Механизмы поиска уязвимостей

Михаил СУХОВ,
руководитель отдела анализа защищенности Angara Security

Поиском «слабого звена» в продуктах ведущих ИТ- и ИБ-разработчиков чаще всего занимаются энтузиасты, специально выделенные сотрудники компании или злоумышленники. У каждого большой опыт обнаружения недостатков безопасности и высокий уровень компетентности. Некоторые компании готовы платить энтузиастам за найденные уязвимости в своих продуктах. Такой поиск отличается от процесса обнаружения «дыр» в системе ИБ конкретного заказчика. Рассмотрим общепринятые механизмы организации процедуры и устранения уязвимостей.

В мире ИТ и информационной безопасности поиск уязвимостей – это, как правило, непрерывный процесс, поскольку инфраструктура компаний постоянно меняется. Процедура направлена на выявление слабых мест в программных продуктах либо информационных системах, которые представляют собой совокупность программных комплексов, потенциально интересных злоумышленникам для нанесения ущерба. Подходы к поиску уязвимостей в продуктах ведущих ИТ- и ИБ-разработчиков и поиску «дыр» в конкретной системе различаются. Основное различие заключается в составлении модели нарушителя, определении, какие доступы есть у злоумышленника в системе.

При наличии готового продукта можно запустить его в своей виртуальной среде – «песочнице», – чтобы получить полный контроль над приложением, посмотреть системные вызовы (strace, procmon) и трафик. Особое внимание стоит обратить на используемые протоколы и содержимое сетевых пакетов. Представляет интерес также обращение к базам данных и другим служебным процессам и сервисам. В некоторых случаях, если повезет, можно получить доступ к исходному коду. Если применяются интерпретируемые языки программирования, код будет сразу в открытом и удобочитаемом виде. Но если повезло меньше, то извлекать код придется путем декомпиляции исполняемых файлов, т. е. проводить реверс-инжиниринг. Вторая ситуация представляется несколько более сложной, поскольку код может быть дополнительно обфусцирован, стать неочевидным, запутанным и плохо читаемым.

После получения информации с тестового стенда будет намного проще искать сложные уязвимости и возможные «точки входа», например, места обработки вводимой пользователем информации. Также на этом стенде исполнитель может использовать различные программы для статического или динамического анализа кода приложения, фазеры для поиска уязвимостей с высоким количеством запросов в секунду и т. д. Помимо прочих преимуществ при возникновении непоправимых изменений, таких как удаление или повреждение критически важных файлов, всегда можно будет вернуть систему в исходное состояние за несколько действий. Таким образом, возможность исследования готового продукта на собственном стенде дает свободу действий и позволяет выявить значительно больше уязвимостей.

Большинство CVE (Common Vulnerabilities and Exposures) найдено благодаря использованию различных фазеров. Они бывают трех видов: black-box, gray-box и white-box. Различаются между собой необходимостью доступа к исходному коду приложения. Так, если для black-box-фазера не нужны исходные коды и исполняемые файлы (он работает только с вводом и выводом приложения), то для gray-box необходим доступ к бинарным файлам приложения, а в случае с white-box еще и к исходному коду. Последние два подхода позволяют оценивать покрытие тестовыми данными исходного кода (количество веток, по которым прошелся фазер), находить незначительные утечки памяти и переполнения буфера, что впоследствии помогает обнаружить скрытые уязвимости в автоматическом режиме. Этот процесс не только используется сторонними специалистами для поиска уязвимостей в коде, но и чаще всего внедрен в процессы

разработки большинства крупных компаний, а некоторые, например, Google, имеют собственные разработки для осуществления фазинга приложений.

В последнее время прослеживается тенденция поиска уязвимостей не в самом продукте, а в обновлениях к нему. Злоумышленники сравнивают две версии, чтобы выявить изменения в приложении с обновлением. Существуют несколько сценариев поиска:

- уязвимость в новом функционале, который добавился в обновлении;
- уязвимость в старой версии, которую закрыли в обновлении, но не опубликовали информацию о ней;
- недостаточное закрытие уязвимости, которая была опубликована.

Данный подход эффективно показывает себя не только в поисках уязвимостей в продуктах, но и в рамках программ bug bounty при анализе защищенности информационных систем. «Белые» хакеры всегда отслеживают изменения в информационных системах для поиска уязвимостей и получения выплат в рамках различных программ.

> «Белые» хакеры всегда отслеживают изменения в информационных системах для поиска уязвимостей и получения выплат в рамках различных программ.

Некоторые компании готовы платить энтузиастам за найденные уязвимости в своих продуктах. Например, у Google, NextCloud есть свои bug bounty-программы, где исследователям предлагают найти уязвимость в самом продукте компании и получить солидные вознаграждения (выплаты могут достигать миллиона рублей). Конечно, такой подход повышает безопасность продуктов и их репутацию у покупателей, развивает сообщество активистов по поиску уязвимостей, предлагает альтернативу продаже уязвимости на черном рынке. Другие компании покупают услуги внешних интеграторов для повышения безопасности своих продуктов, что также позволяет пользоваться внешней оценкой защищенности.

Что касается российского законодательства, то ФСТЭК издал методические рекомендации по безопасной разработке программного обеспечения. Например, от многих команд разработки требуется соответствие ОУД4 (оценочный уровень доверия). Международные стандарты, в частности, OWASP также предоставляют множество рекомендаций по выстраиванию процесса разработки для поиска и своевременного устранения уязвимостей. Кроме того, описываются рекомендации по обучению разработчиков и повышению их осведомленности – практика security champions.

