1 История угроз для цепочек поставок ПО

В этой главе раскрываются такие темы, как мотивы злоумышленников, анатомия атак на цепочку поставок ПО и некоторые яркие примеры таких атак. Начнем с обсуждения мотивов, которые подталкивают хакеров совершать нападения именно на цепочки поставок.

Мотивы злоумышленников

Традиционные виды защиты более уязвимы для атак на цепочки поставок, что очень привлекает хакеров. Компании вкладывают значительные средства в создание межсетевых экранов, предотвращение взломов и управление доступом. Все эти виды защиты применяются против атак в стиле «передачи» (push), непосредственно нацеленных на инфраструктуру компании. Что же касается атак на цепочки поставок, то они действуют по схеме «получения» (pull), когда легитимные пользователи информационной технологии запрашивают уже зараженные программные обновления, непреднамеренно подвергая опасности всю компанию. Поскольку запрос в этом случае происходит от доверенного пользователя, находящегося внутри корпоративной среды, или поступает от иной доверенной сущности, уже одобренной системой управления рисками, получаемые обновления также считаются доверенными. По факту компании компрометируют сами себя.

При изучении средств для защиты от атак становится ясно, что одного уровня недостаточно. По аналогии с тем, как сетевые администраторы осознали потребность мониторинга исходящего трафика или реализации

контроля на стороне хоста, нам также крайне важно углублять защиту, учитывая факторы, выходящие за внешний периметр. Из-за особенностей облачной и мобильной инфраструктур, а также инфраструктуры социальных сетей и современных приложений понятие внешнего периметра необходимо расширить. Теперь его следует рассматривать как разделительный слой между зонами доверия независимо от того, находятся они на границе вашей сети или выступают логическим барьером внутри вашего приложения либо механизма контроля доступа. Таким образом, при использовании средств контроля и моделировании угроз необходимо учитывать всю поверхность атаки, анализируя каждую точку взаимодействия и доверительную связь.

Атаки на цепочки поставок также выступают в роли своеобразных усилителей, то есть способны оказать воздействие на множество других субъектов, связанных с этой целью через цепочку поставок, тем самым умножая ущерб и разрушения, вызванные атакой. Выявив ключевые зависимости или широко используемое ПО, атакующий может попытаться внедрить вредоносный код в любую среду, которая этот код использует. Подобные случаи происходят регулярно. Принцип работы похож на атаку типа «водопой», когда злоумышленник взламывает сайт, широко используемый целевой группой, например форум, посвященный обсуждению программируемых логических контроллеров, которым пользуются интеграторы промышленных систем управления. Если атакующему удастся скомпрометировать каждого посещающего форум пользователя, он теоретически может получить доступ к любой сущности критической инфраструктуры, на которую этот интегратор работает. Аналогичным образом любое ПО, используемое этими скомпрометированными интеграторами, может внести нежелательную или вредоносную функциональность в среду, где установлено это ПО, даже если человек к тому времени уже будет занят другим проектом.

Все это означает, что атаки на цепочки поставок являются для хакеров крайне привлекательными. В кибершпионаже присутствует экономическая составляющая, которая сильно выигрывает от повторно используемых эксплойтов и низких базовых затрат при реализации атак на цепочки поставок. Кроме того, многие недавние атаки включали использование вирусов-вымогателей, внедряемых через цепочки поставок. И это не только упрощает взлом, но также несет прямую финансовую выгоду для хакера и повышает уровень беспокойства среди организаций, в чей бизнес произошло вмешательство.

Модели угроз

О моделировании угроз часто говорят, но, как показывает опыт, мало какие организации этим реально занимаются. В индустрии принято обсуждать угрозы в весьма обобщенной форме, не разбирая даже контекста ПО или систем, необходимого для правильного их моделирования. Тем не менее в основе этих действий лежат определение поверхности атаки — точек взаимодействия — и изучение возможных проблем.

