеты (C-BPDU или N-BPDU).

Статус: Это однозначно STP-атака.

Администрирование: Для того, чтобы IDS могла использовать этот метод, администратор должен внести в ее конфигурацию информацию о том, что в сети нет STP-совместимых устройств.

15.2.2 Вариант обнаружения по «чужому» Bridge ID

Дано: В сети всего N устройств, поддерживающих STP, либо поддержка STP отключена на всех остальных устройствах сети, сеть не подсоединена к другим сетям, в которых есть STP-совместимые устройства.

Событие: обнаружены STP пакеты (C-BPDU или N-BPDU) с root id (bridge id) иным, нежели у всех имеющихся устройств.

Статус: Это однозначно STP-атака.

Администрирование: Для того, чтобы IDS могла использовать этот метод, администратор должен внести в ее конфигурацию информацию о всех возможных bridge id.

15.2.3 Вариант обнаружения по длительности

Дано: В сети есть N STP-совместимых устройств, администратору необходима возможность подключения новых устройств с автоматическим вводом их в работу в сети, предполагается, что одновременно в сети не может появиться более чем K новых STP-совместимых устройств, bridge ID имеющихся устройств не документируются.

Событие 1: обнаружены STP выборы (смена Designated Root ID, возможно последовательная).

Событие 2: a) в течении периода в <(N+K)*X + ensure_time> секунд выборы периодически повторяются с разными Designated Root ID.
или
b) за любое время обнаружено N+K bridge ID.

Статус: Это однозначно STP-атака.

Администрирование: Для того, чтобы IDS могла использовать этот метод, администратор должен внести в ее конфигурацию информацию о устраивающих его значениях «ensure_time» (времени, суммируя которое с временем на прошедшие выборы можно получить время, за которое сеть должна основательно успокоиться (т.е. все выборы к окончанию этого периода пройти). Этот параметр можно рассчитать, исходя из количества STP-совместимых устройств в сети.

15.2.4 Вариант обнаружения по интенсивности

Дано: В сети есть N STP-совместимых устройств, администратору необходима возможность подключения новых устройств с автоматическим вводом их в работу в сети, предполагается, что одновременно в сети не может появиться более чем K новых STP-совместимых устройств, bridge ID имеющихся устройств не документируются.

Событие: За время t, меньшее, чем минимальное определенное в стандарте, приходит более одного C-BPDU с одинаковыми bridge ID или с одинаковыми designated root ID.

Статус: Это однозначно STP-атака.

Администрирование: Не требуется в случае параметра t, устанавливаемого по умолчанию, но полезной будет возможность выставить параметр t, поскольку это одна из регулируемых (по стандарту) настроек, которые может производить пользователь (администратор сети). В случае, если t настраиваемо, этот вариант становится частным случаем варианта 15.2.8.

15.2.5 Вариант обнаружения по монотонности

Дано: В сети есть некоторое количество STP-совместимых устройств, администратору необходима возможность подключения новых устройств с автоматическим вводом их в работу в сети.

Событие: За время меньшее чем <some_test_time>, bridge id или designated root ID или Designated bridge ID монотонно убывают.

Статус: Это STP-атака с очень большой вероятностью.

Администрирование: задание диапазона изменения этих id-величин, либо задание some_test_time (которое можно устанавливать в некоторое значение по умолчанию).

Комментарий: Для получения большей уверенности в том, что это атака, следует совместить данный способ с вариантом 15.2.2 или 15.2.3.

15.2.6 Вариант обнаружения по цикличности

Дано: В сети есть некоторое количество STP-совместимых устройств, администратору необходима возможность подключения новых устройств с автоматическим вводом их в работу в сети. Предполагается, что в сети нет устройств, сконфигурированных как STP-root-backup (см. раздел 8.1).

Событие: За время меньшее, чем <some_test_time>, bridge id или designated root ID или Designated bridge ID постоянно циклически меняются.

Статус: Это STP-атака с очень большой вероятностью.

Администрирование: не требуется, если some_test_time оставлено по умолчанию, однако, этот параметр должен быть настраиваемым.

15.2.7 Вариант обнаружения по потере производительности

Дано: В сети есть некоторое количество STP-совместимых устройств, администратору необходима возможность подключения новых устройств с автоматическим вводом их в работу в сети.

Событие: Резкое снижение пропускной способности сети.

