PHREAKING ГАДЖЕТЫ КЕЙСЫ ВЗЛОМА

Гадкий утенок. Превращаем обычную флешку в USB Rubber Ducky

Антон Жуков Сен 11, 2015 6 мин на чтение 2 0 17227

Содержание статьи

Как-то давно мы делали в журнале обзор девайсов, которые было бы желательно иметь в своем чемоданчике хакера. Среди прочих девайсов там был и USB Rubber Ducky — устройство, внешне напоминающее обычную флешку, которое притворяется клавиатурой и при подключении к компьютеру быстренько набирает все заданные в нем команды. Штука крутая и очень полезная при проведении пентестов, но зачем выкладывать за нее 40 баксов (да еще и при текущем курсе), если аналогичным трюкам можно научить обычную флешку?

WARNING

Не забывай, что нижеописанные действия с флешкой могут не только лишить тебя гарантии на устройство, но и запросто убить девайс. Экспериментируй на свой страх и риск!

Предисловие

Прошлогодний Black Hat принес много интересных докладов. В числе наиболее обсуждаемых был доклад, посвященный неисправимой уязвимости USB-устройств, позволяющей превращать обычные флешки в инструмент распространения вредоносных программ. Атаку назвали BadUSB, но позже в Сети появились шуточки на тему «USBola», сравнивающие эту атаку с известным вирусом.

Подобные идеи использования HID-девайсов для корыстных целей были уже давно. Грех не воспользоваться тем, что ОС система доверяет устройствам, подключаемым к USB-интерфейсу. Если покопаться в памяти, то в журнале уже была статья по сходной тематике, в которой говорилось, как с помощью специального устройства Teensy можно взять под контроль машину с Windows 7 (в принципе — с любой ОС на борту). Устройство по внешнему виду напоминало собой обычную флешку, под которую собственно и маскировалось. Все это наводило на мысли, что с флеш-накопителями тоже можно провернуть такой трюк.

