Взламываем Хакера

Часть I

Учимся у экспертов борьбе с хакерами

Роджер А. Гримс

Перевод: @Samigg
http:\\skladchik.com

Предисловие

Роджер Граймс работал в индустрии компьютерной безопасности почти три десятилетия, и мне посчастливилось знать его примерно половину этого времени. Он один из тех немногих профессионалов, которых я встречал, у кого безопасность в крови. Он обладает интуитивным пониманием предмета, которое, в сочетании с его богатым опытом поимки плохих парней и устранения уязвимостей защиты на корню, сделало его превосходным кандидатом для написания этой книги.

Роджер впервые начал писать для *InfoWorld* в 2005-ом, когда отправил email, критикующий работу автора статей по безопасности. Это была полная критики статья, в которой было столько аргументов, что мы немедленно попросили его внести свои правки в эту публикацию. С тех пор он написал сотни статей для *InfoWorld*, и во всех этих статьях демонстрируется любовь к предмету, а также психологическое понимание, как хакеров-злодеев, так и людей, которые от них защищаются. В своей еженедельной колонке "Security Adviser" (Советы по безопасности) для *InfoWorld* Роджер демонстрирует уникальный талант, он фокусируется на проблемах, которые действительно важны, а не на эфемерных угрозах или переоцененных новых технологиях. Его страсть к убеждению защитников безопасности и их руководства делать правильные вещи непоколебима, несмотря на печальную склонность столь многих организаций пренебрегать основами и выбирать новый продукт в красивой обертке.

В этой книге Роджер рассказывает про этичных хакеров, которые внесли существенный вклад в индустрию. Их неустанные усилия помогают держать оборону против растущей орды атакующих, чьи цели с годами перешли от баловства к постоянным кражам драгоценной интеллектуальной собственности и миллионов долларов у финансовых учреждений и их клиентов. Мы бесконечно обязаны этим людям. Предоставляя возможность высказаться таким людям, как Брайан Кребс, Доктор Дороти Деннинг и Брюс Шнайер, Роджер отдает должное их усилиям, составив увлекательный сборник, с помощью которого он не только делится информацией, но и развлекает аудиторию. Эту книгу должен прочитать каждый, кто заинтересован в информационной безопасности, и любой, кто стремится, во что бы то ни стало, обеспечить нашу безопасность.

Эрик Кнорр
Главный редактор, *InfoWorld*

Введение

Цель книги - оценить по достоинству людей из сферы компьютерной безопасности, рассказать об одних из лучших в мире «белых шляпах», специалистах по безопасности, преподавателей и писателей. Я надеюсь, что книга вас увлечет, и вы высоко оцените усилия, которые потребовалось приложить огромному количеству людей, чтобы создать тот фантастический мир компьютеров, в котором мы сегодня живем. Без всех этих прекрасных людей, которые сражаются на нашей стороне, против тех, кто пытается причинить нам вред, компьютеры, интернет и все, что с ними связано было бы невозможно. В этой книге пойдет речь про наших защитников.

Я хочу вдохновить всех, кто планирует связать свою профессию с компьютерами, рассмотреть карьеру в информационной безопасности. Я также хочу вдохновить начинающих хакеров, особенно тех, кто возможно не до конца понимает законность применения своих знаний, стать специалистами в информационной безопасности. Я доволен своей жизнью, которую я посвятил сражению с киберпреступниками и их вредоносными творениями. Я смог исследовать все, что меня интересует, как хакера и при этом я находился в правовом поле. То же самое делают десятки тысяч других людей. Информационная безопасность - одна из самых горячих и хорошо оплачиваемых профессий в любой стране. Мне очень понравилось работать в этой области, и возможно, вас она заинтересует не меньше.

Большая часть этой книги состоит из глав, в которых я кратко описываю, как осуществляется определенный вид взлома. После глав я привожу один или несколько эталонных примеров борьбы с этими уязвимостями от экспертов из этой области. Я попытался выбрать представителей из разных сфер: легенд индустрии, светил, и даже некоторых относительно неизвестных людей, которые, тем не менее, внесли выдающийся вклад. Я постарался выбрать разных академиков, корпоративных поставщиков, учителей, лидеров, писателей и честных сотрудников, живущих по всему миру. Я надеюсь читатели, желающие связать свою жизнь с информационной безопасностью, смогут найти ту же мотивацию, которую нашел я, чтобы помочь сделать компьютерную индустрию значительно безопаснее.

Сражайтесь за правое дело!

Глава 1. К Какому Типу Хакера Вы Относитесь?

Много лет назад я переехал в дом, к которому был пристроен прекрасный гараж. Он был идеален для парковки и защиты моего маленького фургона с лодкой. Прочно собранный, без единого сучка на бревнах. Проводка сделана профессионально, а окна были высокого качества и выдерживали порывы ветра со скоростью до 250 км/ч. Большая часть внутренней отделки была облицована ароматическим красным кедром, таким, которым плотники обычно облицовывают сундук или шкаф для одежды, чтобы от него хорошо пахло. Даже при том, что сам я не могу и гвоздь ровно забить, я сразу понял, что строитель знал, что делает, он заботился о качестве и уделял внимание деталям.

Через несколько недель после того, как я переехал, ко мне пришел чиновник из администрации, и сообщил, что гараж был незаконно построен много лет назад, что он был построен без необходимого разрешения, и что мне придется снести его или платить огромные штрафы за каждый день просрочки. Я позвонил в администрацию, чтобы оспорить это решение, поскольку этот гараж стоял здесь много лет и был продан мне вместе с домом. Бесполезно. Его нужно было немедленно снести. Штраф за один день был больше, чем я мог заработать, если бы продавал строительные материалы, постепенно разбирая гараж. В финансовом отношении, чем быстрее я его снесу и вывезу, тем лучше.

Я взял кувалду (по сути, толстый металлический молот для демонтажных работ) и в течение нескольких часов разнес гараж, превратив его в гору древесины и прочего строительного мусора. В тот момент я не оценил, что за считанные часы разрушил своими неумелыми руками то, на что квалифицированный мастер потратил недели, а возможно и месяцы.

Вопреки распространенному мнению, хакеры-взломщики - это скорее разрушители с кувалдами, а не мастера, которые могут что-то создать.

Если вам посчастливилось, и вы решили стать компьютерным хакером, то вам придется выбирать, будете ли вы стремиться к сохранению общего блага или согласитесь на более низкие цели. Хотите ли вы быть киберпреступником, преступающим закон, или праведным, могущественным защитником? Эта книга - доказательство того, что лучшие и самые умные хакеры работают на хорошей стороне. У них есть возможность тренировать свой ум, интеллектуально расти и не беспокоиться о том, что их могут арестовать. У них есть возможность работать на переводной информационной безопасности, завоевывать восхищение своих коллег, двигать человечество вперед во имя всего хорошего, и при этом хорошо

зарабатывать. Это книга о невоспетых героях, которые сделали нашу невероятную цифровую жизнь возможной.

ПРИМЕЧАНИЕ Хотя термины "хакер" и "взлом" могут применяться к кому-то или чьей-либо деятельности как в хорошем, так и в плохом смысле, в широком смысле эти термины почти всегда имеют негативный оттенок. Я понимаю, что хакеры могут быть хорошими или плохими, но ради экономии места я буду использовать эти термины без дополнительного уточнения. В зависимости от контекста вы сможете с легкостью определить о каком типе хакеров идет речь.

Большинство Хакеров Посредственны

К сожалению, почти все, кто пишет о хакерах-престниках, но при этом не имеет опыта во взломе систем, идеализируют их, показывают их как сверхумных, богоподобных, мифических личностей. Они могут разгадать любой пароль меньше, чем за минуту (особенно под дулом пистолета, если верить Голливуду), взломать любую систему, и взломать любую зашифрованную информацию. В основном они работают ночью и в огромных количествах пьют энергетики, засоряя свое рабочее место остатками картофельных чипсов и кексов. Школьник использует украденный у учителя пароль, чтобы изменить оценки, а СМИ рассыпаются перед ним бисером, будто он следующий Билл Гейтс или Марк Цукерберг.

Хакерам не нужно блистать умом. Я - живое тому доказательство. Даже при том, что я взламывал все и везде, куда меня только не нанимали, я никогда полностью не понимал квантовую физику или теорию относительности Эйнштейна. В старших классах я дважды завалил экзамен по английскому, у меня никогда не было выше тройки по математике, а мой средний балл в первом семестре колледжа был 0.62. Это результат пяти двоек и одной пятерки. Одинокая пятерка была за уроки безопасности на воде, потому что на тот момент я уже пять лет работал спасателем на побережье океана. Плохие оценки были не только из-за того, что я не старался. Я просто был недостаточно умен и не старался. Позже я понял, что зачастую, упорно заниматься важнее, чем просто родиться умным. В итоге я получил высшее образование и делал успехи в сфере информационной безопасности.

Тем не менее, даже когда автор не описывает плохих хакеров суперумными, читатель часто представляет их именно такими, потому что они, судя по всему, практикуют какую-то продвинутую черную магию, о которой не знает остальной мир. В мировом сознании "киберпреступник" и "суперумный" являются неотделимыми друг от друга. Это не правда. Некоторые из них умны, большинство - средние, а некоторые вообще не блещут умом, как и весь

остальной мир. Хакер просто знает некоторые факты и процессы, которых не знают другие, как, например, плотник, сантехник или электрик.

Защитники - это хакеры с плюсом

Если провести только интеллектуальное сравнение, то защитники, в среднем, умнее, чем атакующие. Защитник должен знать все, что делает хакер плюс он так же должен знать, как остановить атаку. Защита не поможет, если в ней будет задействован конечный пользователь, и если она не работает в фоне все время. Покажите вредоносного хакера, который использует определенную технику, и я покажу вам огромное количество защитников, которые умнее, чем он. Дело в том, что к атакующим привлекается больше внимания со стороны прессы. Эта книга - повод сравнять счеты.

Хакеры особенные

Даже при том, что я не причисляю всех хакеров к суперумным, хорошим или плохим, у них всех есть общие черты. Такие, как любопытство и желание опробовать работу инструментов за пределами доступного интерфейса или доступных границ. Они не боятся проложить свой собственный путь. Компьютерные хакеры - это обычно лайф-хакеры, взламывающие все возможное за пределами компьютеров. Они - тот тип людей, который, встречаясь с охраной аэропорта, тихо обдумывает, как можно незаметно пронести оружие в обход детекторов, даже при том, что у них на самом деле нет таких намерений. Они выясняют, можно ли подделать дорогостоящие билеты на концерт, даже если не собираются проходить бесплатно. Когда они покупают телевизор, им интересно, могут ли они получить доступ к его операционной системе, чтобы воспользоваться дополнительными преимуществами. Покажите мне хакера, и это точно будет тот, кто ставит под сомнение статус-кво и все время проводит различные исследования.

ПРИМЕЧАНИЕ В какой-то момент моя гипотетическая схема проноса оружия мимо охраны аэропорта заключалась в использовании поддельного инвалидного кресла, где оружие или взрывчатка были бы спрятаны в металлических частях. Инвалидные кресла часто проходят мимо охраны без тщательного досмотра.

Хакеры Настойчивы

Следующая после любопытства полезная черта хакера - это настойчивость. Любой хакер, хороший или плохой, хорошо знаком с многочасовой агонией в бесконечных попытках заставить что-то работать. Вредоносные хакеры ищут слабые места защиты. Одна ошибка защитника, в сущности, делает всю защиту бесполезной. Защитник должен быть идеальным. Каждый компьютер и программа должны быть пропатчены, каждая конфигурация должна быть верно настроена, а каждый конечный пользователь - идеально обучен. Или, по крайней мере, к этому нужно стремиться. Защитник знает, что применяемая защита может работать не всегда или не должным образом использоваться, поэтому он создает "глубокоэшелонированную оборону". И хакеры и защитники ищут слабые места, просто делают это с разных сторон системы. Обе стороны участвуют в длительной войне с множеством сражений, побед и поражений. Победит самая настойчивая сторона.