Статус: Это может быть STP-атакой. Вероятность того, что это STP-MitM, зависит от величины изменения пропускной способности и должна подбираться экспериментальным путем - это один из самых сложных алгоритмов обнаружения STP-MitM атаки, реализация которого возможна скорее лишь теоретически, поскольку скорость работы в сети может меняться в зависимости от многих переменных параметров, например, от количества работающих пользователей и объема передаваемых ими друг другу данных. Такую атаку проще обнаружить визуально по субъективным ощущениям, чем искать способ заставить IDS уловить грань.

Администрирование: потребует от администратора задания нескольких параметров, определяющих допустимые скачки скорости работы сети.

15.2.8 Вариант обнаружения по изменению интервалов между ST событиями

Дано: В сети есть некоторое количество STP-совместимых устройств, администратору необходима возможность подключения новых устройств с автоматическим вводом их в работу в сети. Все параметры STP устройств установлены в одинаковые значения для всех STP устройств.

Событие: зарегистрировано STP-событие: вне графика согласно заданным значениям таймеров.

Статус: Это однозначно STP-атака.

Администрирование: потребует от администратора задания от одного до всех параметров, устанавливаемых на STP устройствах.

15.2.9 Невозможность обнаружить атаку. Пример

Дано: В сети всего N устройств, поддерживающих STP, поддержка STP включена на части устройств сети, сеть подсоединена к другим сетям, в которых есть STP-совместимые устройства, однако STP-root административно зафиксирован (например, заданием меньшего по модулю bridge ID, чем у остальных устройств), bridge id остальных устройств не введены в базу IDS, администратору необходима возможность подключения новых устройств с автоматическим вводом их в работу в сети.

Событие: обнаружены STP пакеты (C-BPDU или N-BPDU) с designated root id иным, нежели у устройства, bridge id которого зафиксирован как указано выше.

Статус: Это не может быть однозначно определено как STP-атака, поскольку в процессе выборов STP-root-bridge может произойти поочередная смена designated root между уже имеющимися STP-совместимыми устройствами, что приведет к рассылке подобных C-BPDU или N-BPDU.

Возможные решения: следует воспользоваться рекомендациями для случая 15.2.2 или случая 15.2.3.

16 Корни проблемы, или «откуда ноги растут»

Теоретически нет возможности *полностью* избежать описанных атак для протокола подобного STP, кроме как отключением поддержки этого протокола для всей сети, поскольку требуется (*все одновременно*): *и* возможность подключить сетевое устройство в любое место сложной ЛВС с динамической топологией любым возможным способом, *и* без дополнительной административной акции получить автоматическую правильную реконфигурацию топологии, *и* не использовать каких-либо криптографических методов для обеспечения аутентификации (проверки того, что данный BPDU послал тот, кто посылал их ранее). Использование для этой цели криптографических алгоритмов с открытым ключом мало поможет, так как вновь подключаемое устройство должно иметь возможнось переконфигурировать топологию. То, что в процессе реконфигурации кадры пользователей не должны передаваться, позволяет реализовать DoS, однако совершенно не влияет на работоспособность атаки MitM.

У этой проблемы есть несколько возможных (несколько - довольно практичных) решений, которые могли бы быть применены производителями сетевого оборудования. Ниже описываются возможные методы

решения, часть из которых, разумеется, может по различным причинам показаться спорной.

Мы расположили варианты в порядке увеличения сложности их применения.

17 Как производители могли бы отреагировать на появление в публичном доступе информации по STP-атакам

1. До тех пор, пока не будут найдены лучшие пути, устройства «из коробки» должны игнорировать STP протокол, как сделано, например, в коммутаторах Hewlett Packard HP212M и 224M (в них поддержка STP «из коробки» выключена). Это позволит избежать атак в тех ситуациях, когда STP не является жизненно необходимым элементом сети (например, в сети с единственным коммутатором).

2. Следует отказаться от принципа plug&play по отношению к сети: например, выпустив новую версию стандарта, которая будет требовать переконфигурации администратором некоего STP пароля (желательно нетранслируемого напрямую в сеть), который требовался бы для участия в изменении топологии по новому STP-протоколу, и без ввода которого поддержка STP была бы выключена. Так можно было бы значительно усложнить воспроизведение атаки на топологию сети. Ничего особенно страшного в том, что устройство перед вводом в полнофункциональную работу с сетью надо будет чуть-чуть отконфигурировать нет: обычно устройства все же конфигурируют, перед тем, как задействовать в сложных сетях, а сети, в которых STP работает не просто так, а с пользой, явно сложнее остальных, которых, кстати, большинство.:) Конкретная реализация может быть выполнена с использованием известных криптографических методов, например, при помощи Кодов Подтверждения Достоверности (Message Authentication Code, MAC). Для этого в пределах группы оборудования, которое должно образовать stp-дерево, выбирается «общий секрет» (пароль, ключ), заносимый в каждое устройство (лучше с помощью dip-switches на панели, т.е. аппаратно)[34]. После этого перед посылкой BPDU-пакета к нему добавляется секрет, для всего массива (пакет вместе

[34]Такой подход позволит автоматически выполнить и предыдущую рекомендацию, если положение переключателей по умолчанию («все в 0») будет означать для устройства отключение поддержки STP.