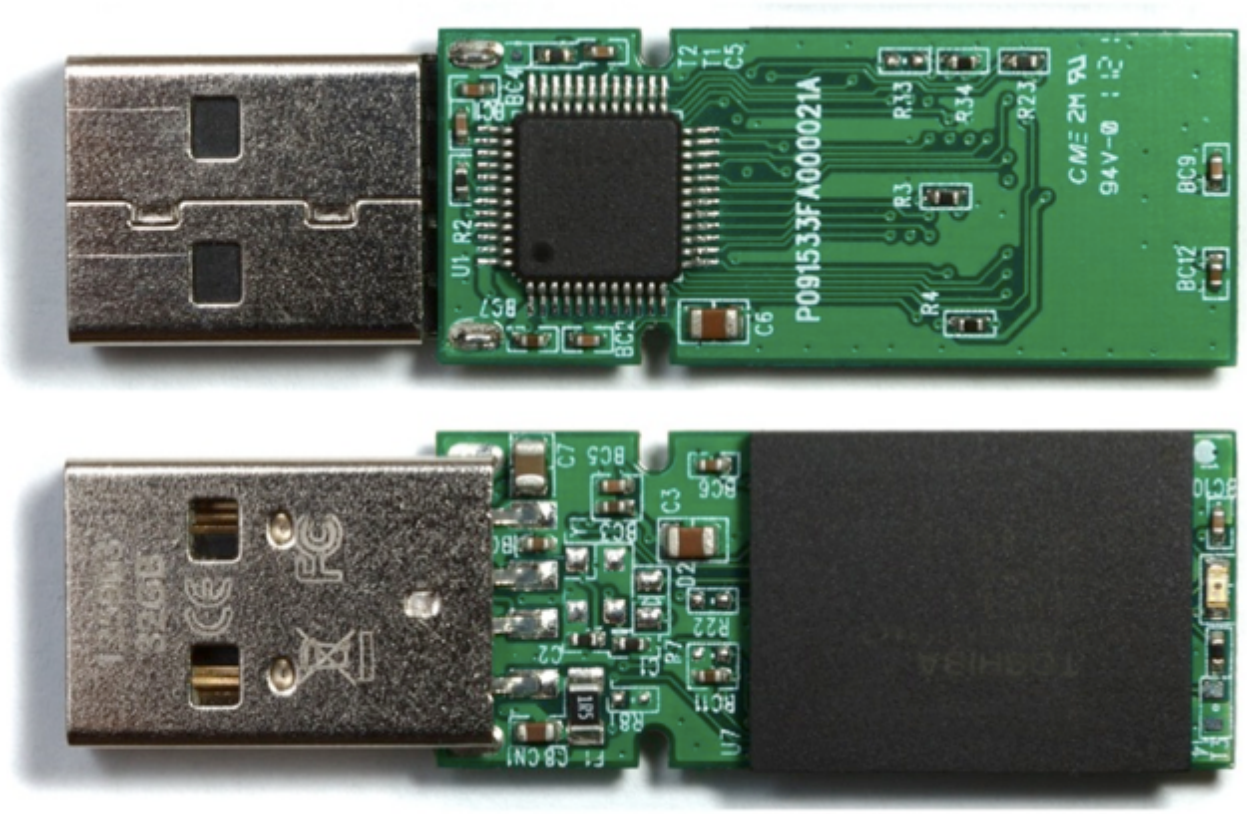
Предпосылки

Вообще, USB — очень универсальный интерфейс. Только подумай, сколько устройств мы к нему подключаем и в состав каких девайсов он входит! Мышки, клавиатуры, принтеры, сканеры, геймпады, модемы, точки доступа, веб-камеры, телефоны и т.д. и т.п. Мы не задумываясь вставляем коннектор в нужный разъем, ОС автоматически определяет тип устройства и подгружает необходимые драйвера.

Но как она это делает?

Устройство Flash накопителей

На самом деле, ОС ничего не знает о подключаемом устройстве. Ей приходится ждать, пока девайс сам не сообщит, к какому классу устройств он принадлежит. Если взять простейший пример, когда мы втыкаем флешку в USB-разъем, то флешка сообщает операционной системе не только что является накопителем, но и свой объем. Тут сразу вспоминаются хитрожелтые китайские товарищи, которые таким образом научились выпускать флешки повышенной емкости (встречались чуть ли не на пару терабайт). Чтобы разобраться, как такое возможно, давай вспомним (или узнаем), как система распознает USB-устройства.



Флешка без красивой обертки

Алгоритм инициализации USB устройств

Назначение USB-устройств определяется кодами классов, которые сообщаются USB-хосту для загрузки необходимых драйверов. Коды классов позволяют унифицировать работу с однотипными устройствами разных производителей. Устройство может поддерживать один или несколько классов, количество которых определяется количеством конечных точек (USB endpoints). В момент подключения хост запрашивает у устройства ряд стандартизованных сведений (дескрипторов), на основании которых принимает решение, как с этим устройством работать. Дескрипторы содержат сведения о производителе и типе устройства, на основании которых хост подбирает программный драйвер.