Хакеры носят шляпы

Я всю свою жизнь был хакером. Я мне платили за взлом систем (на это у меня были законные полномочия). Я взламывал пароли, сети и писал вредоносное ПО. При этом, я ни разу не нарушил закон или пересек черту. Это не означает, что не было людей, которые пытались подбить меня на это. На протяжении многих лет меня подбивали друзья, с просьбами взломать телефон супругов, которых они подозревают в измене, начальники, которые просили меня перехватывать почтовые сообщения своих начальников, или люди, которые просили меня взломать хакерский сервер злоумышленников (без соответствующего ордера), чтобы попытаться предотвратить их дальнейшие атаки. С самого начала вам придется решить, кто вы, и какие у вас ценности. Я решил, что буду хорошим хакером (хакером "в белой шляпе" (whitehat)), и хакеры в белой шляпе не совершают нелегальных или неэтичных действий.

Хакеры, которые охотно принимают участие в нелегальной или неэтичной деятельности называются "черными шляпами" (blackhat). Хакеров, которые известны в обществе, как белые шляпы, но в тайне балуются тем же, чем и черные шляпы, называют "серыми шляпами" (grayhat). В этом вопросе у меня бинарный моранльный код. Серые шляпы - это черные шляпы. Вы либо занимаетесь нелегальными делами, либо нет. Ограбьте банк, и я назову вас грабителем банков, независимо от того, что вы сделаете с деньгами.

Это не означает, что черные шляпы не могут стать белыми шляпами. Такое происходит постоянно. Для некоторых из них, вопрос лишь в том, успеют ли они стать белыми шляпами до того, как проведут немало времени в тюрьме. Кевин Митник (https://en.wikipedia.org/wiki/Kevin_Mitnick), один из самых знаменитых арестованных хакеров в истории (мы поговорим о нем в Главе 5), прожил долгую жизнь защитника, помогая общему делу. Роберт Т. Моррис, первый парень, который написал и выпустил компьютерного червя, парализовавшего интернет (https://en.wikipedia.org/wiki/Morris_worm), в конечном счёте стал лауреатом премии Ассоциации вычислительной техники "за вклад в компьютерные сети, распространенные системы и операционные системы".

Поначалу граница между легальным и нелегальным взломом не была так четко очерчена, как сегодня. На самом деле, большинство хакеров тех времен, которые работали нелегально, получили статус культовых супергероев. Даже я сам не могу не быть заинтересован в некоторых из них. Джон Дрейпер (также известный, как "Captain Crunch") использовал игрушечный свисток из коробки кукурузных хлопьев Cap'n Crunch, чтобы создать звук (на частоте 2600 Гц), который можно было использовать, чтобы получить бесплатный доступ в телефонную сеть дальней связи. Многие хакеры, которые обнародовали частную информацию для "общего блага" часто становятся известными. Но есть некоторые исключения, я никогда не поддерживал слишком идеализированное представление о киберпреступниках. Я всегда достаточно четко понимал, что люди, совершающие несанкционированные действия в отношении компьютеров других людей или их данных, совершают преступление.

Много лет назад, когда я только начинал интересоваться компьютерами, я прочитал книгу Стивена Леви: " *Хакеры: Герои компьютерной революции (Hackers: Heroes of the Computer Revolution)*". В расцветающий век персональных компьютеров Леви написал увлекательную историю о хакерах, хороших и плохих, олицетворяя хакерский дух. Большая часть книги посвящена людям, которые улучшили мир, используя компьютеры, но в ней также были и хакеры, которых бы сегодня арестовали за их действия. Некоторые из тех хакеров верили, что цель оправдывает средства, и следовали слишком свободным моральным принципам, воплощенным в чем-то, что Леви называл "этикой хакера". Главным их заблуждений была философия, что к любому компьютеру можно получить доступ по любой обоснованной причине, что вся информация должна быть свободной, и что нельзя доверять властям. Это было идеализированное видение хакеров и взлома, несмотря на то, что оно не скрывало спорных вопросов этики и правомерности. На самом деле оно формировалось вокруг недавно установленных границ.

Стивен Леви был первым автором, которому я когда-либо отправлял копию его собственной книги, и просил расписаться на ней и отправить обратно (об этом и меня просили уже несколько раз, т.к. к этому моменту я написал 8 книг). Леви перешел к написанию статей или работал техническим редактором в нескольких

крупных журналах, таких, как *Newsweek*, *Wired* и *Rolling Stone*, а также он написал шесть других книг по проблемам информационной безопасности. На сегодняшний день Леви продолжает быть значимым писателем в сфере технологий. С помощью его книги *Хакеры (Hackers)* я познакомился с удивительным миром хакинга.

Позднее, другие книги, такие как *Flu-Shot* Росса Гринберга (больше не печатают) и книга Джона Макафи *Computer Viruses, Worms, Data Diddlers, Killer Programs, and Other Threats to Your System* (https://www.amazon.com/Computer-virusesdiddlers-programs-threats/dp/031202889X) познакомили меня с борьбой с киберпреступностью. Я прочитал эти книги и был настолько вдохновлен, что решил посвятить жизнь сражению с теми же угрозами.

За это время я понял, что защитники - самые умные хакеры. Я не хочу стричь под одну гребенку всех киберпреступников и говорить, что они посредственные. Каждый год хакеры-мошенники открывают что-то новое. Среди них есть несколько действительно умных хакеров. Но подавляющее большинство злоумышленников по-настоящему ничем не выделяются и просто повторяют то, что работало двадцать лет. Откровенно говоря, среднестатистический киберпреступник не имеет достаточного таланта программирования, чтобы написать простое приложение в блокноте и тем более самому разобраться, как взломать систему, шифрование, или сразу успешно разгадать пароль - они делают это с помощью других хакеров, которые действительно ломали голову годами ранее.

Ирония в том, что сверхумные люди в компьютерном мире, которых я знаю, не киберпреступники, а защитники. Они обязаны знать все, что делает хакер, предугадывать, что он может сделать в будущем, и создать дружелюбную, дешевую защиту от всего. Мир защитников полон докторов наук, магистров и успешных предпринимателей. Хакеры меня редко впечатляют. Защитники делают это постоянно.

Защитникам свойственно открывать новые методы взлома, и при этом они держат их в секрете. Работа защитников - защищать, а если рассказывать хакерам о бреши в безопасности, перед тем, как защита будет установлена, то это никак не поможет. Так живут защитники. Они выявляют новые способы взлома и помогают закрыть дыру перед тем, как о ней станет известно остальному миру. Так происходит гораздо чаще, чем, когда киберпреступники сами узнают о дыре.

Я даже видел, как защитники выявляли новый способ взлома, но, по причине дороговизны или нехватки времени, дыра не была сразу устранена, и позже, о ней узнавали киберпреступники, которые потом получали славу “первооткрывателей”. К сожалению, защитники не всегда получают славу и признание делая свою работу.

Наблюдая за хакерами и за защитниками почти три десятилетия, мне стало очевидно, что защитники – это более впечатляющая категория. Хакеры даже и близко не стоят. Если вы хотите показать всем, как хорошо вы умеете обращаться с компьютерами, не показывайте новый способ взлома. Покажите им новую, более продвинутую защиту. Для поиска нового способа взлома много ума не нужно. Как правило, хватает настойчивости. Но для того, чтобы создать что-то, что сможет выдержать постоянные попытки взлома в течение долгого времени нужен особенный и умный человек.

Если вы хотите впечатлить мир, не ломайте гараж. Вместо этого, создайте код, который сможет выдержать кувалду хакера.

Глава 2. Как Хакеры Взламывают

Самое приятное в моей работе - это тестирование на проникновение (также известное, как пентест). Пентестинг - это взлом в самом истинном смысле. Это человек против машины в битве умов. Человек, "атакующий", может использовать собственную изобретательность и новые или существующие инструменты для поиска слабых мест, которые появились в результате работы системы или из-за ошибки разработчика человека. За все годы, что я проводил пентесты, даже не смотря на то, что мне обычно давались недели, чтобы провести тест, в большинстве случаев я успешно взламывал свою цель примерно за час. Самый долги взлом занял три часа. Это были банки, правительственные сайты, больницы или сайты корпораций, которые меня нанимали для тестирования защиты.

При этом, я не такой уж хороший пентестер. По шкале от 1 до 10, где 10 - это лучший результат, у меня примерно 6 или 7. На стороне защитников я чувствую себя лучшим человеком в мире. Но, мой уровень, как атакующего, средний. Меня окружают великолепные пентестеры - мужчины и женщины, которые думают только о написании собственных тестирующих инструментов, и те, кто не считают тест успешным, до тех пор, пока не сгенерируют хотя бы одно событие в логе, которое потребует пристального внимания. Но даже люди, которым я бы поставил 10, обычно считают, что у них средний уровень и они и восхищаются другими пентестерами, которые по их мнению "десятки". Насколько хороши должны быть эти хакеры?

Но вам не нужно быть безмерно хорошим, чтобы быть очень успешным хакером. Вам даже необязательно взламывать клиента, который вас нанял (я исхожу из того, что вам платят за законный заказ на пентест), чтобы вас радовала ваша работа. На самом деле клиент будет в абсолютном восторге, если у вас не получится. Они начинают хвастаться, что наняли хакеров, а их сеть выдержала атаку. Это победа-победа (win-win) для всех участников. Вам платят столько же, а они хвастаются, что их невозможно взломать. Это единственная работа, которую я знаю, где у вас не может быть плохого результата. К сожалению, я не знаю ни одного пентестера, который бы *хоть раз* не взломал *все* свои цели. Я уверен, должны быть те, у кого не получилось, но подавляющее большинство пентестеров "получает свой приз".

ПРИМЕЧАНИЕ Если во время пентестинга вы не нашли слабых мест и вскоре, после этого вашего клиента взломали настоящие злоумышленники, то это не пойдет вам на пользу. Если так произойдет несколько раз, пойдет молва, и вам,

скорее всего, надо будет искать другую работу. Слабые месте есть всегда. Найдите их.

Как правило, пентестеры делают что-то еще, чтобы произвести впечатление на старших менеджеров, например, тайком сфотографируют CEO за его столом, используя камеру на его компьютере, или вставляют пароль администратора домена в картинку с пиратским флагом, которую ставят в качестве скринсейвера администратора. Получившаяся картина стоит тысячи слов. Никогда недооценивайте, насколько сильно доволен вашей работой может быть клиент из-за одной глупой фотки. Если есть такая возможность, всегда заканчивайте работу зрелищно. Делая так вы станете «золотым» консультантом по безопасности.

Секрет Хакерства

Если и существует секрет хакерства, то он заключается в том, что никакого секрета нет. Это процесс обучения правильным методам и использование правильных инструментов для работы, то же самое делают электрики, сантехники или строители. Существует не один способ взлома. Однако, существует определенный набор шагов, которым можно описать процесс взлома. Не все хакеры используют все шаги. Некоторые хакеры используют только один шаг. Но в целом, если следовать всем этим шагам, то велика вероятность стать успешным хакером. Можно пропустить один или несколько шагов и все равно быть успешным. Вредоносное ПО и другие инструменты взлома часто позволяют хакерам пропускать шаги, но хотя бы один из этих шагов, изначальное проникновение, всегда требуется.