с «общим секретом») рассчитывается значение хеш-функции (например, SHA-1) и прибавляется к отправляемому BPDU-пакету («общий секрет», естественно, при этом не передается). На принимающей стороне к принятому пакету также добавляется секрет и рассчитывается значение хеш-функции, которое сравнивается с полученным в пакете. В результате получатель уверен, что пакет послал кто-то из «своих» (обладающих знанием секрета), при этом сам секрет по сети не передается и не может быть перехвачен злоумышленником для последующего использования.

3. Каждое устройство, поддерживающее STP, должно иметь возможность отключить обработку STP на каждом порту по отдельности, это реализуется переписыванием управляющего софта. Без выполнения этого условия применение такого устройства в сети становится небезопасным. Разумеется, в момент выключения STP на порту администратор должен понимать, что он лишается возможности использовать преимущества STP и ставит себе потенциальную ловушку. Однако есть масса случаев, когда это оправдано. Например, предоставление клиентам ISP сервиса по Ethernet сети - в этом случае сети всегда будут разделены и STP на клиентских портах никогда не принесет пользы. С точки зрения реализации, порты с выключенной поддержкой STP должны просто отбрасывать любые STP-BPDU и не должны посылать никаких STP-BPDU.

4. Крайне желательно, чтобы каждое STP-совместимое устройство поддерживало команду, аналогичную командам Cisco STP portfast и BPDU-guard. Это позволило бы избежать некоторых типов DoS (STP portfast), а также (BPDU-guard) локализовать сегмент с атакующим.

5. Каждое устройство, поддерживающее и STP, и VLANы, должно поддерживать отдельное STP-дерево на каждый VLAN, в противном случае атака в одном VLAN может привести к отказу работы всей сети (например, в случае использования intel 460T с включенным STP).

6. Алгоритм работы STP должен быть пересмотрен: на используемые алгоритмом таймауты должны быть наложены более жесткие ограничения, такие, чтобы стала невозможной постоянная реконфигурация устройств, при которой время, проводимое в состояниях с запрещенной трансляцией фреймов, не может быть больше времени устаревания информации.

7. Протокол STP требует доработки в части внесения в него функций обеспечения аутентичности и целостности передаваемой информации (пакетов BPDU), например, с применением криптографичес-

ких методов. Это позволило бы избежать ситуации когда конфигурация изменяется атакующим в любой момент времени посредством посылки bpdu от имени другого устройства.

8. Спецификация алгоритма STP (как и прочих административных протоколов, работающих на 2-м уровне OSI), а также спецификация канального уровня всех LAN могут быть расширены таким образом, чтобы NIC'и физически не могли общаться с устройствами, обслуживающими сеть (маршрутизаторы, мосты, концентраторы и т.п.) с использованием административных протоколов. Иными словами, административные протоколы, используемые коммутаторами для определения топологии, могли бы работать с использованием отдельной схемы электрической сигнализации (другой вольтаж или метод кодирования сигнала). Поскольку сетевые адаптеры конечных устройств не будут иметь такой схемы, они физически не смогут вмешаться в работу административных протоколов. Этот вариант изменения протокола представляется мало перспективным. Во-первых, он достаточно сложен в реализации (необходимо перепроектировать аппаратную часть). Во-вторых, аппаратная несовместимость остановит лишь часть личностей с деструктивными наклонностями (не умеющих держать в руках паяльник). В-третьих, сразу же встает вопрос обратной совместимости с ранее выпущенным оборудованием. Таким образом, несмотря на серьезные затраты, он не является панацеей.

9. Возможно, следует изменить STP протокол таким образом, чтобы пользовательские фреймы передавались всегда, предусмотрев при этом дополнительную маркировку фреймов, переданных в период реконфигурации маркером, который помогал бы уничтожать их по прошествии некоторого таймаута - в конце концов не так уж и много урона сети нанесут продублированные по разным каналам фреймы, в отличие от описанных в этой статье DoS атак.