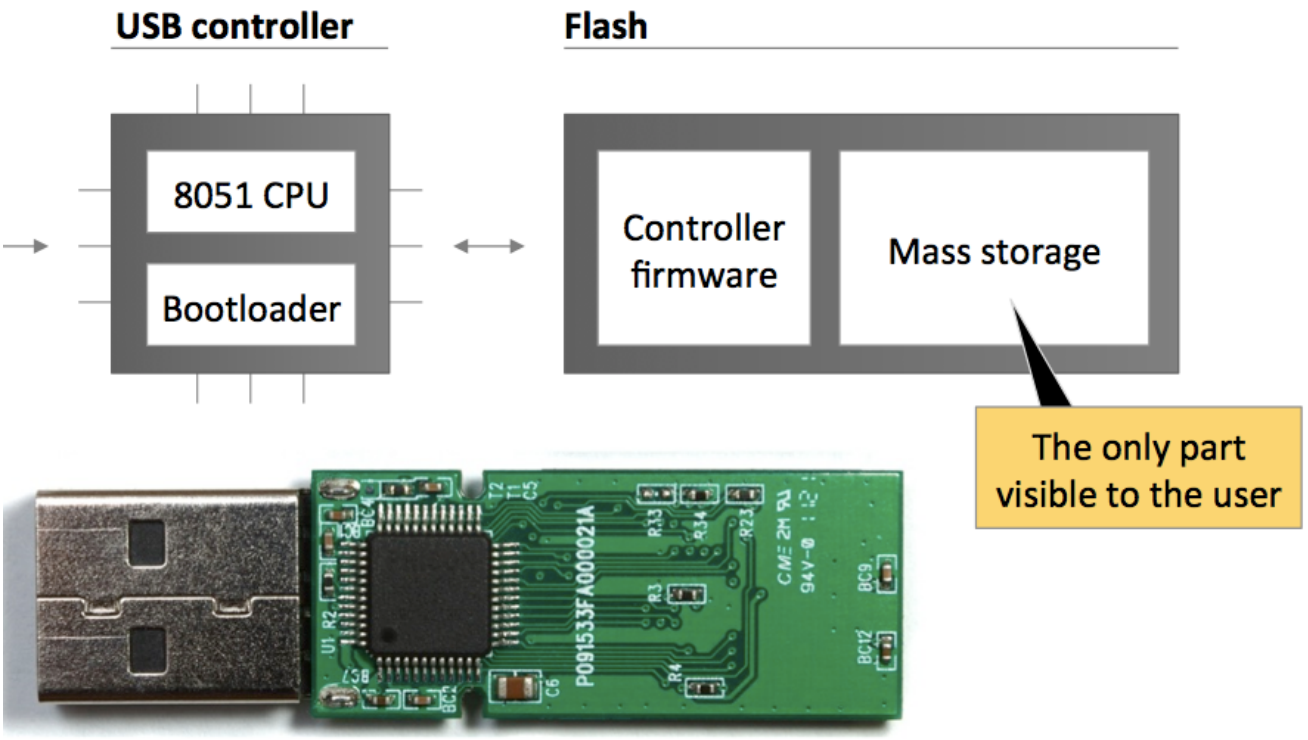
Обычная флешка будет иметь код класса `08h` (Mass Storage Device — MSD), в то время как веб-камера, снабженная микрофоном, будет характеризоваться уже двумя: `01h` (Audio) и `0Eh` (Video Device Class).

	Examples	
Identifier	USB thumb drive	Webcam
Interface class	8 – Mass Storage	a. 1 – Audio b. 14 – Video
End points	0 – Control 1 – Data transfers	0 – Control 1 – Video transfers 6 – Audio transfers 7 – Video interrupts
Serial number (optional)	AA627090820000000702	0258A350

Классы устройств

При подключении USB-устройства оно регистрируется, получает адрес и отправляет свой дескриптор/дескрипторы, чтобы ОС загрузила необходимые драйвера и отправила обратно необходимую конфигурацию. После этого начинается непосредственное взаимодействие с устройством. По завершении работы происходит дерегистрация девайса. Важный момент, который стоит тут отметить: устройства могут иметь несколько дескрипторов, а также могут дерегистрироваться и регистрироваться в качестве другого устройства.

Если вскрыть корпус флешки, то помимо запоминающего устройства (Mass Storage), видимого пользователю, на плате будет еще и контроллер, отвечающий за описанные выше действия.



Единственная часть устройства, видимая пользователю

Bad USB или немного истории

Итак, на конференции Black Hat в прошлом году двое исследователей (Karsten Nohl и Jakob Lell) поделились с общественностью опытом, как перепрошить контроллер флешки своей прошивкой. По истечении некоторого времени такая флешка регистрировалась в качестве клавиатуры и набирала заданные команды. Из-за серьезности проблемы ребята не стали выкладывать код эксплойта. Однако, спустя некоторое время, двое других исследователей (Adam Caudill и Brandon Wilson) уже на конференции Derbycon представили миру работоспособный PoC, заточенный под микроконтроллер Phison 2251-03. Код доступен на github.

Трансформация

Как ты понял, сегодня мы попробуем превратить обычную флешку в секретное оружие пентестера!

Прежде всего нам понадобится подходящий девайс. Так как код выложен только для конкретного микроконтроллера, то у нас есть два варианта — либо найти флешку, управляемую данным контроллером, либо провести очень непростую работу по исследованию и перепрошивке любого другого микроконтроллера. В этот раз мы выберем более легкий путь и попробуем найти подходящую флешку (а вот и список узявимого оборудования). Контроллер достаточно распространенный, так что даже каким-то чудом у меня дома среди десятка флешек нашлась подходящая.