Независимого от того, хотите ли вы сделать карьеру (легального) хакера, если вы будете сражаться с киберпреступниками, то вам нужно понять "методологию взлома" (название методики может отличаться). Модели могут отличаться количеством шагов, названием шагов, и описанием каждого шага, но они все состоят из тех же самых основных компонентов.

Методология Взлома

Методология взлома включает приведенные ниже последовательные шаги:

1. Сбор информации
2. Проникновение
3. Опционально: Обеспечение простого доступа в будущем

4. Разведка системы
5. Опционально: Движение
6. Выполнение запланированного действия
7. Опционально: Заметание Следов

Сбор информации

За исключением случаев, когда инструмент хакера помогает ему случайно получить доступ к уязвимому сайту, как правило, у хакера в голове есть определенная цель. Если хакер хочет проникнуть в определенную компанию, первое, что делает хакер - начинает изучать об этой компании все, что может помочь взлому. Как минимум, это IP-адреса, адреса электронной почты и адреса доменов. Хакер ищет информацию о количестве потенциальных сайтов и сервисов, связанных с компанией, к которым он может получить доступ. Они берут информацию из СМИ и публичных финансовых отчетов. Они узнают имена главных исполнительных лиц и прочих сотрудников, они хотят воспользоваться социальной инженерией. Хакер просматривает новостные ленты, он хочет узнать, какое важное программное обеспечение недавно приобрели те, кого он хочет атаковать, какие происходят слияния или разделения в компании (в этих процессах всегда много неточностей и они часто сопровождаются слабой или отсутствующей безопасностью), и даже у партнеров компании, которую хочет взломать хакер. Многие компании были взломаны через гораздо более слабых партнеров.

Самая важная часть процесса получения информации для большинства хакерских атак - это узнать, какие цифровые активы связаны с компанией. Обычно узнают не только о главных (публичных) сайтах и сервисах - как правило, для атакующего гораздо полезнее найти менее популярные связанные сайты и сервисы, такие, как порталы для сотрудников и партнеров. Чем менее популярны сайты и серверы, тем более вероятно наличие у них слабых мест, по сравнению с главными сайтами, которые уже годами пытаются взломать.

Затем любой хороший хакер начинает собирать все ПО и сервисы, расположенные на каждом из этих сайтов, этот процесс общеизвестен, как *фингерпринтинг* (снятие отпечатков пальцев). Очень важно узнать, какие операционные системы (ОС) и какие их версии ОС используются. Версии ОС могут сказать хакеру, о том, какие уровни патчей и какие баги могут присутствовать или отсутствовать. Например, они могут найти Windows Server 2012 R2 и Linux Centos 7.3-1611. Затем они смотрят какие программы и версии этих программ (чтобы также узнать о багах и патчах) работают на каждой ОС. Если это веб-сервер, они могут найти Internet Information Server 8.5 на сервере Windows или Apache 2.4.25 на сервере Linux. Они проводят инвентаризацию каждого устройства, ОС, приложения и их версий на каждом из интересующих

хакеров объектов. Всегда лучше провести полную инвентаризацию, чтобы получить целостную картину объекта и всего, что с ним связано, но иногда хакер может сразу найти большую уязвимость и просто перейти к следующему шагу. За исключением случаев такого быстрого вторжения, обычно, чем больше информации есть у хакера, тем лучше. Каждое дополнительное ПО и его версия предоставляют дополнительный возможный вектор атаки.

ПРИМЕЧАНИЕ Некоторые хакеры называют общую, не техническую информацию *футпринтинг (отпечаток ноги),* а составление схемы ОС и программного обеспечения *фингерпринтинг (отпечаток пальца)*.

В некоторых случаях, когда хакер подключается к серверу или сайту и тот любезно предоставляет детальную информацию об используемых версиях ПО, никакие инструменты не нужны. Для других ситуаций существует множество инструментов, помогающих сделать фингерпринт ОС и приложений. Безоговорочно, номер один для хакеров - это Nmap (https://nmap.org/). Nmap существует с 1997-го. Есть несколько версий, включая версии для Windows и Linux, и это швейцарский нож хакеров. С помощью этой программы можно осуществлять все виды сканирования и тестирования серверов. С ее помощью можно собрать фингерпринты ОС и приложений. Есть более информативные фингерпринтеры приложений, особенно, если они созданы для фингепринта определенного типа приложений, таких как веб-серверы, базы данных или email-серверы. Например, Nikto2 (https://cirt.net/Nikto2), не только делает фингепринт веб-серверов лучше, чем Nmap, но также предоставляет возможность проводить тысячи тестов на проникновение, и позволяет вам узнать о существующих уязвимостях.

Проникновение

Все начинается с первичного проникновения. Весь процесс взлома зависит от успеха этого шага. Если хакер предварительно собрал всю необходимую информацию, то на этом этапе у него не возникнет никаких проблем. Этот шаг никогда не вызывал у меня затруднений. На компьютерах всегда используется старое ПО, всегда что-то остается не пропатченным, и почти всегда что-то из опознанного ПО неправильно настроено.

ПРИМЕЧАНИЕ Один из моих любимых трюков - атака того самого ПО и тех устройств, которые защитники используют для защиты своих сетей. Часто это доп. оборудование, которое, по сути, является «костылями». Такое оборудование печально известно тем, что на него годами не устанавливаются патчи.

Если, каким-то образом, все ПО и устройства идеально защищены (а такого не бывает), то вы можете использовать человеческий фактор, являющийся самой слабой частью уравнения. Но без первичного проникновения для хакера все потеряно. К счастью для него, существует множество способов проникнуть в цель. Вот различные техники, которые хакер может использовать для взлома:

- Уязвимость нулевого дня
- Непропатченное ПО
- Вредоносное ПО
- Социальная инженерия
- Слабые пароли
- Перехват сообщений / MitM
- Утечка данных
- Неправильная настройка оборудования
- Denial of service (DoS, "отказ в обслуживании")
- Инсайдер / партнер / консультант / производитель / третьи лица
- Ошибка пользователя
- Физический доступ
- Повышение привилегий

Уязвимость нулевого дня Эксплойты нулевого дня (или 0-day) встречаются реже, чем ежедневные уязвимости для которых производитель, как правило, уже давно выпустил патчи. Эксплойт нулевого дня - уязвимость, которая так и осталась на целевом ПО, а общественность (в том числе, как правило, и продавец) не знает об этом. Любой компьютер, на котором используется ПО с багом нулевого дня, по сути, добровольно открывает к себе доступ, если конечно, потенциальная жертва, сама не удалит это ПО или не установит что-то, чтобы минимизировать риск (например, файрволл, ПО, предотвращающее переполнение буфера и так далее).

Уязвимости нулевого дня не так часто используются, как известные, потому что злоумышленники не могут злоупотреблять ими. Если атакующий злоупотребляет нулевым днем, то поставщики обнаружат и исправят желанный эксплойт. В наши дни, большинству производителей понадобиться от нескольких часов до нескольких дней, чтобы исправить новые уязвимости. Уязвимости нулевого дня либо масштабно применяются сразу против огромного количества целей, чтобы извлечь из этого максимальную выгоду, либо их используют "тише воды, ниже травы", то есть эпизодически, редко и только когда надо. У лучших профессиональных хакеров в мире, как правило, есть коллекция нулевых дней, которые они используют только тогда, когда все остальное не сработало, но даже при этом они используют их тайком, не оставляя следов. Уязвимость

нулевого дня может использоваться для получения первичного доступа к самой стойкой цели, а затем все следы использования этой уязвимости удаляются, и с этого момента применяются более традиционные способы.

Непропатченное ПО Непропатченное ПО - это одна из самых частых причин наличия эксплойта у компьютера или устройства. Каждый год публично анонсируются тысячи (обычно между 5000 и 6000, или 15 в день) новых уязвимостей среди популярного ПО. (Посмотрите отчеты Microsoft *Security Intelligence Report* по каждой проблеме, http://microsoft.com/sir). Производители, в целом, стали писать более безопасный код и находить свои собственные баги, но количество программ постоянно увеличивается, пишутся миллиарды строк кода, поэтому общее количество багов, в целом, не изменилось за последние два десятилетия.

Многие производители действительно хорошо работают, вовремя выпуская патчи для своего ПО, особенно после того, как об уязвимости стало общеизвестно. К сожалению, потребители печально известны тем, что медленно устанавливают эти патчи, а часто, даже заходят еще дальше, и отключают автоматическую установку обновлений от производителя. Существенный процент пользователей никогда не устанавливает патчи на свои системы. Они либо игнорируют многочисленные предупреждения о необходимости установки патчей, воспринимая их только, как раздражающий фактор, либо вообще не знают, что патч должен быть установлен. (Например, многие кассовые терминалы не сообщают кассирам, что нужно установить обновление). Большинство программных эксплойтов применяются против ПО, на которое многие годы не устанавливались патчи.

Даже если конкретная компания или пользователь устанавливают патчи, исправляющие критические уязвимости, сразу после того, как о них стало общеизвестно, настойчивый, терпеливый хакер может просто дождаться сообщения о новом патче для программы, которая есть в его списке, и начать соответствующую атаку, перед тем, как у защитника будет время устранить уязвимость. (Хакеру не составит труда осуществить реверс-инжениринг патча и выяснить, как использовать определенную уязвимость).

И уязвимости нулевого дня и обычные уязвимости ПО - это, в сущности, результат небезопасных методов написания ПО. Программные уязвимости будут описаны в Главе 6.

Вредоносное ПО Вредоносное программное обеспечение, известное, как малварь- это всем известные вирусы, троянские программы и черви, но сегодня вредоносное ПО - это часто гибридная смесь нескольких типов таких программ. Вредоносное ПО позволяет хакеру облегчить использование эксплоитов и облегчить процесс атаки своих жертв, либо увеличить число жертв за небольшой промежуток времени. Когда обнаруживается новый эксплойт, защитники знают,

что создатели вредоносного ПО используют автоматические программы, чтобы как можно быстрее заразить как можно больше машин, то есть, так сказать, "вооружиться". Хотя эксплойты - это то, чего нужно избегать, процесс «вооружения» создает наибольший риск для конечных пользователей и общества. Без вредоносного ПО злоумышленник вынужден атаковать только одну жертву за раз. Но с ним, за несколько минут, можно взломать миллионы машин. Более детальное рассмотрение вредоносного ПО будет описано в Главе 9.

Социальная Инженерия Одна из самых успешных стратегий взлома - социальная инженерия. Социальная инженерия, осуществляемая либо самим злоумышленником, либо автоматически, и представляет собой хакерский трюк, суть которого состоит в том, чтобы заставить конечного пользователя обманным путем сделать что-то, причиняющее вред его компьютеру или безопасности. Это может быть электронная письмо, обманывающее конечного пользователя, чтобы тот перешел по вредоносной ссылке или открыл сомнительное вложение. Это может что-то или кто-то, обманом заставляющий пользователя раскрыть личные данные для входа (то есть, *фишинг*). Социальная инженерия давно присутствует в арсенале хакеров. Долгое время, хакер в белой шляпе, Кевин Митник, был одним из лучших в применении социальной инженерии. Митник представлен в Главе 5, а социальная инженерия более детально рассмотрена в Главе 4.