10. Возможно, стоит создавать более сложные административные протоколы - с установлением соединения (подобно tcp). Это может серьезно усложнить спуфинг конфигурационных сообщений.

11. Также, возможно, стоит расширить STP таким образом, чтобы устройства запрашивали подтверждения происходящего поверх заблокированных каналов в тех случаях, когда это возможно. Такая модификация отбросит возможность определенных подвидов атак (в частности, MitM).

18 Эпилог

Хочется верить, что практически полное отсутствие в сетевых протоколах 2-го уровня OSI средств аутентификации сущностей административных протоколов, а также игнорирование других основ построения безопасных систем (за исключением, разве что, концепции VLAN-divided сетей), вызвано исключительно беспечностью авторов этих протоколов, создававших их во времена, когда о безопасности почти не беспокоились, а не результат «демократичности» законов США, обязывающих оставлять черные ходы для сотрудников спецслужб.

18.1 Кратко: что можно сделать с помощью нецелевого применения STP?

1. Злоумышленник может осуществить MitM-атаку (man-in-the-middle - «человек посередине»), если его рабочая станция подключена двумя или более интерфейсами к *различным* STP-совместимым устройствам, например, коммутаторам (switches).

2. Злоумышленник имеет возможность осуществить атаку на отказ в обслуживании (DoS) даже если его рабочая станция имеет всего один сетевой интерфейс, при условии, что функционирующие в сети STP-совместимые устройства не поддерживают BPDU-guard или STP portfast, а также в случае, если указанные возможности на этих устройствах просто не задействованы.

3. В случае, когда STP-совместимые устройства в сети поддерживают STP portfast, злоумышленник может спровоцировать ситуацию, когда STP-совместимый коммутатор (switch) переходит в режим концентратора (hub), и прослушивать при помощи сетевого анализатора весь трафик между узлами сети, подключенными к этому коммутатору. Эта атака частично реализуема также и для случая с несколькими коммутаторами, хотя расчеты немного усложнятся.

18.2 Кратко: чего нельзя сделать с помощью нецелевого применения STP?

1. Невозможно организовать STP-MitM атаку, имея только один сетевой интерфейс, но это становится возможным в случае, если этот интерфейс подключить к концентратору, при этом поведение STP-совместимого устройства неочевидно.

2. Злоумышленник не может производить изменения в топологии сети, будучи подключенным только к одному порту коммутатора, не

спровоцировав при этом отказ в обслуживании (DoS) для какой-либо части сети. Злоумышленник не может реализовать изменения в топологии, касающиеся некоторого сегмента сети, подключившись к коммутатору, который имеет только одно соединение с этим сегментом, не спровоцировав при этом отказ доступа к этому сегменту из той части сети, которую затрагивают внесенные изменения.

3. В ethernet технологии невозможно организовать MitM атаку с рабочей станции злоумышленника для двух конечных узлов ЛВС до тех пор, пока нет физического кольца между этими точками. Хорошим примером будет случай двух обширных сетей, соединенных между собой по единственному не дублируемому кабелю. В таком случае организовать MitM между станциями в разных сетях невозможно без прокладки кабеля для организации кольца. Такое кольцо может быть образовано самим злоумышленником, если он имеет физический доступ к коммутаторам этой сети и порты коммутатора по умолчанию включены (настройки по умолчанию для всех известных авторам устройств).

19 Благодарности

19.1 Олег Артемьев

Хочу сказать спасибо:

- Моей жене, Виктории Артемьевой – за терпение проявленное в моменты, когда я уделял время исключительно компьютеру и документации.
- Metaltelecom ISP (www.mtcm.ru) и МИСиС (www.misis.ru) – за интересную работу и возможность изучать интеллектуальное «железо» на практике.
- Avaya Communications – за организацию тренингов по их сетевым устройствам.
- Николаю И. Лаптеву, инструктору Microinform – за очень интересные лекции.
- Компании Microinform (www.microinform.ru) – за предоставление лабораторного оборудования для проведения испытаний, а также, еще раз, Н.И. Лаптеву, за содействие в организации лабораторных исследований в Microinform.
- Simple Nomad'у и команде www.nmrc.org – за их FAQ'и, которые поддержали мой интерес к IT security.