Начинаем колдовать

Найдя подходящий девайс (который не жалко в случае неудачи потерять), можно приступать к его перевоплощению. Прежде всего нам потребуется скачать исходники, которые выложили ребята. В принципе, содержание расписано у них в официальной вики, но на всякий случай еще раз напомню, что же они выложили на гитхаб:

- DriveCom — приложение для взаимодействия с флешками, основанными на контроллере Phison;
- EmbedPayload — приложение, предназначенное для встраивания RubberDucky-скриптов `inject.bin` в кастомную прошивку с целью их последующего выполнения при подключении флешки;
- Injector — приложение, извлекающее адреса из прошивки и встраивающее код патча в прошивку;
- firmware — кастомная 8051 прошивка, написанная на C;
- patch — коллекция 8051 патчей, написанных на C.

Подготавливаем систему

Скачав с гитхаба архив с сорцами, ты обнаружишь, что большинство из них написано на C# и нуждается в компиляции, поэтому без студии не обойтись. Еще один инструмент, который понадобится — Small Device C Compiler, или SDCC. Его надо будет установить в `C:\Program Files\SDCC`, он понадобится для компиляции прошивки и патчей.

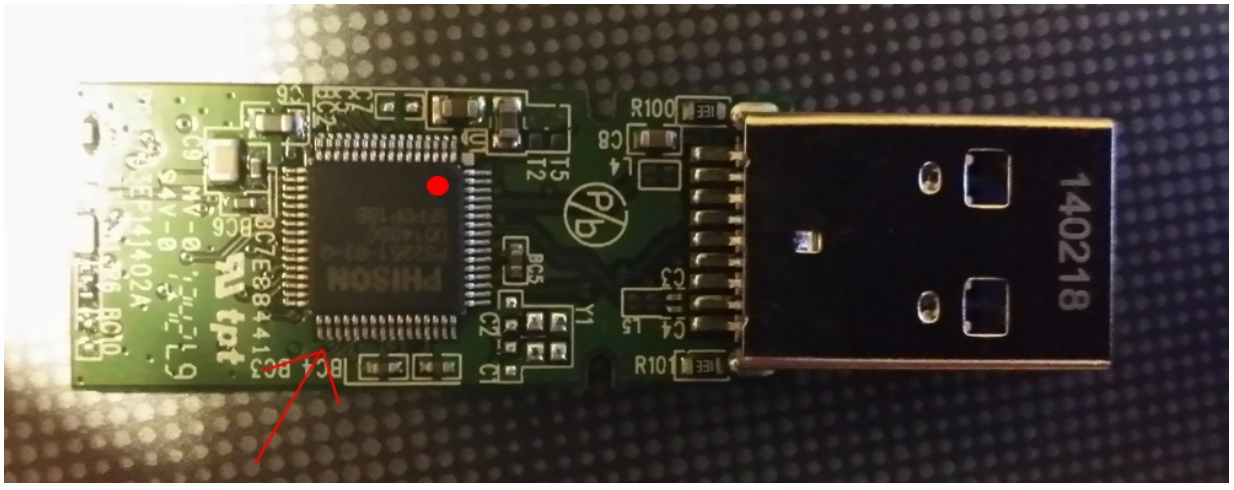
Скомпилировав все инструменты, входящие в архив, можно будет еще раз проверить, подходит ли данная флешка для перепрошивки:

```
DriveCom.exe /drive=F /action=GetInfo
```

где `F` — соответственно, буква накопителя.

TIPS

Если эксперименты пошли не так и с флешкой творится что-то непонятное, то можно попытаться вернуть ее к жизни, вручную переведя ее в boot-режим, и воспользовавшись утилитой для восстановления оригинальной прошивки. Для этого надо перед ее подключение замкнуть 1 и 2 (иногда 2 и 3) контакты контроллера, расположенные по диагонали от точки (чтобы было понятней смотри соотвествующий рисунок). После этого можно попытаться восстановить устройство с помощью официальной утилиты MPAL



Переводим флешку в boot-режим, замыкая указанные контакты

Получаем burner image

Следующим важным шагом является выбор подходящего burner image-а (8051 бинарник, ответственный за действия по дампу и заливке прошивки на устройство). Обычно их имена выглядят примерно так:

```
BNxxVyyyz.BIN
```

Где `xx` — номер версии контроллера (например, в случае PS2251-03 это будет 03), `yyy` — номер версии (не важно), а `z` отражает размер страницы памяти и может быть следующим:

- 2KM — для 2K NAND чипов;
- 4KM — для 4K NAND чипов;
- M — для 8K NAND чипов.