Слабые пароли Пароли могут быть подобраны или украдены. Долгое время простой подбор пароля (или социальная инженерия) был одним из самых популярных методов получения первоначального доступа к компьютерной системе или сети, кстати, он до сих пор таким остается. Но за последние пять лет кража учетных данных и их повторное использование (например, атаки pass-the-hash) получили по-настоящему широкое распространение. Используя атаку кражи учетных данных, злоумышленник, как правило, получает доступ к учетной записи администратора компьютеруа или администратора устройства и получает данные одной или нескольких учетных записей, хранящиеся в системе (неважно, хранятся ли те в оперативной памяти или на жестком диске). Украденные учетные данные затем используются, чтобы получить доступ к другим системам, в которых используются те же самые учетные данные. Почти каждая атака на крупную корпорацию заключалась в краже учетных данных. Это основной уязвимый компонент. Из-за этого традиционный подбор паролей больше не так популярен. Взлом паролей описан в Главе 21.

Прослушка/MitM Прослушка и атаки "человек посередине" (man-in-the-middle, MitM) заключаются в незаконном использовании сетевого соединения для получения доступа к сообщениям или для участия в обмене информацией. В большинстве случаев прослушка возможна из-за уязвимостей в протоколах сетей

или приложенияй, но оно также может осуществляться из-за человеческих ошибок. В наши дни эта атака чаще всего направлена на беспроводные сети. Сетевые атаки будут описаны в Главе 33, а атаки на беспроводные сети - в Главе 23.

Утечка данных Утечка личной информации может быть результатом взлома или результатом непреднамеренного (или намеренного) действия человека. В большинстве случаев утечка данных происходит из-за их неосторожного (или недостаточно защищенного) хранения, или потому что какой-то хакер узнал какой-то способ, с помощью которого можно получить доступ к чьим-то данным. Также довольно распространенной формой взлома являются атаки изнутри, когда сотрудник, которому пообещали «плюшки», или наемный исполнитель намеренно крадет или использует личною информацию. Несколько глав в этой книге будут посвящены предотвращению утечки личных данных.

Неправильная настройка Довольно часто пользователи и администраторы (порой не намерено) используют довольно слабые защитные механизмы. Я уже и не вспомню, на скольких сайтах для самых важных файлов стоят права “Доступен для всех” или “Всему миру” - и эти права работают именно так. Когда вы сообщаете всему миру, что они могут использовать любые файлы с вашего сайта, то он, а точнее файлы, которые на нем хранятся, довольно быстро станут достоянием общественности. Настройка и безопасность операционных систем описана в Главе 30.

Denial of Service (DoS) Даже, если бы никто не допустил ни единой ошибки и были бы установлены все патчи, по-прежнему можно отключить от интернета почти любой веб-сайт или компьютер. Даже, если делать все идеально, компьютеры полагаются на сервисы, которые нельзя контролировать, а они не идеальны. Сегодня массивные распределенные атаки типа «отказ в обслуживании» (DDoS) могут “уронить” или серьезно повлиять на любой веб-сайт или компьютер, подключенный к интернету. Такие атаки часто заключаются в отправке миллиардов вредоносных пакетов в секунду, которые перегружают целевой сайт (или канал). Существуют десятки коммерческих (иногда нелегальных) сервисов, которые каждый может использовать, как для осуществления, так и для защиты от серьезных DDoS атак. DDoS атаки описаны в Главе 28.

Инсайдер / партнер / консультант / производитель / третьи лица Даже, если ваша сеть и все компьютеры в ней идеальны (а это не так), вас могут взломать с помощью уязвимости на компьютере партнера или инсайдеры, которые на вас работают. Это довольно широкая категория и она связана со многими другими методами взлома.

Ошибка пользователя Эта категория проникновения также связана со многими другими методами взлома. Например, пользователь может случайно отправить личные данные, вписав один неверный символ в адресе электронной почты. Пользователь может случайно не установить патч для важного сервера, или случайно выставить не те права. Часто ошибка пользователя заключается в том, что он отвечает на электронную почту, и при этом думает, что отвечает только одному человеку или небольшой группе людей, но на самом деле, он отвечает гораздо большему количеству людей или даже человеку, о котором они пренебрежительно говорят. Здесь я отдельно выделил ошибку пользователя только потому, что иногда ошибки случаются, и хакеры готовы извлечь из этого выгоду.

Физический доступ Народная мудрость гласит, что, если злоумышленник получил физический доступ к устройству, то он может просто его украсть (пуф, ваш телефон исчез) или уничтожить или в конце концов обойти все методы защиты и получить доступ к личным данным. И пока такое суждение является довольно точным, даже учитывая, что такой способ используется для борьбы с системами, которые призваны защитить от физического взлома. Например, многие программы шифрования дисков можно обойти, используя электронный микроскоп, с помощью которого можно найти секретный защитный ключ, обнаружив отдельные электроны, из которых состоит этот ключ. Или можно заморозить ОЗУ баллончиком со сжатым воздухом, чтобы в открытом виде увидеть секретный шифровальный ключ, все дело в ошибке в алгоритме физического хранения данных.

Повышение привилегий Хакеры использует различные методы проникновения, описанные в предыдущих разделах, чтобы получить первоначальный доступ к целевой системе. Единственный вопрос, который возникает после получения первоначального доступа - это какой тип доступа к безопасности у них есть. Если они взламывают программное обеспечение или сервис, работающий на том уже уровне безопасности, что и пользователь, то изначально у них есть только те же права и разрешения, которые есть у (авторизированного) пользователя, зашедшего под своими данными. Или они могут сразу взять Святой Грааль системы, получив права администратора. Если у злоумышленников есть только обычный, непривилегированный доступ, то, чтобы получить доступ с большими правами, они, как правило проводят атаку повышения прав. Для этой атаки нужно применить весь диапазон приемов, используемых для проникновения, фактически делая то же самое, но с той разницей, что теперь у хакера есть хоть какой-то доступ. А, так как первоначальное проникновение почти всегда завершается успехом, то свершить атаку повышения привилегий уже гораздо легче.

Обеспечение Простого Доступа в Будущем

Хотя это и опционально, большинство хакеров, после того, как получили первоначальный доступ, работают над тем, чтобы в следующий раз было легче получить доступ к тому же устройству или ПО. Многие хакеры устанавливают бэкдор к которому в следующий раз они смогут быстро подключиться. Либо, вместо этого, они могут взломать пароли или создать новые аккаунты. Атакующий всегда может использовать тот же эксплойт, который успешно сработал в прошлый раз, когда он получил первоначальный доступ, но обычно злоумышленникам нужен другой метод, который будет работать, даже, если жертва закрыла эту уязвимость.

Разведка Системы

Большинство хакеров, как только проникают в систему, запускают несколько команд или программ, чтобы больше узнать о цели, к которой получили доступ, а также все, что с ней связано. Обычно они просматривают содержимое памяти, жесткого диска, сетевых подключений, а также осуществляют подсчет пользователей, доступных файлов, используемых сервисов и программ. Вся эта информация нужна для лучшего понимания цели, и она так же используется в качестве отправной точки для следующей атаки.

Движение

Очень редко злоумышленник или вредоносное ПО взламывают только одну цель. Почти все хакеры и вредоносные программы хотят распространить свое влияние на максимальное количество целей. Как только они получают доступ к первой цели, они довольно легко распространяют свое влияние внутри сети этого компьютера или организации. Хакерские методы, перечисленные в этой главе, характеризуют различные способы проникновения, но по сравнению с усилиями, необходимым для первоначального проникновения, последующее движение осуществить намного легче. Если злоумышленник переходит к схожим целям, которые используются для тех же целей, что и первая машина, то это называется горизонтальным движением. Если он движется от устройства с одними правами к устройству с более высокими или низкими правами, то это называется вертикальным движением.

Большинство злоумышленников движется от более низких к более высоким правам, используя техники вертикального движения (повторюсь, используя хакерские методы проникновения, описанные в этой главе). Например, очень часто методология хакеров заключается в том, чтобы сначала взломать одну рабочую станцию обычного конечного пользователя. Они используют изначальный доступ, чтобы найти и скачать пароли локальных администраторов. Затем, если эти учетные данные локального администратора используются на

нескольких машинах (зачастую так и есть), то злоумышленники движутся горизонтально, повторяя процесс, пока не получат доступ к самому привилегированному аккаунту. Иногда это делается во время первого же взлома, потому что взломанный пользователь или система уже имеет высокие привилегии. Затем они двигаются к серверу аутентификации и получают учетные данные всех пользователей. В наши дни это стандартный modus operandi для большинства хакерских групп, и движение от изначального взлома до полного контроля над сетью (на хакерской терминологии pwning) может занять менее часа.

Мой собственный опыт, и помните, я всего лишь средний хакер, показывает, что, как правило, мне нужен примерно час, чтобы получить изначальный доступ, и еще час, чтобы захватить централизованную базу данных аутентификации. То есть мне, среднему хакеру, нужно примерно два часа, чтобы полностью завладеть компанией. Самый долгий мой взлом занял три часа.

Выполнение Запланированного Действия

После того, как доступ получен и устройство взято под контроль, хакеры делают то, что задумали (если только сам взлом не является для них чем-то новым). У каждого хакера есть свои намерения. У тестировщика безопасности, работающего по найму, есть определенные обязательства по контракту. Киберпреступник может распространять вредоносное ПО, читать или красть конфиденциальную информацию, добавлять вредоносный код в программы или нанести ущерб. Вся суть взлома хакером одной или нескольких систем заключается в намерении что-то сделать. В прежние времена (два или три десятилетия назад) факта взлома было достаточно для большинства хакеров. Сегодня взлом на 99% мотивирован преступными намерениями, и хакеры хотят нанести цели какой-либо ущерб (даже, если весь урон, который они нанесли, заключается в обеспечении последующих подключений к системе для осуществления возможных планов). Несанкционированный доступ без нанесения прямого урона - это все равно урон.

Заметание Следов

Некоторые хакеры пытаются замести следы. Раньше почти все хакеры это делали, но в наши дни компьютерные системы настолько сложны и их так много, что большинство владельцев устройств не проверяет наличие следов взлома. Они не проверяют логи, не проверяют файрвол, не ищут никаких следов нелегальной активности, пока не столкнуться с ее последствиями. Каждый год в отчете компании Verizon *Data Breach Investigations Report* (http://www.verizonenterprise.com/verizon-insights-lab/dbir/) сообщается, что большинство атакующих остаются незамеченными на протяжении от нескольких

месяцев до нескольких лет, и более 80% атак были бы замечены, если бы защитники озаботились проверкой. Из-за этой статистики большинство хакеров больше не беспокоится о заметании следов.

В наши дни хакерам можно не волноваться, потому что они используют методы, которые невозможно обнаружить, используя традиционные средства обнаружения взлома. Точнее то, что используют хакеры настолько близко к обычным для жертвы действиям, что почти невозможно отличить легальную активность от нелегальной. Например, после взлома хакер, как правило, совершает те же действия, что и обычный пользователь, часто получая доступ к тем же серверам и сервисам, с которыми работает обычный пользователь. И они используют те же инструменты (такие как программы удаленного доступа и сценарные языки), с которыми работают администраторы. Сфера обнаружения вторжения описана в Главе 14.

Взлом Прост

Если вы хотите знать, как взламывают хакеры, то пожалуйста. Вся эта глава посвящена описанию их действий. Все, что остается добавить - это инструменты, любопытство и настойчивость. Цикл взлома настолько хорошо работает, что многим тестировщикам безопасности, которые поначалу были в восторге от того, что им платят за профессиональный взлом, в итоге становится скучно, и через пару лет они находят другую сферу деятельности. Нужны ли еще подтверждения безупречной работы этого цикла? И это в рамках понимания того, что защитникам нужно сражаться против атакующих.