- Iron_Lung (i_L,ex-hrg, Russian Federation) – за то, что нашел для меня FAQ Nomad'овский и еще просто так. :)
- Сергею В. Расникову <rserg@mtcmNOSPAM.ru> – за то, что обратил мое внимание на сосуществование VLAN и STP, а также за комментарии к моим идеям.
- Владиславу Мяснянкину, моему соавтору в этой статье – за то, что согласился поучаствовать в разработке, и просто потому, что он – интересный человек.:)
- Некоторым преподавателям Московского Авиационного института, особенно г-ну Роговцеву А. А. (13 ф-т) – за умение заинтересовать предметом и стиль ведения занятий.
- Дмитрию Голубеву <dgoloube@ciscoNOSPAM.com> – за комментарии к первому варианту статьи и активное ее обсуждение.

19.2 Владислав Мяснянкин

Хочется сказать спасибо:

- Моей семье – за понимание и заботу во время ночных бдений перед монитором.
- Олегу Артемьеву – за его оригинальный и нестандартный подход к изучению сетевых технологий, а также за то, что я смог поучаствовать в меру сил и возможностей в этом проекте и что он терпимо относился к моей безынициативности в процессе реализации. ;)
- Константину Тайтурову – за то, что в 1995 году он притащил мне древний дистрибутив Slackware Linux и рассказал, что это рулез.
- Donald E. Knuth, придумавшему TEX. Использованная при подготовке данного документа система LATEX 2_ε сэкономила нам массу времени и помогла структурировать материал не только на бумаге, но и, что важнее, в голове.

20 Ссылки на связанные с этим исследованием статьи, стандарты, обзоры и другая справочная информация

20.1 Ссылки, относящиеся к Spanning Tree

1. http://www.protocols.com/pbook/bridge.htm#BPDU
2. Описание STP port fast в Cisco.
   http://www.cisco.com/warp/public/473/65.html
3. Обзор по troubleshooting с STP.
   http://www.cisco.com/warp/public/473/5.html#tshoot
4. Общая статья cisco по STP.
   http://www.cisco.com/warp/public/473/#SpanningTree
5. Описание поддержки STP в 3COM SuperStack II Switch 1000 User Guide.
   http://support.3com.com/infodeli/tools/switches/s_stack2/3c16902/manual.a02/chap51.htm

20.2 Ссылки, относящиеся к теме «bridging»

1. http://bridge.sourceforge.net/
2. http://www.freebsd.org/handbook/bridging.html
3. [FreeBSD-host / $] man bridge
4. [FreeBSD-host / $] links /usr/share/doc/handbook/bridging.html

20.3 Ссылки по конструированию фреймов

1. http://netgroup-serv.polito.it/winpcap/
2. http://www.eeye.com/html/Research/Tools/libnetnt.html
3. http://www.packetfactory.net/libnet/

20.4 Open Systems Interconnection refernce model

1. http://www.rad.com/networks/1994/osi/osi.htm

20.5 Другие интересные ссылки

1. http://www.phenoelit.de/stuff/CiscoCDP.txt
2. Assigned numbers.
   http://www.iana.org/numbers.html

Список литературы

[1] MEDIA ACCESS CONTROL (MAC) BRIDGES ANSI/IEEE Std 802.1D, 1998 Edition

[2] Open System Interconnection (OSI) reference model

[3] Медведовский И.Д., Семьянов П.В., Леонов Д.Г. «Атака на internet»

[4] ANSI/IEEE Std 802.1Q

[5] Request For Comments 2878 (RFC 2878), PPP Bridging Control Protocol (BCP), Network Working Group, M.Higashiyama (Anritsu), F. Baker (Cisco), July 2000.

[6] IEEE Std 802.3, 2000 Edition: Part 3: Carrier sence multiple access with collision detection (CSMA/CD) access method and physical layer specifications.

[7] Reynolds, J. and J. Postel, «Assigned Numbers», STD 2, RFC 1700, October 1994. See also: http://www.iana.org/numbers.html

Глоссарий

мост - bridge:

- двухпортовый коммутатор;
- некое устройство соединяющее два или более сегментов LAN, при этом, возможно, с разными методами MAC. Работа такого устройства может определяться различными стандартами, в том числе разработанными IEEE.

коммутатор - switch (см. также *мост*).

таблица коммутации - switching table, внутренняя таблица соответствия port/mac[/vlan], используемая устройством для коммутации пакетов.

STP - Spaning Tree Protocol.

ST - в рамках этой статьи - Spaning Tree алгоритм.

ЛВС - LAN, Local Area Network, Локальная Вычислительная Сеть, ЛВС.