Где искать подходящий burner image для своей флешки, можно посмотреть по этой ссылке.

Дампим оригинальную прошивку

Прежде чем приступить к своим грязным эсперементам, которые могут убить флешку, настоятельно рекомендуется все таки сделать дамп оригинальной прошивки, чтобы, если что-то пойдет не так, можно было попытаться восстановиться работоспособность устройства. Сначала переводим девайс в boot-режим:

```
tools\DriveCom.exe /drive=F /action=SetBootMode
```

После этого опять нужно воспользоваться утилитой `DriveCom`, которой надо будет передать букву нашего флеш-драйва, путь до burner image-а и путь к файлу, в который будет сохранена оригинальная сдампленная прошивка. Выглядеть это будет так:

```
tools\DriveCom.exe /drive=F /action=DumpFirmware /burner=BN03V104M.BIN /
```

Если ты все сделал правильно, то исходная прошивка сохранится в файл `fw.bin`.

Подготавливаем payload

Теперь настало время подумать о том, какой функционал мы хотим получить от нашей флешки. Если вспомнить Teensy, для него есть отдельный тулкит Kautilya, который позволяет автоматизировать создание пейлоадов. Для USB Rubber Ducky тут есть целый сайт, позволяющий посредством удобного веб-интерфейса прямо в онлайне создавать скрипты для девайса по своему вкусу. И это помимо списка уже готовых скриптов, которые лежат на гитхабе проекта. На наше счастье, Ducky-скрипты можно сконвертировать в бинарный вид, чтобы затем встроить их в прошивку. Для этого нам пригодится утилита Duck Encoder.

Что же по поводу самих скриптов, то тут есть сразу несколько вариантов:

- можно набросать нужный скрипт самостоятельно, благо используемый синтаксис не сложен в освоении (см. официальный сайт проекта);
- воспользоваться уже готовыми вариантами, выложенными на гитхаб, благо там есть и reverse shell, и прочие плюшки — остается только подправить и сконвертировать в бинарный вид;
- либо же воспользоваться вышеупомянутым сайтом, который в пошаговом режиме проведет через все настройки и позволит скачать готовый скрипт в виде Ducky-скрипта (либо уже в сконвертированном бинарном виде).

INFO

При использовании Ducky-скриптов следует помнить, что команда DELAY, выполняюшая задержку на указанное число миллисекунд, на флешке будет работать несколько иначе, чем на Rubber Ducky, поэтому время задержки придется поднастраивать.

Для того чтобы перевести скрипт в бинарный вид, необходимо выполнить следующую команду:

```
java -jar duckencoder.java -i keys.txt -o inject.bin
```

где `keys.txt` — Ducky-скрипт, а `inject.bin` — выходной бинарник.

Заливаем прошивку

Как только у нас на рукам появится готовый пейлоад, настанет время внедрять его в прошивку. Выполняется это следующими двумя командами:

```
copy CFW.bin hid.bin
tools\EmbedPayload.exe inject.bin hid.bin
```

Обрати внимание, что сначала прошивка копируется в `hid.bin`, и только затем перепрошивается. Делается это так потому, что пейлоад можно вннедрить в прошивку только один раз, поэтому оригинальный `CFW.bin` надо сохранить нетронутым.

После такой манипуляции у нас на руках будет файл кастомной прошивки `hid.bin` с внедренной в него полезной нагрузкой. Остается только залить полученную прошивку на флешку:

```
tools\DriveCom.exe /drive=F /action=SendFirmware /burner=BN03V104M.BIN /
```

где `F` — опять же, буква накопителя.

Альтернативные варианты

Помимо использования HID-природы флешки и превращения ее в клавиатуру, набирающую наши пэйлоады, можно сотворить еще несколько трюков. Например, можно создать на устройстве скрытый раздел, уменьшив место, которое будет видеть ОС. Для этого сначала надо получить размер устройства в логических блоках:

```
tools\DriveCom.exe /drive=E /action=GetNumLBAs
```

Затем в папке `patch` нужно найти файл `base.c`, раскомментировать строку `#define FEATURE_EXPOSE_HIDDEN_PARTITION` и добавить еще одну директиву `define`, задающую новое число LBA: `#define NUM_LBAS 0xE6C980UL` (это число должно быть четным, так что если на предыдущем шаге ты получил, скажем, `0xE6C981`, то можно уменьшить число до `0xE6C940`, например).