Автоматическое Вредоносное ПО, Как Инструмент Взлома

Когда используется вредоносное ПО, с его помощью может осуществляться один или несколько шагов, все может быть автоматизировано, либо инструмент может передавать ручное управление, как только получен доступ к цели (pwned). Большинство хакерских групп использует комбинацию социальной инженерии, автоматического вредоносного ПО для реализации своих целей. В больших группах у отдельных хакеров могут быть свои роли и обязанности. Вредоносное ПО может выполнить проникновение за 1 шаг, ему нет необходимости применять остальные. Например, размер самой быстрой в истории вредоносной программы SQL Slammer был всего 376 байт. Она переполняет буфер SQL UDP порта 1434, независимо от того, работает ли в данный момент SQL. Так как большинство компьютеров не используют SQL, можно подумать, что это малоэффективно. Но нет, за 10 минут эта программа изменила мир. Ни одна вредоносная программа

даже близко никогда не была к тому, чтобы заразить так много хостов за такой короткий промежуток времени.

ПРИМЕЧАНИЕ Если я пропустил какой-то шаг в методологии хакеров, или пропустил метод проникновения, я извиняюсь. Но повторюсь, я просто средний хакер.

Этичное Хакерство

Мне бы хотелось думать, что мои читатели - этичные хакеры, которые убедились в том, что у них есть все права на взлом цели, на которую они устремили свой взгляд. Взламывать сайт, не имея при этом заранее определенного и явного разрешения, неэтично и часто нелегально. Неэтично даже (и, как правило, нелегально) взламывать сайт и бесплатно сообщить владельцам о найденной уязвимости. Неэтично, и часто нелегально искать уязвимости, а затем просить владельцев сайта нанять вас пентестером. Последний сценарий происходит постоянно. Мне жаль, что не существует способа сообщить, что вы нашли как взломать сайт или сервер, и попросить за это деньги или работу, и при этом не выглядеть вымогателем. Я точно могу вам сказать, что почти все сайты, получающие подобные неожиданные запросы, не считают вас полезным и не хотят вас нанимать. Они видят в вас врага и тут же звонят юристам.

Остальная часть этой книги посвящена описанию конкретных типов взлома, определенных методов проникновения, тому, как защитники борются с этими методами, а также экспертам в своей области и их борьбе с хакерами. Если вы хотите посвятить свою жизнь взлому, точнее сражению с хакерами, вам нужно понять хакерскую методологию. Люди о которых пойдет речь в этой книге - гиганты в своей области, и у них можно многому научиться. Они проложили путь. Можно начать с 3-ей Главы, где речь идет Брюсе Шнайере, которого многие считают отцом современной компьютерной криптографии.

Глава 3. Профиль: Брюс Шнайер

Брюс Шнайер - человек с таким большим количеством опыта и знаний, что многие, представляя его, используют слова "светило индустрии". Начав тем, кого многие люди называют "отцом современной компьютерной криптографии", Шнайер вышел за рамки шифрования, и начал серьезнее задумываться об информационной безопасности, и почему она не стала значительно лучше после всех этих десятилетий. Его влияние и четкое понимание ситуации позволяет ему говорить на различные темы, связанные с информационной безопасностью. Его часто приглашают в качестве эксперта на национальные телевизионные шоу, и он несколько раз свидетельствовал перед Конгрессом США. Шнайер пишет и ведет блог, и то, чему я научился изучая блог я всегда считал своей неформальной степенью магистра в информационной безопасности. Я бы и наполовину не был тем специалистом в информационной безопасности, которым являюсь сегодня, если бы не он. Он - мой неофициальный наставник.

Шнайер известен за обезоруживающе простые высказывания, в которых раскрывается самая суть, а иногда даже больше, того, что раньше считалось абсолютно правильным убеждением или догмой. Например, "Если вы сосредоточились на SSL-атаках, то вы добьетесь больших успехов в информационной безопасности, чем весь остальной мир". Он имел в виду, что существует так много других, чаще используемых и более эффективных, эксплойтов, о которых стоит волноваться, что если вы действительно беспокоитесь о редко используемом SSL-эксплойте, то вы, должно быть, уже решили более вероятные, более важные проблемы. Другими словами, нужно приоритезировать усилия в информационной безопасности, вместо того, чтобы реагировать на каждую недавно анонсированную (и порой никогда не используемую) уязвимость.

Он также давал комментарии по поводу недовольства работников информационной безопасности, связанного с халатным отношением сотрудников к паролям. Многие сотрудники используют слабые пароли, одинаковые пароли на многих, не связанных веб-сайтах (так и просят, чтобы их взломали), и часто дают свои пароли друзьям, коллегам и даже незнакомцам. Мы расстраиваемся из-за чувства бессилия, потому что знаем вероятные последствия для бизнеса, но конечный пользователь не осознает риск для компании, используя слабый пароль. Шнайер научил нас, что конечный пользователь оценивает пароль на основе риска для самого себя. Сотрудников редко увольняют за использование слабых паролей. Даже, если хакер крадет банковские средства конечного пользователя, их, как правило, немедленно возвращают. Шнайер также показал

нам, что это мы, профессионалы в компьютерной безопасности, не осознаем риск. И пока риск не причинит вреда, непосредственно, конечным пользователям, они не станут добровольно менять свое поведение. Каково это, думать, что вы были экспертом в этом вопросе, а оказалось, что конечный пользователь лучше осознает риск?

Он автор более 12 книг, включая такие, как книгу 1996 года *Applied Cryptography: Protocols, Algorithms and Source Code in C* (https://www.amazon.com/Applied-Cryptography-Protocols-Algorithms-Source/dp/1119096723). Он так же написал несколько книг по криптографии (в том числе, в соавторстве с Нильсом Фергюсоном) и стал работать над давно интересовавшими его причинами, из-за которых информационная безопасность не становится лучше. В результате появилась серия книг, каждая из которых исследовала нетехнические причины (доверие, экономика, общество и так далее) постоянной слабости защиты. Эти книги наполнены простыми для восприятия историями и наглядными примерами. Вот несколько моих любимых книг Шнайера:

- *Secrets and Lies: Digital Security in a Networked World* (https://www.amazon.com/Secrets-Lies-Digital-Security-Networked/dp/0471453803)
- *Beyond Fear: Thinking Sensibly About Security in an Uncertain World* (https://www.amazon.com/Beyond-Fear-Thinking-Sensibly-Uncertain/dp/0387026207)
- *Liars and Outliers: Enabling the Trust that Society Needs to Thrive* (https://www.amazon.com/Liars-Outliers-Enabling-Society-Thrive/dp/1118143302/)
- *Data and Goliath: The Hidden Battles to Collect Your Data and Control Your World* (https://www.amazon.com/Data-Goliath-Battles-Collect-Control/dp/039335217X/)

Если вы действительно хотите понять информационную безопасность, почему она не становится лучше, и какие проблемы ей угрожают, я советую вам прочитать эти книги. Вам также стоит почитать его блог (https://www.schneier.com/) и подписаться на его ежемесячную новостную рассылку Crypto-Gram (https://www.schneier.com/crypto-gram/). Люди, которые регулярно читают Шнайера совершенствуются по сравнению с теми, кто этого не делает. Его стиль письма доступен и развлекателен, и он нисколько не похож на тех, кто пишет о "выдуманных " проблемах безопасности. Его прошлые разгромные статьи о мошенниках, которые "изобрели" новый способ шифрования, в разделе "Doghouse" (в блоге) - это уроки сами по себе. Он постоянно пишет о самых важных вопросах повседневности.

Я много раз за эти годы брал интервью у Шнайера, и иногда они могут поставить интервьюера в неловкое положение. Не потому, что с ним сложно общаться (это не так), и не потому что он смотрит на вас свысока (это тоже не так), а из-за того, что он часто ищет способ позволить интервьюеру самому сделать окончательный вывод, следуя своим убеждениям и предположениям. Если вы чего-то не понимаете или соглашаетесь с ним, то он не начинает сразу отрицать ваши доводы. Вместо этого он будет задавать вам вопрос за вопросом в интеррогативном стиле, позволяя вашим ответам на эти вопросы, привести вас к окончательному выводу. Шнайер всегда учит, даже когда у него берут интервью. Вы понимаете, что он уже давно обдумывал эти серьезные вопросы, и обсудил их в своей голове гораздо подробнее, чем вы. Я попытался позаимствовать некоторые из его техник самоопроса, когда приходил к собственным новым убеждениям.

Я спросил Шнайера, как он впервые заинтересовался информационной безопасностью. Он ответил: "Меня всегда интересовала математика и секретные шифры - криптография. Моя первая книга, *Applied Cryptography* в итоге стала книгой, которую я сам хотел бы прочитать. Но я всегда стремился расширить свой кругозор. Я понял, что самая большая проблема была не в технологиях. Самая большая проблема в человеке, точнее в интерфейсе, с которым взаимодействует человек. Главные проблемы компьютерной безопасности заключаются не в технологиях, а в том, как мы используем эти технологии, учитывая социальные, политические и экономические факторы, связанные с информационной безопасностью. Я много времени думаю о пользователях, с высоким уровнем риска. У нас есть технологии, с помощью которых мы можем их защитить, но можем ли мы создать эффективные решения, которые не помешают им делать свою работу? В противном случае, у нас никогда не получится убедить их использовать эти решения".

Я спросил его, что он думает о недавних утечках из разведывательных служб США. Он сказал: "Вся эта информация не была такой уж неожиданной, по крайней мере для тех, кто следил за ситуацией. Мы получили подтверждение и узнали детали, и эти детали поражают. Поражает секретность. Я не думаю, что вещи, которые происходят, можно было бы предотвратить, если бы мы знали о них больше, потому что в мире после 11-го Сентября все, о чем попросит правительство, будет одобрено. Так что, к сожалению, это не нанесло большого урона, по крайней мере сразу. Приняли один (предотвращающий сбор и хранение АНБ всех метаданных телефонных звонков в США) незначительный закон. Однако обществу стало известно о государственной слежке. Это изменило общественное восприятие. Теперь люди знают об этом и их это волнует. Чтобы ощутить все последствия может потребоваться еще лет десять, но, в конечном счете, благодаря этому, политика измениться к лучшему".

Я спросил Шнайера, что по его мнению является наибольшей проблемой в компьютерной безопасности, и он сказал: "Слежка корпораций! Корпорации

хотят следить за нами больше, чем правительства. Это Facebook и Google, которые следят за людьми против их воли, а ФБР может потребовать копии, независимо, хотят ли того корпорации. Контролирующий капитализм - это реальная, фундаментальная проблема".

Я спросил Шнайера, над какой книгой он работает в данный момент (Он всегда работает над новой книгой). Он ответил: "Я думаю над написанием новой книги о физических проблемах кибербезопасности, таких, как интернет вещей, и как все меняется, насколько компьютеры, на самом деле, становятся опасны. Одно дело, когда электронная таблица имеет уязвимость, ломается ("крашится") или ее взламывают. Другое, когда это ваша машина. Слабая компьютерная безопасность убьет людей. Это все меняет! В прошлом месяце я выступал на эту тему перед Конгрессом (США). Сказал, что пора отнестись к ней серьезно. Игры кончились. Нам нужно ввести механизмы регулирования. На кону человеческие жизни! Мы не можем позволить себе такого же отвратительного, глючного ПО. Но индустрия не готова отнестись к этому серьезно, а придется. Как люди, могут работать над улучшением программной безопасности автомобиля, при том, что мы так и не смогли остановить хакеров и исправить уязвимости? Что-то должно изменится. И это изменится".

На протяжении десятилетий Брюс Шнайер остается признанным авторитетом в мире информационной безопасности, и продолжает выступать на передовой в самых важных обсуждениях. Если вас интересует информационная безопасность, то пусть для вас он тоже станет неофициальным наставником.