Для максимальной ясности определим несколько ключевых терминов.

Угроза. Негативное событие, ведущее к нежелательному результату. Угрозы могут быть по своей природе как естественными и незлонамеренными, так и откровенно вредоносными. В качестве примеров можно привести невозможность платежной системы вашего бизнеса регистрировать записи о транзакциях или разрушение дата-центра ураганом.

Источник угрозы. Сущность, ответственная за возникновение угрозы. Примерами являются хакеры, внутренние злоумышленники, враждебно настроенный партнер по бизнесу, скомпрометированный консультант и даже метеосводка. Было отмечено, что источник угрозы нередко определяется ошибочно, в результате чего может выбираться неподходящая модель защиты. Например, если вы считаете, что угроза исходит из страны, известной склонностью к распространению вирусов-вымогателей, но по факту это оказывается страна, ориентированная на шпионаж, как это повлияет на вашу схему защиты? Имеет ли значение фактическая страна-источник угрозы или же это просто создает излишнюю предвзятость?

Реализация угрозы. Действие, предпринимаемое источником угрозы, в конечном счете ведущее к ее возникновению. Например, источник угрозы, скажем хактивист, может подкупить вашего системного администратора, чтобы тот перенастроил платежную систему. Это создаст угрозу, которая в итоге повлияет на финансы компании.

Модель угрозы. Процесс документирования систем и угроз таким образом, чтобы можно было моделировать определенные виды решений, связанные с управлением рисками для системы из-за угрозы. Моделирование угроз преследует различные задачи, включая общее управление киберрисками, проектирование и анализ систем, а также модели обмена информацией.

Методы моделирования угроз

Существует несколько методологий моделирования угроз. Первой разберем методику под названием STRIDE.

STRIDE

STRIDE — это очень распространенная техника, позволяющая определить, что может пойти не так. Ее название представляет собой акроним, каждая буква которого означает категорию угроз.

Spoofing (спуфинг): выдача себя за пользователя или компонент системы с целью получить соответствующий ему доступ.

Tampering (подмена): изменение системы или данных, в результате чего они становятся менее полезными для своих пользователей.

Repudiation (отрицание): правдоподобное отрицание действий, предпринятых от лица конкретного пользователя или процесса.

Information Disclosure (разглашение информации): утечка конфиденциальной информации.

Denial of Service (отказ в обслуживании): приведение системы в состояние недоступности для конечных пользователей.

Elevation of Privilege (повышение привилегий): предоставление пользователю или процессу дополнительного уровня доступа к системе без авторизации.

Большинство атак на цепочки поставок являются результатом работы с системой, и традиционные модели вроде STRIDE спроектированы для противодействия прямым атакам на систему. В этом смысле они противоположны опосредованному влиянию, возникающему, когда администратор запрашивает очередное обновление приложения, которое до этого проблем не вызывало, но на этот раз оказывается вредоносным.

STRIDE-LM

Исследователи из Lockheed Martin расширили методику STRIDE, добавив в нее седьмое измерение, известное как *горизонтальное перемещение*. Несмотря на то что STRIDE хорошо подходит для проектирования систем,

эта техника не отвечает требованиям сетевой защиты. При этом STRIDE-LM предоставляет механизм, с помощью которого защитники реализуют схемы контроля, позволяющие более эффективно защищать систему уже после ее компрометации.

При оценке моделей угроз на предмет их возможного использования в атаках на цепочки поставок нужно задаваться вопросом: «Для чего модель проектировалась и как она вписывается в рассматриваемый сценарий?» Например, многие атаки на цепочки поставок реализуются через этапы обслуживания ПО посредством зараженных обновлений или же опираются на доверительные отношения, за счет чего обходят системы контроля, предназначенные для обнаружения первичной точки входа в программу. Также из-за единой точки отказа, характерной для цепочек поставок ПО, нижестоящее влияние таких действий нельзя эффективно учесть в оригинальной методологии STRIDE. По мере изучения примеров на протяжении книги вы начнете понимать, как использование горизонтального перемещения обеспечивает дополнительный контекст для моделирования и делает технику STRIDE-LM более пригодной для противодействия атакам на цепочки поставок.