После правки исходников, надо поместить прошивку, которую ты хочешь пропатчить, в папку `patch` под именем `fw.bin` и запустить `build.bat`, который создаст в `patch\bin\` файл модифицированной прошивки `fw.bin`. Останется только залить его на флешку.

Аналогичным образом делается Password Path и No Boot Mode Patch, про которые ты можешь подробней посмотреть на гитхабе проекта. Моей же основной целью было научить флешку выполнять заданные действия, чего мы с тобой и добились.

Итог

Поставленной цели мы добились. Более того: думаю, ты теперь понял, что флешки (да и прочие USB-девайсы) нельзя больше рассматривать как просто абстрактный накопитель, хранящий твою информацию. На самом деле — это уже практически компьютер, который можно научить выполнять определенные действия. Хотя на данный момент PoC выложен только для одного конкретного контроллера, будь уверен, что в момент чтения статьи кто-то наверняка ковыряет другие.

Так что будь осторожен при подключении USB-устройств и держи ухо востро.

WWW

Для того, чтобы проверить, какой контроллер установлен на флешке, можно воспользоваться утилитой usbflashinfo.

Теги: Rubber Ducky USB Выбор редактора Статьи

← Ранее
Компания Zimperium обнародовала эксплоит для уязвимости Stagefright

Далее →
Windows 10 загружается на компьютеры без разрешения пользователей

Антон Жуков

Редактор рубрики «ВЗЛОМ»

ДРУГИЕ СТАТЬИ В РУБРИКЕ PHREAKING

Подмена прошивки МФУ Epson позволяет атаковать компанию через факс-модем устройства

2 недели назад

Флешку для «убийства» компьютера теперь можно приобрести за 50 евро

Сен 9, 2016

Модифицированный Ethernet адаптер способен похищать данные с заблокированных ПК и Mac

Сен 8, 2016

Малварь RIPPER, вероятно, использовали при ограблениях банков на Тайване и в Таиланде

Авг 29, 2016

Ключи миллионов авто можно подделать с помощью дешевого RF-трансивера на базе Arduino

Авг 12, 2016

Исследователь нашел бреши в платежной системе Samsung Pay, но Samsung все отрицает

Авг 11, 2016

2 комментария

frolpaxa

Ноя 29, 2015 at 1:48 пп

https://www.youtube.com/watch?v=jfIkrzxhAXg

Ответить

AqulaGamer

Мар 27, 2016 at 8:35 дп

А можно ссылку на …»мы делали в журнале обзор девайсов, которые было бы желательно иметь в своем чемоданчике хакера»

Ответить

Оставить мнение

Ты залогинен как ecp_itsec (выйти)

Комментарий

ОТПРАВИТЬ

Уведомлять меня о новых комментариях

Последние взломы

Хакерская группа StrongPity распространяет вредоносные версии WinRAR и TrueCrypt

15 часов назад

Исследователи нашли шифровальщика, написанного на Go, и уже его взломали

16 часов назад

JavaScript-вредонос выключает ПК, если кто-то пытается завершить его процесс

18 часов назад

Колумнисты «Хакера»

Мифы о файловой системе F2FS. Колонка Евгения Зобнина

2 дня назад

Играем в панели уведомлений, вводим PIN-код взмахами и превращаем смартфон в 3D-пульт. Колонка Евгения Зобнина

3 недели назад

Новые угрозы для старых PoS-терминалов. Колонка Дениса Макрушина

4 недели назад

Последние новости

13 часов назад

На территории Турции заблокировали Dropbox, OneDrive, Google Drive и GitHub

15 часов назад

Хакерская группа StrongPity распространяет вредоносные версии WinRAR и TrueCrypt

16 часов назад

Исследователи нашли шифровальщика, написанного на Go, и уже его взломали

17 часов назад

MITRE предлагает $50 000 за работающий метод обнаружения вредоносных IoT-устройств

18 часов назад

JavaScript-вредонос выключает ПК, если кто-то пытается завершить его процесс

Три индийских колл-центра зарабатывали на скаме более $75 млн в год

2 дня назад

43 разновидности продуктов на рынке ИБ. Колонка Александра Полякова

Авг 29, 2016

Вопросы и помощь по сайту, подписке, журналу: support@glc.ru
Отдел рекламы и спецпроектов: yakovleva.a@glc.ru
Контент 18+