Оглавление

19 Самостоятельный поиск уязвимостей

К сожалению, у хакинга нет волшебной формулы. Технологии постоянно развиваются, и их слишком много, поэтому я не могу объяснить каждый метод поиска ошибок. Поэтому я собрал сведения о методологиях, которым следуют успешные охотники за уязвимостями. Эта глава познакомит вас с базовым подходом ко взлому любого приложения. Материал систематизирован благодаря беседе с успешными хакерами, чтению блогов, просмотру видеороликов и, собственно, хакингу.

Когда вы только начинаете заниматься взломом приложений, свои успехи лучше всего оценивать по тем знаниям и опыту, которые вы получаете, а не по найденным вами уязвимостям или полученным вознаграждениям. Если ваша цель — выявлять уязвимости в престижных программах Bug Bounty, создавать как можно большее количество отчетов или просто зарабатывать деньги, то начало пути покажется трудным. Известные программы от таких компаний, как Uber, Shopify, Twitter и Google ежедневно проверяют опытные хакеры, поэтому там остается очень мало невыявленных проблем. Сосредоточьтесь на приобретении навыков, распознавании закономерностей и исследовании технологий, и вы сможете сохраните оптимизм даже без видимых результатов.

Предварительное исследование

Присоединившись к любой программе Bug Bounty, проведите *предварительное исследование* приложения, чтобы лучше понимать, с чем имеете дело. Для начала ответьте на простые вопросы:

- Каков охват этой программы? Что она в себя включает: *.<example>.com или просто www.<example>.com?
- Сколько поддоменов принадлежит компании?
- Сколькими IP-адресами владеет компания?
- Это программное обеспечение или сервис? Доступен ли исходный код? Предназначен ли он для совместной работы? Что на сайте является платным?
- Какие применены технологии? На каком языке программирования сайт написан? Какую БД он использует? На каких фреймворках он основан?

Это лишь несколько вопросов, которыми вы должны задаться, приступая к хакингу. В этой главе мы опишем приложение с неограниченным охватом ввода *.<example>.com. Выберите инструменты, которые могут работать в фоновом режиме, чтобы заниматься другими исследованиями в ожидании результатов. Но помните, если вы запустите их на своем компьютере, брандмауэр для веб-приложений вроде Akamai может заблокировать ваш IP-адрес.

Чтобы этого избежать, создайте виртуальный выделенный сервер (virtual private server, или VPS) на облачной платформе, которая разрешает использовать свои системы для проведения проверок безопасности. Обязательно ознакомьтесь с правилами предоставления услуг вашего облачного провайдера, так как некоторые компании запрещают такого рода деятельность (например, на момент написания этой главы для проведения проверок безопасности в сервисе AWS требуется отдельное разрешение).

Составление списка поддоменов

Исследование можно начать с поиска поддоменов с помощью VPS. Чем больше поддоменов вы найдете, тем большей будет поверхность вашей атаки.

Используйте утилиту SubFinder: она написана на Go и имеет высокую производительность. SubFinder загрузит записи о поддоменах сайта с разных источников, включая центры сертификации, поисковые системы, Internet Archive Wayback Machine и пр.

Иногда такой подход позволяет найти не все поддомены, однако с обнаружением тех из них, у которых есть SSL-сертификаты, обычно не возникает проблем, так как журналы прозрачности сертификатов содержат все необходимые записи. Например, если сайт регистрирует сертификат для test.<example>.com, этот поддомен, скорее всего, существует, по крайней мере на момент регистрации. В то же время сайт может зарегистрировать сертификат для открытого поддомена *.<example>.com. В таком случае вам, возможно, удастся подобрать лишь некоторые доменные имена.

К счастью, SubFinder поддерживает подбор поддоменов на основе списка распространенных слов. Этот список находится в репозитории GitHub под названием SecLists (Приложение А). Еще один полезный список опубликовал Джейсон Хэддикс: gist.github.com/jhaddix/86a06c5dc309d08580a018c66354a056/.

Если вам нужно просто узнать, был ли зарегистрирован wildcard-сертификат, и вы не хотите использовать SubFinder, можете зайти на сайт crt.sh. Если такой сертификат обнаружится, его хеш можно найти на сайте censys.io. Обычно crt.sh предоставляет прямую ссылку на censys.io для каждого сертификата.

Составив список поддоменов для *.<example>.com, можете просканировать порты и сделать снимки экрана с найденными сайтами. Но прежде чем двигаться дальше, подумайте о том, стоит ли искать домены более низких уровней. Например, если сайт зарегистрировал SSL-сертификат для *.corp.<example>.com, в этом поддомене, скорее всего, есть другие поддомены.