ISP - Internet Service Provider, поставщик услуг интернет.

NIC - Network Interface Card, сетевая карта.

cdp - cisco discovery protocol, используется в cisco устройствах для получения информации о «соседних» cisco устройствах. Собственность cisco.

BVI - Bridging Virtual Interface, некий виртуальный интерфейс, который можно настроить например на маршрутизаторе и тем самым ускорить его работу за счет коммутации, а не маршрутизации части пакетов.

A Программа формирования ST пакетов

```
/*
 * Written 2001-09-23
 *
 * Copyright 2001 Vladislav V. Myasnyankin, Yekaterinburg Russia
 * hugevlad@yahoo.com http://cybervlad.port5.com
 * All rights reserved.
 *
 * WARNING! Be very careful, because you can occasionaly crash
 * your network.
 *
 * This program is free software; you can redistribute it and/or
 * modify it under the terms of the GNU Library General Public License
 * as published by the Free Software Foundation; either version 2 of
 * the License, or (at your option) any later version.
 *
 * THIS SOFTWARE IS PROVIDED BY THE AUTHOR ``AS IS'' AND ANY EXPRESS OR
 * IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED
 * WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE
 * ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT,
 * INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES
 * (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR
 * SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)
 * HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT,
 * STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING
 * IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE
 * POSSIBILITY OF SUCH DAMAGE.
 */

#include <stdio.h>
#include <unistd.h>
#include <ctype.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <netdb.h>
#include <arpa/inet.h>
#include <net/if.h>
#include <netinet/if_ether.h>
#include <sys/ioctl.h>
#include <sys/socket.h>

#define DEFAULT_DEVICE "eth0"
```

```
#define ETH_HW_ADDR_LEN 6

char usage[]={
"usage: stp \
[-v] \
[-dev <device>] \
[-dmac <dmac>] \
[-smac <smac>] \n\
-protoid <proto_id> \
-protovid <proto_v_id> \
-bpdu <bpdutype> \
-flags <flags> \n\
-rootid <rootid> \
-rootpc <rootpc> \
-brid <brid> \
-portid <portid> \
-mage <mage> \n\
-maxage <maxage> \
-htime <hellotime> \
-fdelay <fdelay> \n\
where:\n\
-v - be verbose and write output to file packet.dmp instead socket \n\
device - ethernet device name (default - eth0)\n\
dmac - destination MAC (default - 01:80:C2:00:00:00)\n\
smac - source MAC (default - MAC on given or default device)\n\
proto_id - Protocol Identifier (hex, 2 bytes)\n\
proto_v_id - Protocol Version Identifier (hex, 1 byte)\n\
bpdutype - BPDU type (hex, 1 byte)\n\
flags - flags value (hex, 1 byte)\n\
rootid - Root Identifier (hex, 8 bytes)\n\
rootpc - Root Path Cost (hex, 4 bytes)\n\
brid - Bridge Identifier (hex, 8 bytes)\n\
portid - Port Identifier (hex, 2 bytes)\n\
mage - Message Age (hex, 2 bytes)\n\
maxage - Max Age (hex, 2 bytes)\n\
hellotime - Hello Time (hex, 2 bytes)\n\
fdelay - Forward Delay (hex, 2 bytes)\n\
\n"};

struct bpdu_packet {
        u_char targ_hw_addr[ETH_HW_ADDR_LEN]; /* dest. ether address */
        u_char src_hw_addr[ETH_HW_ADDR_LEN];  /* src. ether address */
        u_char frame_type[2];                 /* 0x00 0x26 */
        u_char llc_dest;                      /* 0x42 */
```

```
        u_char llc_src;             /* 0x42 */
        u_char unknown;             /* 03 why? I don‘t know :( */
        u_char proto_id[2];         /* Protocol Identifier */
        u_char proto_v_id;          /* Protocol Version Identifier */
        u_char bpdu_type;           /* BPDU Type */
        u_char flags;               /* Flags */
        u_char rootid[8];           /* Root Identifier */
        u_char rootpc[4];           /* Root Path Cost */
        u_char brid[8];             /* Bridge Identifier */
        u_char portid[2];           /* Port Identifier */
        u_char mage[2];             /* Message Age */
        u_char maxage[2];           /* Max Age */
        u_char hellotime[2]         /* Hello Time */
        u_char fdelay[2];           /* Forward Delay */
        u_char padding[8];          /* padding packet to 60 bytes */
};

void fireexit(char *); /* print error message and exit with code 1*/
void get_hex_value(char*,char*,int); /* convert string to hex */


/**** MAIN PART *****/
int main(int argc,char** argv){

FILE *f;
struct bpdu_packet pkt1;
struct sockaddr saddr;
struct ifreq ifr;
int sd,i;
char eth_dev[30];
unsigned char verbose=0;
unsigned int complete=0;
/*
array of bites:
14 device        7 rootid        0 fdelay
13 dmac          6 rootpc
12 smac          5 brid
11 proto_id      4 portid
10 proto_v_id    3 mage
9 bpdutype       2 maxage
8 flags          1 hellotime
*/
```

```