Подробнее о Брюсе Шнайере

Подробнее о Брюсе Шнайере вы можете найти на этих ресурсах:

- Блог Брюса Шнайера: https://www.schneier.com/
- Новостная рассылка Брюса Шнайера Crypto-Gram: https://www.schneier.com/crypto-gram/
- Книги Брюса Шнайера: https://www.amazon.com/Bruce-Schneier/B000AP7EVS/

Глава 4. Социальная Инженерия

В компьютерном мире, социальную инженерию можно описать, как обманные действия, вынуждающие людей сделать что-то, что причиняет вред им самим или окружающим. Социальная инженерия - одна из самых распространенных техник хакеров, потому что, в большинстве случаев, она оказывается эффективной. Также очень часто защитники оказываются перед ней бессильны, так как ее воздействие нельзя предотвратить, используя только технологии.

Методы Социальной Инженерии

Социальная инженерия может осуществляться множеством способов, включая использование компьютера, телефонных звонков, личного контакта, или с помощью традиционной, обычной почты. Существует так много способов и разновидностей социальной инженерии, что в любом списке, претендующем, на то, чтобы перечислить их все, какие-то методы все равно будут отсутствовать. Хотя социальная инженерия специализируется на получении доступа к компьютеру, как правило, она осуществляется с помощью электронной почты или через интернет (хотя для этого также используются мессенджеры и почти все остальные компьютерные программы).

Фишинг

Одной из главных целей социальной инженерии является получение учетных данных пользователя. Для этого можно использовать *фишинг*. Фишинговые электронные письма или веб-сайты пытаются обмануть пользователя, притворяясь настоящим веб-сайтом или администратором, с которым знаком конечный пользователь, чтобы последний предоставил свои учетные данные. Один из главных приемов фишинга - отправка пользователю электронного письма, якобы от администратора, в котором говориться, что пользователь должен подтвердить свой пароль, иначе его доступ к сайту будет ограничен.

Целевой фишинг - это разновидность фишинга, направленного против конкретного человека или группы людей, с использованием непубличной информации, с которой знакома цель. Например, менеджеру проекта приходит

электронное письмо с документом от, якобы, участника проекта, над которым они работают, и, когда менеджер открывает этот документ, то выполняются вредоносные команды. Целевой фишинг часто используется против высокопоставленных сотрудников корпораций.

Троянские Программы

Другая, не менее популярная, уловка социальной инженерии заключается в том, чтобы ничего не подозревающий конечный пользователь запустил троянскую программу. Это может быть сделано через электронную почту, с использованием как прикрепленного файла, так и URL-адреса. Так же часто это делается с помощью веб-сайтов. Часто настоящий веб-сайт взламывают, и, когда доверчивый пользователь загружает страницу, то ему сообщают, что нужно запустить файл. Этим файлом может быть «необходимое» расширение от сторонней компании, поддельный антивирус, или «необходимый» патч. Взломан может быть, как сам веб-сайт или независимый от него элемент, например, сторонний рекламный баннер. В любом случае, у пользователя, который много лет без проблем заходил на настоящий сайт, нет оснований подозревать, что сайт, которому он доверяет, был взломан.

Телефонный Звонок

Мошенники могут также звонить пользователям, представляясь технической поддержкой, популярным производителем или сотрудниками правительственного агентства.

Одним из самых популярных трюков является звонок пользователю из, якобы, технической поддержки, во время которого, пользователя уверяют, что на его компьютере обнаружена вредоносная программа. Затем они просят пользователя скачать “антивирус”, который, как ни удивительно, обнаруживает бесконечное количество вредоносных программ. Затем они просят пользователя скачать и запустить программу удаленного доступа, с помощью которой, “человек из техподдержки” устанавливает еще больше вредоносных программ на компьютере жертвы. Лже-программа от технической поддержки заканчивает свою работу, когда жертва покупает поддельный антивирус, используя номер своей кредитной карты.

Телефонные мошенники могут также представиться сотрудниками налоговой, правоохранительных органов, или других государственных служб, которые

должны взыскать оплату, чтобы конечный пользователь смог избежать строго наказания или тюрьмы.

Развод при Покупках

Еще одна популярная форма мошенничества используется во время покупки или продажи товаров на веб-сайтах, например, сайтах-аукционах, или таких, как Craigslist. Невинная жертва либо что-то покупает, либо продает.

При «разводе» во время продажи покупатель быстро отвечает, как правило, предлагает оплатить всю стоимость плюс доставку, и просит продавца воспользоваться услугами "доверенного" эскроу-агента. Затем они отправляют жертве поддельный чек, в котором сумма больше оговоренной, и который жертва размещает в своем банковском счете (К сожалению, банки, охотно принимают такие чеки, но, в конечном счете, виноватой за потерю денег остается жертва). Покупатель просит жертву-продавца вернуть "дополнительные" деньги своему поставщику или эскроу-агенту. Как правило, в конечном счете у жертвы-продавца не остается даже и этой суммы.

При «разводе» во время покупки жертва отправляет деньги, но не получает товар. При покупке средний доход мошенников составляет по меньшей мере тысячу долларов. При продаже он может измеряться десятками тысяч долларов.

Личное Общение

Одним из самых печально известных способов применения хакерами социальной инженерии является личное общение. В следующей главе мы поговорим о бывшем хакере Кевин Митник. Десятилетия назад он был одним из самых наглых социальных инженеров. Митник только и думал о том, чтобы переодеться мастером по ремонту телефонов или сервисным работником, и затем проникнуть в охраняемую зону. Физический доступ с помощью социальной инженерии включает в себя установку кейлоггера на терминал сотрудника, притворившись сервисным работником. Люди не доверяют незнакомцам, однако если незнакомец представляется ремонтником, особенно, когда он говорит что-то вроде: "Я слышал, Ваш компьютер в последнее время медленно работает". Кто будет спорить с этим утверждением? Ремонтник совершенно точно осведомлен о текущей проблеме, и он тут, чтобы ее устранить.

Кнут или Пряник

Как правило, конечного пользователя запугивают наказанием, если он чего-то не сделает или обещают награду, если он что-то сделает. Уловка начинается с того, что на жертву начинают давить, поскольку в стрессовой ситуации люди неадекватно оценивают риски. Жертва должна либо заплатить штраф, либо отправиться в тюрьму. Она должна запустить программу или рискует оставить свой компьютер зараженным и потерять все деньги на банковском счете. Она должна отправить деньги или дорогой ей человек останется в иностранной тюрьме. Она должна сменить пароль своего начальника или у нее будут с ним проблемы.

Одна из моих любимых уловок социальной инженерии во время пентеста - отправка электронного письма сотрудникам компании, где я притворяюсь CEO или финансовым директором и объявляю о слиянии компании со своим крупнейшим конкурентом. Я пишу, что нужно открыть прикрепленный документ-ловушку и посмотреть, затронет ли это слияние их должности. Или я отправляю электронное письмо, похожее на настоящее, сотрудникам-мужчинам, притворяясь адвокатом их бывшей жены, и прошу увеличить алименты. Вы удивитесь, насколько успешно работают такие трюки с электронной почтой.

Защита от Социальной Инженерии

Защита от социальной инженерии требует комбинации технологий и подготовки людей.

Подготовка

Самой лучшей и наиболее эффективной защитой против социальной инженерии является специальная подготовка. Такая подготовка должна включать примеры самых распространенных приемов социальной инженерии, и то, как потенциальная жертва может обнаружить признаки незаконных действий. В моей нынешней компании каждый сотрудник обязан просматривать специальные видео по социальной инженерии, и каждый год проходить небольшой тест. Наиболее успешные результаты давали консультации с самыми умными, доверенными и всеми любимыми сотрудниками, которые делились своим личным опытом, попавшись на определенный тип уловок социальной инженерии.

Я думаю каждая компания должна проводить поддельные фишинговые кампании, в которых их сотрудникам рассылались бы электронные письма, похожие на фишинг, запрашивающие их данные для входа. Сотрудники,

предоставившие свои данные, должны пройти дополнительную подготовку. Существует множество ресурсов, как коммерческих, так и бесплатных, проводящих поддельные фишинговые кампании, само собой, коммерческие проще использовать, и они дают лучший результат.

Обучение тактикам социальной инженерии нужно проводить среди всех пользователей компьютеров. Люди, покупающие и продающие товары в интернете должны знать о мошенниках и способах обмана. Они должны пользоваться только официальными эскроу-сервисами, и следовать всем рекомендациям сайта по осуществлению безопасной сделки.

Осторожность при Установке ПО со Сторонних Веб-сайтов

Пользователей нужно научить тому, что нельзя устанавливать программное обеспечение напрямую с сайтов, если это не сайты производителей написавших это ПО. Если веб-сайт просит вас установить стороннее ПО, чтобы продолжить его просмотр, и вы считаете, что так и должно быть, закройте этот веб-сайт и перейдите на сайт производителя ПО и скачайте его оттуда. На исходном сайте, на самом деле, может быть и настоящее ПО, но риск слишком велик.

Цифровые сертификаты EV

Интернет-пользователи должны знать, что на большнстве популярных сайтов есть цифровые сертификаты (“extended validation”, EV) расширенной проверки (https://en.wikipedia.org/wiki/Extended_Validation_Certificate). Часто веб-сайты с EV выделяются в адресной строке (обычно, зеленый либо сам адрес, либо область рядом с ним), что является для пользователя подтверждением URL-адреса и подлинности данного веб-сайта. Пример сайта с EV вы можете увидеть по ссылке https://www.bankofamerica.com.

Избавьтесь от Паролей

Фишинг учетных данных не работает, если сотрудник не может дать своих учетных данных. Простые имена пользователей и пароли уходят в прошлое и заменяются на двухфакторною аутентификацию (2FA), цифровые сертификаты, устройства для входа, аутентификация по внешнему каналу и другие методы авторизации где фишинг невозможен.

Технологии Противодействия Социальной Инженерии

Большинство антивирусов, программ для безопасного интернет-серфинга и блокировки спама пытаются минимизировать эффект от социальной инженерии при использовании компьютеров. Программы безопасности пытаются обнаружить вредоносные файлы. Интернет-фильтры пытаются определить вредоносные веб-сайты, когда браузер пользователя загружает страницу. Анти-спам часто полностью фильтрует социальную инженерию. Однако, технологии никогда не будут совершенными, поэтому должна применяться совокупность методов, таких, как подготовка конечного пользователя и т.д.

Социальная инженерия - очень успешный хакерский метод. Некоторые эксперты в информационной безопасности скажут вам, что невозможно подготовить всех сотрудников. Всегда найдется тот, который попадется на уловку хакера. Они не правы. Сочетание достаточной подготовки и правильных технологий может значительно уменьшить риски от социальной инженерии.

В следующей главе речь пойдет об эксперте в социальной инженерии, Кевине Митнике. Его опыт и знания социальной инженерии помогают ему защитить своих клиентов на протяжении десятелетий.

Глава 5. Профиль: Кевин Митник

Когда произносят термин "компьютерный хакер", большинство людей вспоминает Кевина Митника. В 1970-ые, 1980-ые и 1990-ые Кевин Митник был тем самым хакером. Митник использовал сочетание социальной инженерии и исследовал операционные системы на низком уровне чтобы проворачивать всевозможные фокусы, тем не менее общий вред, нанесенный им, довольно спорный, особенно, по сравнению с современными APT-атаками (целевыми кибератаками) и программами-вымогателями.