Методика оценки рисков проекта OWASP

Методика проекта OWASP применяется в качестве модели количественной оценки конкретных рисков, используя критерии технической угрозы и влияния на бизнес. В основе данной методики лежит простая формула: *влияние* × *вероятность*. Самое же интересное в том, как создаются эти две составляющие. Ниже описывается, из чего конкретно строится эта формула, но здесь нужно учесть, что на практике иногда множители изменяются или же к наиболее значимым из них применяются веса. Например, в критической инфраструктуре мы видим, что многие сущности желают добавить к оценке влияния на бизнес пятый фактор «безопасности» и зачастую наделяют этот критерий бóльшим весом.

Вероятность = AVG (источник угрозы + уязвимость),

где *источник угрозы* = навык, мотивация, возможность и размер; а *уязвимость* = легкость обнаружения, удобство эксплуатации, осведомленность и обнаружение вторжения.

Влияние = AVG (технические последствия + бизнес-последствия),

где *технические последствия* = утрата конфиденциальности, целостности, доступности и отслеживаемости; а *бизнес-последствия* = финансовый ущерб, ущерб для репутации, несоответствие требованиям и конфиденциальность.

Показатель рисков = вероятность × влияние.

OWASP пригождается для оценки обнаруженного риска и определения приоритета реагирования на него, но эта метрика не так полезна для моделирования угроз с целью отслеживания неизвестных рисков. Внедрение OWASP в ваши процессы может иметь смысл *после* применения других техник моделирования угроз. По нашему опыту данный метод намного превосходит такие механизмы приоритизации, как общая система оценки уязвимостей (Common Vulnerability Scoring System, CVSS), но для эффективного применения требует наличия контекста.

DREAD

Техника DREAD — это наследие Microsoft. Мы редко встречали ее применение в современном цифровом мире. На сегодняшний день она считается «мертвой» методологией. Ее название можно разбить также, как STRIDE.

Damage (ущерб): насколько серьезной будет атака?

Reproducibility (воспроизводимость): насколько легко будет воспроизвести атаку?

Exploitability (эксплуатируемость): насколько сложно будет запустить атаку?

Affected users (потенциальные пострадавшие): какое количество людей окажется подвержено атаке?

Discoverability (обнаруживаемость): насколько легко обнаружить угрозу?

Использование деревьев атак

Деревья атак (https://clck.ru/3CKMaw) позволяют наглядно прорисовать путь от момента нежелательных последствий (чтобы понять, как это произошло и каковы наиболее вероятные векторы атаки), а также создать ранжированный список методов контроля, которые нужно реализовать для предотвращения возникших последствий. На рис. 1.1 показан самый «дешевый»

путь, которым злоумышленник может открыть сейф. В данном случае будет разумным сделать упор на системы физического контроля, чтобы предотвратить доступ к сейфу в принципе, или же купить более дорогой сейф, устойчивый к подобным атакам. И хотя внутренние угрозы являются допустимым вектором атаки, из экономических соображений атакующий вряд ли пойдет подобным путем.