Если рассматривать конкретные информационные системы в рамках анализа защищенности или пентестов, то важно, какая модель нарушителя рассматривается. Чаще всего принимается во внимание black-box, модель внешнего нарушителя, у которого нет знаний об атакуемой системе и учетных записях в ней, но имеется сетевой доступ к системе. Как правило, данная система находится в функционирующей (production) среде, и у исполнителей существует ограничение на деструктивность воздействий и количество запросов в секунду. Это накладывает ограничение на фазинг и полезные нагрузки, которые использует исполнитель.

> В рамках поиска уязвимостей чаще всего ищут неправильные настройки «коробочных» решений, слабые пароли, уязвимости «самописных» приложений.

В рамках поиска уязвимостей чаще всего ищут неправильные настройки «коробочных» решений, слабые пароли, уязвимости «самописных» приложений (которые разработаны заказчиками, обычно Web-приложения) и пр. В данном случае целью является обнаружение чаще всего не новых «проблемных мест» в решении вендоров, а неправильных настроек или старых версий с известными уязвимостями.

Для поиска уязвимостей в программных комплексах могут применяться различные методы:

- пентесты собственной командой (внутренние);
- аудиты собственной командой (внутренние);
- пентесты внешними подрядчиками;
- аудиты внешними командами;
- программы bug bounty;
- автоматизированное сканирование инфраструктуры;
- информация из открытых источников или аналитика трендовых уязвимостей;
- инциденты безопасности.

Все это укладывается в процесс управления уязвимостями, точнее, в первый этап – обогащение базы данных информации об уязвимостях в инфраструктуре. Практика показывает, что все перечисленные услуги дополняют друг друга и позволяют найти максимальное количество уязвимостей. Однако для этого необходимы зрелые процессы информационной безопасности в организации, что сейчас является большой редкостью, так как для этого требуется человеческий ресурс. В России разработаны методические рекомендации ФСТЭК, которые дают описание процесса управления уязвимостями, а также рекомендации по внедрению и регламенты по срокам устранения недостатков информационной безопасности.

- растянутость сроков исправления уязвимостей по причине нехватки человеческого ресурса и невстроенных процессов информационной безопасности в организациях. Как результат – отсутствие инструментов воздействия на ИТ со стороны ИБ.

Таким образом, многие уязвимости могут оставаться в инфраструктуре долгое время, пока не произойдет инцидент, что понижает уровень защищенности.

Нехватка квалифицированных кадров – проблема, актуальная как на российском, так и на международном рынке. Компании компенсируют ее с помощью различных инструментов: начиная с услуг консалтинговых компаний (от MSS до отдельных услуг) и заканчивая публичными программами bug bounty. В последнее время интерес к публичным программам bug bounty не только формирует площадку для роста начинающих специалистов в сфере наступательной безопасности, так как дает возможность легально потренироваться на реальных инфраструктурах и сайтах, но и предлагает альтернативные источники дохода состоявшимся профессионалам. В России даже многие государственные органы открывают публичные программы bug bounty.

> По нашей статистике, 40% уязвимостей не закрываются как следует в течение года.

Для устранения найденных в рамках различных мероприятий уязвимостей в организации должны быть прописаны сроки исправления недостатков, ответственные лица и последствия в случае несоблюдения сроков.

Наряду с этим стоит обратить внимание на проверку корректности закрытия уязвимостей. По нашей статистике, 40% уязвимостей не закрываются как следует в течение года, что обусловлено множеством причин. К наиболее распространенным относятся следующие:

- неправильная оценка руководством рисков найденных уязвимостей в инфраструктуре, чаще всего вследствие давления со стороны ИТ-подразделения;
- неправильное или неполное исправление уязвимостей, как правило, из-за отсутствия необходимых компетенций, халатности или нехватки человеческого ресурса;

Если говорить о новых тенденциях в области безопасности, то в последние годы наблюдается рост интереса к новым концепциям безопасности, одна из них – Zero Trust, основанная на принципе «не доверяй никому». Концепция активно внедряется в корпоративных системах и разработке продуктов ведущих ИТ-компаний, что стимулирует развитие новых методов поиска уязвимостей, нацеленных на нарушение цепочки доверия.

В отличие от традиционных моделей безопасности, где акцент делается на защите периметра сети (например, с помощью межсетевых экранов и VPN), Zero Trust предполагает, что угроза может исходить как изнутри, так и извне. Поэтому каждая попытка доступа к ресурсам должна быть проверена и авторизована. Внедрение Zero Trust требует времени и усилий, но это один из самых перспективных подходов к обеспечению информационной безопасности в условиях современного цифрового мира.

Использование машинного обучения и искусственного интеллекта (ИИ) для поиска уязвимостей также становится все более популярным. Технологии позволяют быстрее и точнее находить уязвимости, автоматизируя процесс анализа и сокращая количество ложных срабатываний. Применение ИИ снижает требования к квалификации исполнителей, которые будут перепроверять срабатывания анализаторов в ручном режиме. Это проще, чем анализировать «с нуля», и также позволяет обучать специалистов искать уязвимости самостоятельно.

В конечном итоге поиск и устранение уязвимостей в продуктах ведущих ИТ- и ИБ-разработчиков и системах конкретных заказчиков – это два фундаментально важных, но различных по сути процесса, которые играют ключевую роль в обеспечении информационной безопасности. Различия между ними заключаются не только в масштабе и методологиях, но и в последствиях, которые могут возникнуть при эксплуатации выявленных уязвимостей. Объединяющим фактором остается одна цель – обеспечение максимальной безопасности и защиты информации. Причем как в случае с продуктами крупных ИТ-компаний, так и с системами конкретных заказчиков успех зависит от своевременного выявления уязвимостей, их эффективного устранения и постоянного совершенствования мер защиты. ■