Сканирование портов

Сканирование портов позволяет определить дополнительные поверхности для атаки, включая активные сервисы. С помощью этого подхода Энди Гилл нашел на сайте Pornhub уязвимый сервер Memcache и получил за это 2500 долларов (глава 18).

Результаты сканирования портов могут также дать представление о защищенности сайта. Если на сайте закрыты все порты, кроме 80 и 443 (стандартные

веб-порты для HTTP и HTTPS), это говорит о серьезном отношении к безопасности. Множество открытых портов свидетельствует о потенциальных уязвимостях.

Nmap и Masscan являются популярными инструментами для сканирования портов. Nmap более старый и без оптимизации может демонстрировать низкую производительность. Его преимущество состоит в том, что вы можете передать ему список URL-адресов, и он сам определит, какие IP нужно сканировать. К тому же Nmap имеет модульную структуру и позволяет во время сканирования выполнять другие проверки, среди которых поиск названий файлов и директорий (скрипт http-enum). Инструмент Masscan отличается высокой скоростью и лучше сканирует готовый список IP-адресов. Я применяю его для поиска стандартных открытых портов (80, 443, 8080, 8443 и т. п.) и использую полученные результаты для снимков экрана.

Обращайте внимание на IP-адреса, принадлежащие найденным поддоменам. Если все они, за исключением одного, находятся в одном и том же диапазоне IP-адресов (которым, к примеру, владеет AWS или Google Cloud Compute), этот поддомен стоит исследовать подробней. Нетипичный IP-адрес может указывать на самописное или стороннее приложение, которое, возможно, защищено хуже, чем основные продукты компании, находящиеся в общем диапазоне. Франс Розен и Роян Риял воспользовались сторонними сервисами для захвата поддоменов, принадлежавших компаниям Legal Robot и Uber (глава 14).

Создание снимков экрана

Снимки веб-страниц визуализируют охват программы, демонстрируя новые закономерности. Обращайте внимание на сообщения об ошибках от сервисов, через которые уже происходил захват поддоменов. Приложение, использующее сторонние сервисы, может со временем поменяться, а его DNS-записи могут остаться прежними (главе 14). Захват такого сервиса злоумышленником будет иметь серьезные последствия для приложения и его пользователей. Но даже если сообщения об ошибках отсутствуют, снимки экрана могут раскрыть зависимость поддомена от стороннего сервиса.

Дальше можно поискать конфиденциальные данные. Например, если все поддомены, кроме одного, найденные в зоне *.corp.<example>.com, возвращают страницу 403 «Access Denied», а необычный поддомен содержит форму входа на

подозрительный веб-сайт, это может стать причиной нестандартного поведения сайта. Внимательно рассмотрите страницы, которые выводятся по умолчанию сразу после установки приложения, формы для входа администраторов и т. д.

И наконец, обращайте внимание на приложения, которые выделяются на фоне других поддоменов сайта. Например, если на всех поддоменах компании используется Ruby on Rails, и только на одном PHP, сосредоточьтесь на этом PHP-приложении, так как для разработчиков сайта этот язык, по всей видимости, не является основным. Важность приложения, найденного в одном из поддоменов, сложно определить без предварительного исследования, но это может быть хорошим шансом получить высокое вознаграждение. Жасмин Лэндри использовал полученный им SSH-доступ для удаленного выполнения кода (главе 12).

С созданием снимков экрана могут помочь несколько инструментов. Сейчас я использую HTTPScreenShot и Gowitness. Утилита HTTPScreenShot имеет два преимущества: во-первых, вы можете передать ей список IP-адресов, и она создаст снимки экрана не только для них, но и для других поддоменов, связанных с соответствующими SSL-сертификатами. Во-вторых, она разбивает результаты по категориям в зависимости от кодов состояния (например, 403 или 500), систем управления содержимым, которые в них используются, и других факторов. Она также включает в свои результаты найденные HTTP-заголовки, что тоже полезно.

Утилита Gowitness быстрая и легковесная. Я использую ее в случаях, когда у меня есть список URL-адресов вместо IP. Она тоже предоставляет заголовки, полученные при создании снимков экрана.

Aquatone тоже заслуживает упоминания, хотя я не использую этот инструмент. Он недавно был переписан на Go и среди прочих возможностей поддерживает кластеризацию и простой экспорт результатов в разные форматы, совместимые с другими инструментами.