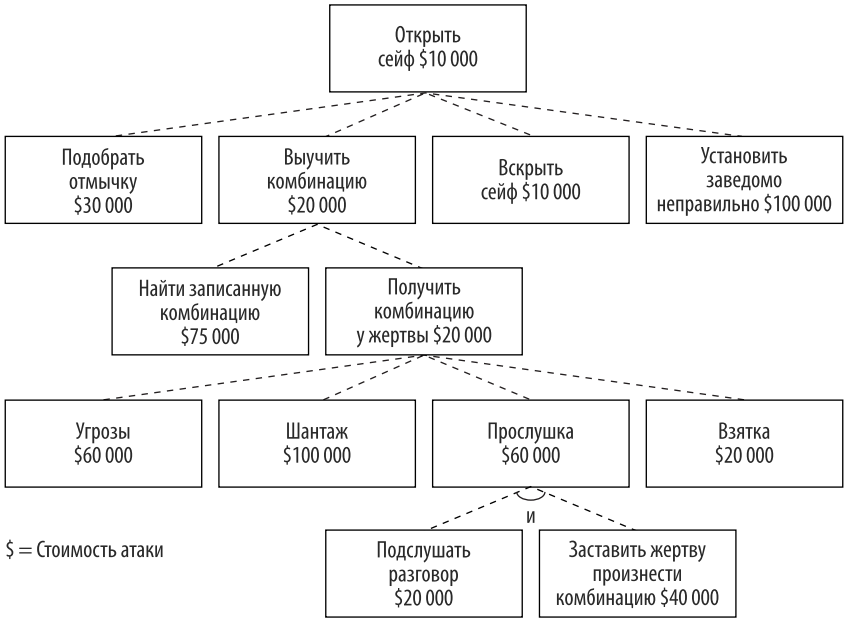

Рис. 1.1

Это хороший подход для изучения принципов атак на цепочки поставок и осознания того, насколько легко их реализовывать в сравнении с более традиционными атаками. В случае последних половина успеха заключается в получении физического доступа, необходимого для обхода системы контроля доступа. В случае же атаки на цепочку поставок в действиях, необходимых для взлома системы, участвует доверенная сущность.

В таком сценарии приходится преодолеть гораздо меньше преград. Разберем пример атаки, построенной по принципу тайпсквоттинга (рис. 1.2). Тайпсквоттинг — это метод мошенничества, когда злоумышленники выкладывают на GitHub зараженные пакеты с именами, похожими на имена настоящих, маскируя тем самым вредоносный компонент под вид легитимной библиотеки. Имейте в виду — это неполный пример.