О нем и его эксплойтах написано несколько книг, снят фильм, а также, благодаря ему, появилась целая хакерская субкультура эксцентричных историй, связанных с ним, ни одна из которых, не является правдой. Страх правительства перед Митником был настолько велик, что он был единственным заключенным в США, которому не позволяли использовать телефон в тюрьме и держали в изоляции, поскольку боялись, что одно его слово или звук может запустить ядерную ракету. Если вы когда-нибудь видели фильм, где главный герой говорит одно слово по телефону, и затем начинается повсеместный киберужас, эта сцена стала результатом паранойи, окружающей Митника.

Я решил рассказать о Митнике в самом начале книги, потому что с момента появления киберугроз, он посвятил свою жизнь сражению с компьютерной преступностью, и он один из нескольких хакеров, перешедших на светлую сторону, которым я полностью доверяю. На сегодняшний день Митник написал несколько книг по информационной безопасности (https://www.amazon.com/s/ref=dp_byline_sr_book_1?ie=UTF8&text=Kevin+Mitnick&search-alias=books&field-author=Kevin+Mitnick&sort=relevancerank), работает с несколькими компаниями (включая KnowBe4), имеет свою собственную консалтинговую фирму (Mitnick Security Consulting), имеет самый загруженный график выступлений из всех специалистов в информационной безопасности, которых я знаю, был на шоу *Отчет Кольбера*, и даже выступил в камео на популярном телешоу *Alias*. Благодаря урокам Митника, индустрия начала серьезней относиться к социальной инженерии, и к тому, как с ней бороться. В конце концов, если вы хотите остановить преступника, то не будет лишним пообщаться с умным бывшим преступником.

Я спросил Митника, как он стал хакером. Он сказал: "Меня с детства интересовала магия. Я любил магию. В школе один парень показал мне несколько трюков с телефоном, например, как бесплатно осуществлять звонки на дальние расстояния, как найти чей-то адрес, зная только телефонный номер, переадресация звонков и так далее. Мы заходили в телефонную будку, звонили

кому-нибудь (телефонной компании) прикидывались кем-то другим, и происходила магия. Это был мой первый опыт социальной инженерии. Для меня это было магией. Я не знал, что это называлось телефонным мошенничеством и социальной инженерией. Я знал только, что это было забавно и увлекательно, и я погрузился в это с головой. Вот, что я тогда делал. В школе мне было скучно, а из-за того, что я всю ночь занимался телефонным мошенничеством, мои оценки начали ухудшаться".

Я спросил, что его родители думали о его хакерской деятельности. Он ответил: "Ну, поначалу они ничего и не знали. Или, возможно, они думали, что я занимался чем-то странным по телефону. Но моя мать должно быть думала: "Что страшного можно сделать по телефону, кроме как доставать кого-то?". Но, на самом деле, мои родители понятия не имели, чем я занимался, пока мама не получила письмо от AT&T, в котором они говорили, что отключают нам телефон. Она была очень расстроена. И помните, тогда еще не было сотовых телефонов. Домашний телефон был единственным средством связи с другими людьми. Я сказал ей, чтобы она успокоилась, и что я все исправлю.

По сути, я вернул нам телефон с помощью социальной инженерии. Во-первых, я выдумал новый адрес. Мы жили в доме №13. Я позвонил в служебное подразделение телефонной компании, притворившись другим человеком, и выдумал дом 13B. Я подождал несколько дней, пока новый дом подключат к системе, затем позвонил в отдел предоставления услуг и попросил, чтобы новый телефон установили в доме 13B. Я даже пошел в магазин хозтоваров и купил букву B, чтобы повесить ее на двери рядом с номером. Я позвонил, притворившись новым клиентом по имени Jim Bond из Англии. Я дал им настоящий телефонный номер из Англии, который я нашел с остальной информацией, потому что знал, что они не смогут проверить информацию из-за рубежа. Затем я спросил, могу ли я выбрать "красивый телефонный номер", и они сказали: "да", и я выбрал номер, который заканчивается на 007. В конце я спросил, могу ли я использовать псевдоним Джим или мне нужно предоставить полное настоящее имя. Они сказали, что нужно настоящее, и я сказал, что меня зовут Джеймс. Таким образом, AT&T зарегистрировали меня под именем Джеймс Бонд, и мой телефонный номер заканчивался на 007, а моей матери вернули телефон. В AT&T взбесились, когда, в конце концов, узнали о моей схеме".

В этот момент я понял, что в нашем интервью еще не было ни слова о компьютерных взломах. Это были только трюки с телефоном. Я спросил, как он начал взламывать компьютеры. Он ответил: "В старших классах был парень, который знал, что я занимался телефонным мошенничеством, и он подумал, что меня заинтересуют продвинутые уроки информатики, которые преподавались в школе. Сначала я ответил, что мне это не интересно, но тот парень сказал: "Знаешь, я слышал телефонные компании переходят на компьютеры". И для меня этого было достаточно. Мне нужно было разобраться с этими компьютерами".

“Мне пришлось пойти к учителю по информатике, мистеру Крису, и спросить, могу ли я ходить на его уроки, потому что я совсем не соответствовал необходимым условиям (тогда это были углубленные математика и физика), и хороших оценок у меня тоже не было, поскольку я все время не высыпался. Мистер Крис сомневался, стоит ли меня брать, и тогда я показал ему телефонные фокусы, сообщим ему его собственный номер телефона, которого не было в справочнике, и номера его детей. Он сказал: “Это магия!” и пустил меня на урок.

Сперва мы изучали Fortran, который предназначался для высчитывания чисел Фибоначчи. Это показалось мне слишком скучным. Я даже пошел в местный университет, Нортридж, и попытался получить время для работы за компьютером. У них были такие же компьютеры на которых была установлена та же самая операционнаяя система. Но там мне нельзя было сидеть дольше пяти минут. Тогда я пошел к руководителю компьютерного класса и попросил больше времени. Он сказал, что я даже не студент колледжа и меня здесь вообще не должно быть, но, при этом, он видел, насколько сильно я интересовался компьютерами, и, чтобы вдохновить меня, он дал мне свои личные логин и пароль, чтобы я мог попрактиковаться. Вы можете в это поверить? Вот как тогда относились к компьютерам”.

“Когда пришел день показать Мистеру Крису, сколько чисел Фибоначчи высчитали наши программы, у меня ничего не было. Мистер Крис отругал меня перед всем классом за то, что разрешил мне ходить на уроки и взял на себя риск, а у меня даже нечего показать. Весь класс смотрел только на меня. Я сказал: “Ну, я был слишком занят написанием программы взлома пароля и ваш пароль johnco!” Он сказал: “Как ты это сделал?” Я объяснил ему, и он поздравил меня и сказал всему классу, что у меня талант в компьютерах. И он совсем не злился. Возможно, это был самый плохой первый урок по этике.

Я спросил Митника, что должны делать родители, если они видят, что их ребенок занимается хакингом. Он предложил: “Покажите им, как взламывать легально. Перенесите их интерес на легальные и этичные возможности, например, посещение конференций по компьютерной безопасности и участие в соревнованиях по “захвату флага”. Родитель должен бросить ребенку вызов, сказав что-то, вроде: “Так ты думаешь, ты достаточно хорош, чтобы принять участие в соревновании по захвату флага?”. Родитель может применить к ребенку социальную инженерию, и ребенок получит то же удовольствие и воодушевление, но легальным способом. Буквально сегодня, я легально взламывал компанию, и это было так же волнующе, как нелегальный и неэтичный взлом. Я бы хотел, чтобы им были доступны все законные способы взлома, которые сейчас есть. Я бы хотел вернуться во времени и сделать все по-другому. Вы знаете в чем единственная разница между законным и незаконным взломом? Написание отчетов!”.

Я поинтересовался у Митника, у которого есть опыт работы по обе стороны закона, что он чувствует по поводу права правительства знать что-то о человеке,

нарушая его право на частную жизнь. Он сказал: "Я думаю мы все имеем полное право на частную жизнь. На самом деле, моя последняя книга *The Art of Invisibility* (https://www.amazon.com/Art-Invisibility-Worlds-Teaches-Brother/dp/0316380504/), полностью посвящена тому, как можно сохранить свою частную жизнь в тайне. Я думаю очень сложно сохранить частную жизнь в тайне от таких структур, как АНБ или правительства, с их бесконечными деньгами. Я имею ввиду, если они не смогут взломать ваше шифрование, то они просто могут использовать одну из многочисленных уязвимостей нулевого дня, или купить ее и в итоге взломать вас. За 1.5 миллиона долларов можно купить уязвимость нулевого дня для Apple, за полмиллиона можно купить уязвимость нулевого дня для Android, и так далее. Если у вас есть средства и ресурсы, то можно получить любую информацию. Тем не менее, я думаю в книге *The Art of Invisibility*, у меня есть способ, который работает даже против влиятельных людей, но он очень сложный и включает много действий с OPSEC. Но его можно применить так, что, думаю, даже АНБ или правительству будет сложно обойти его. Я понимаю, что в некоторых случаях, таких как терроризм, правительство должно знать подробности, но они хотят знать все обо всех. И, если за вами следят, вы меняете свое поведение, а это означает, что у вас меньше свободы. Не думаю, что свобода возможна отдельно от частной жизни."

Я закончил интервью, напомнив Митнику, что мы как-то однажды встречались на конференции по безопасности много лет назад, где он поднимался на сцену после меня. Как только он прошел мимо меня он понял, что ему нужна USB-флэшка, чтобы запустить презентацию на ноутбуке, который стоял на сцене. У меня была одна в кармане, и я предложил ее. Он почти взял ее, но после нескольких секунд раздумий, пересмотрел свое мнение, вернул флэшку, и сказал, что не доверяет ни чьему USB-ключу. Несколько человек вокруг усмехнулись от его паранойи. В конце-концов, нельзя заразиться через USB-устройство, точнее, в то время почти все так думали.

Однако, никто не осознавал, что к тому моменту я открыл способ автоматически запустить любую программу с любого портативного устройства (используя трюк со скрытым файлом, который называется desktop.ini, который потом использовался в программе Stuxnet), и, по чистой случайности, на той флэшке была демонстрационная версия этого эксплойта. У меня не было намерений специально заразить Митника. В то время на всех моих флэшках был этот эксплойт, и я нечаянно предложил одну из них.

Постоянная паранойя Митника спасла его от уязвимости нулевого дня, которую я обнаружил. Это также доказывает, что сложно обмануть профессионального социального инженера, который еще пребывает в расцвете сил.

Подробнее о Кевине Митнике

Подробнее о Кевине Митнике вы можете найти на этих ресурсах:

- Веб-сайт Кевина Митника: https://mitnicksecurity.com/
- *Ghost in the Wires*: https://www.amazon.com/Ghost-Wires-Adventures-Worlds-Wanted/dp/0316037729/
- *The Art of Invisibility*: https://www.amazon.com/Art-Invisibility-Worlds-Teaches-Brother/dp/0316380504/
- *The Art of Deception*: https://www.amazon.com/Art-Invisibility-Worlds-Teaches-Brother/dp/0316380504/
- *The Art of Intrusion*: https://www.amazon.com/Art-Intrusion-Exploits-Intruders-Deceivers/dp/0471782661/
- KnowBe4's Kevin Mitnick Security Awareness Training: https://www.knowbe4.com/products/kevin-mitnick-security-awareness-training/
- Kevin Mitnick's Slashdot Q&A: https://news.slashdot.org/story/11/09/12/1234252/Kevin-Mitnick-Answers

Глава 6. Уязвимости программного обеспечения

Уязвимости программного обеспечения - это недостатки (“баги”) ПО, часто являющиеся результатом ошибки разработчика, или частью языка программирования. Но к уязвимостям безопасности относятся не все баги. Баг должен быть использован злоумышленником, чтобы стать угрозой или риском. Большинство программных багов вызывают проблемы при использовании (которые даже не будут сразу заметны пользователю), или могут даже полностью остановить работу программы, но, злоумышленник не сможет их использовать, чтобы получить несанкционированный доступ к системе.