strncpy(eth_dev,DEFAULT_DEVICE,29);

for(i=1; i<argc; i++){
   if (!strncasecmp(argv[i],"-v",2)) verbose=1;
   if (!strncasecmp(argv[i],"-dev",4)){
      if (verbose) printf("Device: ");
      strncpy(eth_dev,argv[++i],29);
      if (verbose) printf("Ok\n");
      complete|=0x4000;
      };
   if (!strncasecmp(argv[i],"-dmac",5)){
      if (verbose) printf("Destination MAC: ");
      get_hex_value(pkt1.targ_hw_addr,argv[++i],6);
      if (verbose) printf("Ok\n");
      complete|=0x2000;
      };
   if (!strncasecmp(argv[i],"-smac",5)){
      if (verbose) printf("Source MAC: ");
      get_hex_value(pkt1.src_hw_addr,argv[++i],6);
      if (verbose) printf("Ok\n");
      complete|=0x1000;
      };
   if (!strncasecmp(argv[i],"-protoid",8)){
      if (verbose) printf("Protocol Identifier: ");
      get_hex_value(pkt1.proto_id,argv[++i],2);
      if (verbose) printf("Ok\n");
      complete|=0x0800;
      };
   if (!strncasecmp(argv[i],"-protovid",9)){
      if (verbose) printf("Protocol Version Identifier: ");
      get_hex_value(&pkt1.proto_v_id,argv[++i],1);
      if (verbose) printf("Ok\n");
      complete|=0x0400;
      };
   if (!strncasecmp(argv[i],"-bpdu",5)){
      if (verbose) printf("BPDU Type: ");
      get_hex_value(&pkt1.bpdu_type,argv[++i],1);
      if (verbose) printf("Ok\n");
      complete|=0x0200;
      };
   if (!strncasecmp(argv[i],"-flags",6)){
      if (verbose) printf("Flags: ");
      get_hex_value(&pkt1.flags,argv[++i],1);
      if (verbose) printf("Ok\n");
```

```
   complete|=0x0100;
   };
if (!strncasecmp(argv[i],"-rootid",7)){
   if (verbose) printf("Root Identifier: ");
   get_hex_value(pkt1.rootid,argv[++i],8);
   if (verbose) printf("Ok\n");
   complete|=0x0080;
   };
if (!strncasecmp(argv[i],"-rootpc",7)){
   if (verbose) printf("Root Path Cost: ");
   get_hex_value(pkt1.rootpc,argv[++i],4);
   if (verbose) printf("Ok\n");
   complete|=0x0040;
   };
if (!strncasecmp(argv[i],"-brid",5)){
   if (verbose) printf("Bridge Identifier: ");
   get_hex_value(pkt1.brid,argv[++i],8);
   if (verbose) printf("Ok\n");
   complete|=0x0020;
   };
if (!strncasecmp(argv[i],"-portid",7)){
   if (verbose) printf("Port Identifier: ");
   get_hex_value(pkt1.portid,argv[++i],2);
   if (verbose) printf("Ok\n");
   complete|=0x0010;
   };
if (!strncasecmp(argv[i],"-mage",5)){
   if (verbose) printf("Message Age: ");
   get_hex_value(pkt1.mage,argv[++i],2);
   if (verbose) printf("Ok\n");
   complete|=0x0008;
   };
if (!strncasecmp(argv[i],"-maxage",7)){
   if (verbose) printf("Max Age: ");
   get_hex_value(pkt1.maxage,argv[++i],2);
   if (verbose) printf("Ok\n");
   complete|=0x0004;
   };
if (!strncasecmp(argv[i],"-htime",6)){
   if (verbose) printf("Hello Time: ");
   get_hex_value(pkt1.hellotime,argv[++i],2);
   if (verbose) printf("Ok\n");
   complete|=0x0002;
   };
```

```
   if (!strncasecmp(argv[i],"-fdelay",7)){
      if (verbose) printf("Forward delay: ");
      get_hex_value(pkt1.fdelay,argv[++i],2);
      if (verbose) printf("Ok\n");
      complete|=0x0001;
      };
};

/* Check, if all needed parameters set */
if ((complete & 0x0FFF) < 0x0FFF) fireexit(usage);

/* Set constant values */
pkt1.frame_type[0]=0x00;
pkt1.frame_type[1]=0x26;
pkt1.llc_dest=0x42;
pkt1.llc_src=0x42;
pkt1.unknown=03;
bzero(pkt1.padding,8);

if (verbose) {
    f=fopen("packet.dmp","a");
    fwrite(&pkt1,1,sizeof(pkt1),f);
    fclose(f);
    exit(0);
    };

/* Open raw socket */
if ((sd = socket(AF_INET, SOCK_PACKET, htons(ETH_P_ALL))) < 0){
        perror("Can't get raw socket: ");
        exit(1);
        }

/* Get our hardware address, if not given */
if (!(complete & (12<<1))){
    strcpy(ifr.ifr_name, eth_dev);
    if (ioctl(sd, SIOCGIFHWADDR, &ifr) < 0){
 perror("Can't get hardware address: ");
         close(sd);
         exit(1);
        };
    memcpy(pkt1.src_hw_addr,ifr.ifr_hwaddr.sa_data,6);
    };

/* Set device to use */
```

```
strcpy(saddr.sa_data,eth_dev);

/* Send prepared packet */
if(sendto(sd,&pkt1,sizeof(pkt1),0,&saddr,sizeof(saddr)) < 0)
   perror("Send packet");

/* Close socket */
close(sd);

exit(0);
}

/*************** END MAIN PART ********************/

void fireexit(char* str){
fprintf(stderr,"%s\n",str);
exit(1);
}

void get_hex_value(char* buf,char* str,int len){

int i;
char c,val;

for(i=0;i<len;i++){
        if( !(c = tolower(*str++))) fireexit("Invalid hex value");
        if(isdigit(c)) val = c-'0';
        else if(c >= 'a' && c <= 'f') val = c-'a'+10;
        else fireexit("Invalid hex value");

        *buf = val << 4;
        if( !(c = tolower(*str++))) fireexit("Invalid hex value");
        if(isdigit(c)) val = c-'0';
        else if(c >= 'a' && c <= 'f') val = c-'a'+10;
        else fireexit("Invalid hex value");

        *buf++ |= val;

        if(*str == ':')str++;
        }
}
```