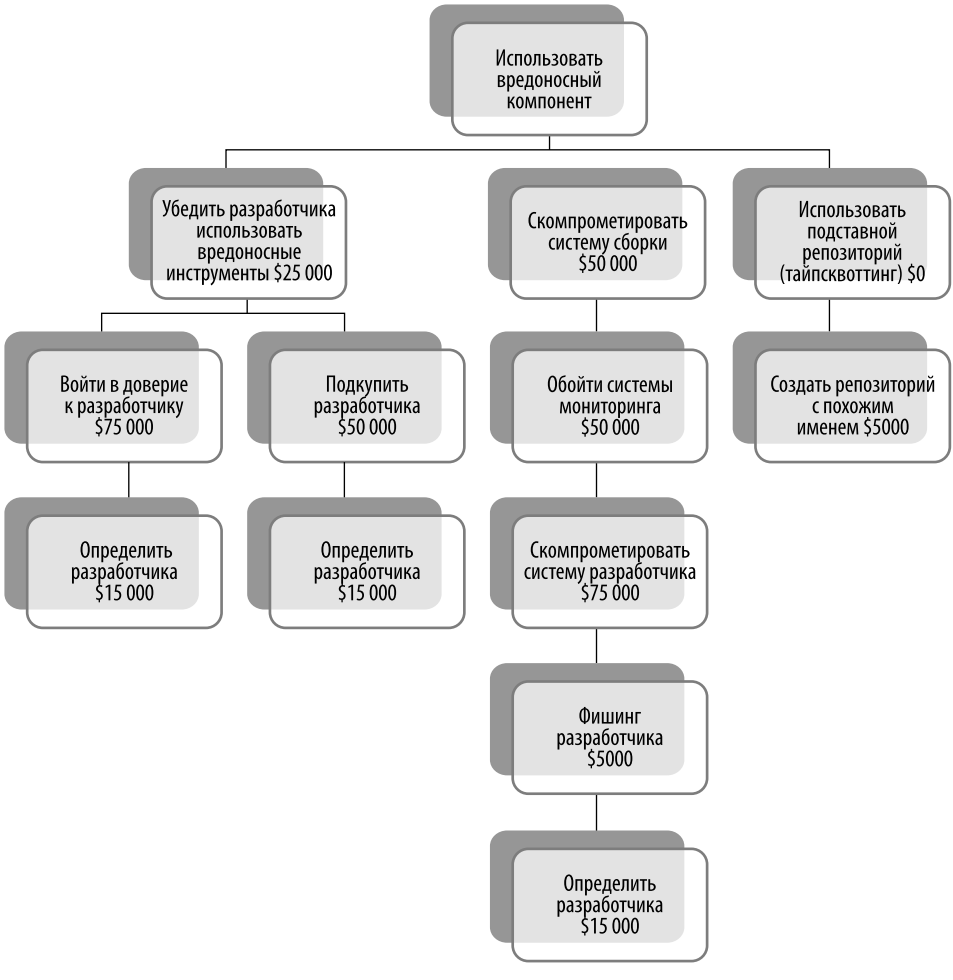

Рис. 1.2

Для сравнения подумайте, какие шаги потребуются, чтобы реализовать атаку. Исследователи из Lockheed Martin разработали методику, названную ими Cyber Kill Chain (https://clck.ru/3CKMbz), которая описывает этапы атаки. В этом процессе есть много нюансов, но на рис. 1.3 представлены только основные шаги.

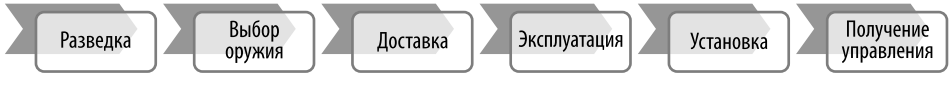

Рис. 1.3

Что, если злоумышленник может пропустить многие из этих шагов и перейти сразу к получению контроля? Для него окажется крайне заманчивым такой вариант, который позволит уменьшить затраты и сложность выполнения атаки, что вполне реально за счет анализа и понимания доступных путей.

Моделирование угроз

В этом разделе мы опишем типичный процесс моделирования угроз. Для начала вам нужно выбрать систему или приложение, которое вы создаете либо обновляете. Определите ее компоненты и то, каким образом они взаимодействуют друг с другом. Так вы начнете закладывать основу для определения поверхности атаки системы. Это может быть, к примеру, крайне востребованный API или HTTP-сервис. Возможно, в ней присутствуют сторонние зависимости вроде промежуточного ПО или сервера базы данных (БД), с которым необходимо взаимодействовать для выполнения логики приложения, или службы аутентификации, использующие интеграцию. Как бы то ни было, но для начала определения поверхности атаки вам потребуется фундаментальное понимание архитектуры.

В стандарте для проверки безопасности приложений OWASP ASVS (Application Security Verification Standard, https://clck.ru/3CKMdh) это понимание архитектуры определяется в качестве основного требования, включая обязательное моделирование угроз во время проектирования и внесения изменений.

Стандарт для проверки программных компонентов OWASP SCVS (Software Component Verification Standard, https://clck.ru/3CKMgN) расширяет данную концепцию, внося в нее список сторонних зависимостей, о чем мы поговорим позднее. Тем не менее становится ясно, что для понимания угроз и способов защиты от них необходимо уверенно ориентироваться в защищаемой системе.

Для документирования системы можно использовать различные инструменты. Один из типичных — это разработанный Microsoft инструмент моделирования угроз. Долгое время он являлся единственным эффективным решением, но индустрия не стоит на месте. Этот инструмент позволяет архитекторам ПО выявлять потенциальные угрозы на ранних стадиях процесса разработки еще до того, как их устранение станет затратным. Моделирование угроз в нем реализуется с помощью модели STRIDE.