Обнаружение содержимого

Исследовав и визуализировав найденные поддомены, займитесь поиском интересного содержимого. К этому этапу можно подойти по-разному. Можно найти файлы и директории, подбирая их имена. Успешность этого подхода зависит

от списка слов, который вы используете (хорошие списки содержатся в репозитории SecLists, например raft-*). Или отследить результаты, полученные на этом этапе, формируя собственные списки на основе часто находимых файлов.

Исследуйте составленный список с именами файлов и директорий. Лично я для этого применяю Gobuster и Burp Suite Pro. Gobuster — это гибкая и быстрая утилита для подбора имен, написанная на Go. Если указать ей домен и список слов, она проверит существование соответствующих файлов и директорий и подтвердит ответ сервера. Следует также упомянуть утилиту Meg, разработанную Томом Хадсоном, тоже написанную на Go, которая позволяет проверять разные пути сразу на нескольких сайтах. Она подходит для ситуаций, когда вы нашли много поддоменов и хотите одновременно выполнить поиск содержимого в каждом из них.

Пропуская трафик через пакет Burp Suite Pro, я использую либо встроенный в него инструмент для обнаружения содержимого, либо Burp Intruder. Встроенный инструмент имеет гибкие настройки и позволяет работать как с пользовательским, так и с собственным списком, искать файлы с определенными расширениями, определять глубину вложенности папок и т. д. Если же выбрать второй вариант, запрос, направленный сайту, сначала попадает в Burp Intruder, а передаваемые данные (то есть список) добавятся к корневому пути. После этого начнется атака. Обычно я сортирую полученные результаты по размеру или состоянию ответа, в зависимости от реакции приложения. Если я нахожу интересную папку, то запускаю Intruder еще раз, чтобы поискать содержащиеся в ней файлы.

Дополнительно можете воспользоваться поиском Google, как это сделал Бретт Буэрхаус (глава 10). Поиск Google может сэкономить время, особенно если найденные URL-адреса включают в себя параметры, которые часто оказываются уязвимыми, — например, `url`, `redirect_to`, `id` и т. д. Сайт Exploit DB ведет базу поисковых запросов для разных ситуаций: https://www.exploit-db.com/google-hacking-database.

На сайте GitHub вы можете найти открытый исходный код или полезную информацию о технологиях, которые использованы на сайте. Именно так Михель Принс обнаружил уязвимость с удаленным выполнением кода на сайте Algolia (глава 12). Для проверки GitHub-репозиториев на предмет секретных ключей приложений и других конфиденциальных данных можно применить утилиту Gitrob. В дополнение к этому вы можете проанализировать

репозитории с кодом приложения и сторонние библиотеки, от которых оно зависит. Программа Bug Bounty может охватывать как заброшенный проект, так и уязвимость во внешнем репозитории. Репозитории с кодом также могут дать представление об исправлении предыдущих уязвимостей, что актуально для таких компаний, как GitLab, которые открывают код своих приложений.

Ранее обнаруженные уязвимости

Ресурсами для анализа предыдущих уязвимостей могут послужить статьи хакеров, раскрытые отчеты, база данных общеизвестных уязвимостей информационной безопасности (CVE), опубликованные эксплойты и т. д. Как отмечается на протяжении всей книги, сам факт обновления кода вовсе не означает, что все уязвимости были исправлены. Обязательно проверяйте любые изменения. Исправление подразумевает добавление нового кода, который может содержать ошибки.

За обнаружение уязвимости на сайте Shopify Partners Таннер Эмек получил 15 250 долларов (глава 15). Он применил анализ отчета об ошибке, раскрытый ранее.

Итак, мы рассмотрели все основные области предварительного исследования. Теперь пришло время поговорить о проверке приложения. Но предварительное исследование на этом не заканчивается. Оно является неотъемлемой частью процесса поиска уязвимостей. Приложения постоянно улучшаются, поэтому вы всегда можете посетить уже проверенные сайты и посмотреть, что в них поменялось.

Тестирование приложений

Методология и технические приемы тестирования зависят от типа приложения. Я сделал общий обзор факторов, о которых нужно помнить, и мыслительных процессов, стоящих за проверкой нового сайта. Но, независимо от того, что именно вы тестируете, нет лучшего совета, чем тот, который дал Маттиас Карлссон: «*Не думайте, что все уже проверено и больше не на что смотреть. Подходите к каждому случаю так, словно до вас им еще никто не занимался. Ничего не нашли? Двигайтесь дальше*».