B Сценарий для запуска программы формирования ST пакетов

```
#!/bin/sh
#
# Copyright 2001 Vladislav V. Myasnyankin, Yekaterinburg Russia
# hugevlad@yahoo.com http://cybervlad.port5.com
# All rights reserved.
#
# WARNING! Be very careful, because you can occasionaly crash
# your network.
#
# SEE DISCLAMER IN MAIN PROGRAM
#
# note:
# all numbers can be like 00010203040506 or like 00:01:02:03:04:05:06
#

device=eth0             # ethernet device name (default - eth0)
dmac=01:80:C2:00:00:00  # destination MAC (default - 01:80:C2:00:00:00)
smac=00:01:38:00:b4:c7  # source MAC (default - MAC on given
                        # or default device)
proto_id=0000           # Protocol Identifier (hex, 2 bytes)
proto_v_id=00           # Protocol Version Identifier (hex, 1 byte)
bpdutype=00             # BPDU type (hex, 1 byte)
flags=00                # flags value (hex, 1 byte)
rootid=800000013800b4c7 # Root Identifier (hex, 8 bytes)
rootpc=00000000         # Root Path Cost (hex, 4 bytes)
brid=800000013800b4c7   # Bridge Identifier (hex, 8 bytes)
portid=8002             # Port Identifier (hex, 2 bytes)
mage=0000               # Message Age (hex, 2 bytes)
maxage=1400             # Max Age (hex, 2 bytes)
hellotime=0200          # Hello Time (hex, 2 bytes)
fdelay=0f00             # Forward Delay (hex, 2 bytes)

./stp -v -dev $device -dmac $dmac -smac $smac -protoid $proto_id\
-protovid $proto_v_id -bpdu $bpdutype -flags $flags\
-rootid $rootid -rootpc $rootpc -brid $brid -portid $portid\
-mage $mage -maxage $maxage -htime $hellotime -fdelay $fdelay
```