Очень большой (если не самый большой) процент взломов в данный период времени - это результат программных уязвимостей, которые можно использовать, и это при том, что другие методы взлома (например, троянские программы и социальная инженерия) также очень эффективны. Некоторые эксперты в информационной безопасности думают, что большинства проблем с безопасностью можно было бы избежать, если бы все ПО было без багов, хотя это неправда, и это невозможно. Тем не менее, хоть это и не панацея, более безопасный код исправил бы много проблем связанных со взломом и сделал бы информационную среду значительно безопасней.

Количество Программных Уязвимостей

Существует огромное количество ресурсов для отслеживания общеизвестных программных уязвимостей, однако количество багов на каждом из них может значительно отличаться. В среднем, каждый год крупные разработчики ПО и тестировщики публично анонсируют 5000 - 6000 новых программных уязвимостей. Это, примерно, 15 багов в день. Сервис Common Vulnerabilities and Exposures (CVE) (база данных общеизвестных уязвимостей), можно посмотреть по адресу http://cve.mitre.org, а их списки (http://cve.mitre.org/data/downloads/index.html) по отслеживанию публичных уязвимостей считаются объемными, доверенными и независимыми. Многие производители также сами отслеживают, как свои собственные уязвимости, так и общеизвестные. Посмотрите отчет Microsoft *Security Intelligence Report* (http://www.microsoft.com/sir), если хотите узнать о самых последних уязвимостях и получить подробный анализ.

Конечно, это только те баги, о которых известно всем. Многие производители не делают публичных сообщений о каждом новом баге. Многие не сообщают о багах, которые нашли сами, или тех, которые были исправлены перед релизом. Хотя это и никак нельзя подтвердить многие эксперты думают, что "реальное" количество багов значительно больше известного общественности.

ПРИМЕЧАНИЕ Количество программных уязвимостей не отображает общей картины безопасности программы или системы. Единственное, что важно - какой урон нанесли программные уязвимости. Количество программных уязвимостей может уменьшаться, но нанесенный вред может увеличиваться, хотя, в целом лучше использовать более безопасные программы.

Почему Программные Уязвимости До Сих Пор Остаются Большой Проблемой?

В наше время производителям нужно от нескольких часов до нескольких дней на выпуск патча, который исправит критическую уязвимость. Почему же программные уязвимости до сих пор остаются серьезной проблемой, с учетом того, что у большинства производителей есть механизмы автообновлений, с помощью которых можно быстро установить патч? Проблема в том, что на огромную часть компьютеров патчи устанавливаются не сразу, а во многих случаях, вообще не устанавливаются. И, каждый патч может испортить работу программы, иногда настолько, что это вызовет больше расстройства у конечного пользователя, чем потенциальный урон от самого бага.

Количество всех эксплойтов довольно велико и постоянно. Значительная часть компьютерного администрирования заключается в распределении и установке патчей. На это тратится огромное количество времени, денег и других ресурсов, которые можно было бы использовать более продуктивно. Даже когда пользователи и администраторы вовремя устанавливают патчи, существует период времени между тем, как производитель выпустит патч и пользователь или администратор установит его, и хакер может его использовать для успешной атаки на данную систему. Если я терпеливый, настойчивый хакер, заинтересованный в конкретной цели, я могу просто подождать, пока производитель анонсирует новый патч, и использовать его для взлома своей цели.

Когда производители выпускают патчи, как белые, так и черные шляпы анализируют их, стараются найти уязвимость, которую он закрывает. Затем они создают эксплойты, и начинают использовать уязвимость. Существуют десятки коммерческих компаний, несколько бесплатных сервисов, и неизвестное

количество хакеров, которые занимаются этим каждый день. Можно купить и/или скачать сканеры уязвимостей, каждый из который будет сканировать определенное устройство и сообщать об открытых уязвимостях. Часто у самих этих сканеров уязвимостей есть тысячи и тысячи уязвимостей. По всему миру есть множество хакерских веб-сайтов, с тысячами индивидуальных эксплойтов, которые можно скачать и которыми можно воспользоваться. Один из самых популярных бесплатных инструментов, который используют как белые, так и черные шляпы - это Metasploit (https://www.metasploit.com/).

Защита от Программных Уязвимостей

Самая главная защита от программных уязвимостей - это более качественная подготовка разработчиков и использование более безопасных языков программирования.

Жизненный Цикл Безопасной Разработки

Сейчас процесс снижения количества программных уязвимостей известен, как Security Development Lifecycle (SDL). SDL фокусируется на каждом компоненте жизненного цикла программы, начиная от изначальной разработки, до выпуска патчей под недавно обнаруженные уязвимости, чтобы сделать ПО более безопасным. Хотя это не было придумано в Microsoft, эта корпорация, Microsoft, проделала большую часть работы и предоставила огромное количество свободной информации и инструментов в этой области (https://www.microsoft.com/sdl), чем кто либо другой. Склонность людей к ошибкам доказывает, что в программном коде всегда будут уязвимые баги, но используя SDL, мы можем уменьшить их количество (при этом оставив столько же строк кода).

ПРИМЕЧАНИЕ Доктор Дэниэл Дж. Бернштейн (https://en.wikipedia.org/wiki/Daniel_J._Bernstein) - профессор колледжа, который пишет и продвигает концепцию невероятно безопасного кода. Он создал несколько бесплатных и широко используемых программ, таких как dbjdns и qmail, в которых пристуствет очень небольшое количество багов. Он даже предлагает тестировщикам заплатить из своего собственного кармана если они найдут уязвимости. Он верит, что публичное анонсирование багов, перед тем,

как производители смогут их проанализировать и выпустить патч, сильно затрудняет работу последних.

Более Безопасные Языки Программирования

Более безопасные программы не могут создаваться без более безопасных языков программирования. В течение многих лет, большинство языков программирования стремилось к тому, чтобы сделать свои стандартные версии более безопасными. Такие языки пытаются снизить количество или полностью избавиться от общих причин эксплойтов. На данный момент, у них многое получилось, и программы, написанные на этих языках взломать значительно сложнее, чем те, которые написаны на менее безопасных языках.

Анализ Программы и Кода

После того, как версия программы написана, она всегда должна быть проанализирована на известные и узнаваемые баги. Такой анализ может быть выполнен человеком или программными инструментами. Анализ, проводимый человеком, как правило, менее эффективен, так как за час обнаруживается меньше багов, но такой анализ может обнаружить гораздо более уязвимые баги, нахождение которых, не запрограммировано в специальных инструментах. ПО для обнаружения багов часто классифицируется на "статический анализ" и "фаззеры". Статический анализ проверяет исходный код (или программу) на известные программные баги в самом коде. Фаззеры вводят неожиданные данные, проверяя программу на уязвимости во время работы. Многие нашумевшие охотники за багами, включая Чарли Миллера, представленного в Главе 36, полагались на фаззинг во многих своих открытиях.

Более Безопасные Операционные Системы

Большинство операционных систем не только написаны программистами, погруженным в философию SDL, и использующими более безопасные языки программирования, но также имеют встроенную защиту от многих эксплойтов. Большинство из популярных сегодня операционных систем включают в себя специально разработанную защиту памяти, и они защищают наиболее критичные области операционной системы. В некоторых даже есть встроенная

защита от переполнения буфера, антивирусы и файрволы, которые помогают снизить количество уязвимых багов или последующий урон от их использования.

Защиты от Сторонних Производителей и Дополнения от производителя

Существуют тысячи программ, способных, хотя бы немного, защитить компьютерную систему от ранее известных программных уязвимостей. Некоторые из них предоставляются производителем бесплатно или в качество платных дополнений, другие - отдельными сторонними компаниями. Программы, обещающие обнаружить и остановить новые эксплойты, встречаются часто, и, хотя они не идеальны, они могут существенно снизить риск новых угроз. Один из моих любимых принципов работы защитного ПО называется "контроль приложений" или "белый лист". Такие программы не остановят первоначальное проникновение, но могут остановить или усложнить хакеру, или вредоносному ПО, нанесение дальнейшего урона.

Идеальное ПО Не Вылечит Все Болезни

Никакая защита не будет лучше, чем безопасное написание приложений, в которых изначально содержится меньшее количество багов. Однако, существование идеальных программ без багов невозможно, а даже, если бы было возможно, это все равно бы не вылечило болезнь взлома. К сожалению, программные уязвимости - не единственная наша проблема. Трояны работают после того, как пользователь запустит далеко не самую вредоносную программу. Многие хакеры и вредоносное ПО используют для нанесения вреда заложенные изначально, то есть настоящие свойства данных языков программирования и других компонентов. А социальная инженерия может сделать то, чего не может программное обеспечение.

И все же, никто не спорит, что более безопасные программы нам не помогут. В Главах 7 и 8 представлены два эксперта, посвятивших свою жизнь доведению программного обеспечения до идеала. В Главе 7 речь пойдет про Майкла Ховарда, который популяризировал более безопасное написание программ, а Глава 8 фокусируется на Гэри МакГроу, одном из лучших тестировщиков в мире.

Глава 7. Профиль: Майкл Ховард

Майкл Ховард – заразителен. Он талантливый преподаватель, сильный оратор, и спустя почти 20 лет все так же любит профессию специалиста по компьютерной безопасности и безопасному написанию кода. Сложно находиться рядом с ним дольше нескольких минут и, при этом, не захотеть сделать мир безопаснее хотя бы на одну строку кода. Впервые, он стал широко известен благодаря соавторству книги *Writing Secure Code* (https://www.amazon.com/Writing-Secure-Code-Michael-Howard/dp/0735615888) с Дэвидом Лебланом, а также благодаря тому, что стал причиной, по которой в Microsoft стали намного тщательнее относиться к написанию более безопасного кода. Ховард, родившийся в Новой Зеландии, сейчас живет в Остине, Техас, стал соавтором нескольких книг по написанию более безопасного кода, и постоянно ведет блог.

ПРИМЕЧАНИЕ Дэвид Леблан, соавтор Ховарда по книге *Writing Secure Code* – еще один прогрессивный специалист по информационной безопасности. Благодаря его усилиям Microsoft Office стал значительно безопаснее. Он так же создал более безопасную модель браузера, которую в итоге используют Google, Adobe и Microsoft.

Я спросил Ховарда, как он попал в информационную безопасность. Он ответил: "Я работал над самыми ранними версиями Windows NT для Microsoft. У меня были довольно низкоуровневые задачи, такие как контроль доступа, криптография и настройка GINA (графические интерфейсы, которые использовались для проверки подлинности в Microsoft Windows и другими поставщиками проверки подлинности). Это на самом деле привело меня к тому, что я начал больше задумываться о безопасности в будущем. Примерно в 2000-х стало понятно, что наличие программных компонентов для безопасности не делают продукт безопасным; скорее нужно сфокусироваться на том, чтобы сами эти программные компоненты были безопасными, а это уже совершенно другое направление".

Я спросил его, как концепция SDL появилась в Microsoft. Он сказал: "Со временем команды, работающие над .NET Framework, Windows, Office, SQL Server и другие, вовлеченные в Security Development Lifecycle (SDL) изучили различные подходы безопасности. Концепция SDL помогла популяризировать безопасное написание кода и безопасную разработку, а также поддержание продукта, и сейчас является лидирующей силой, демонстрирующей, сколько компаний готовы улучшить безопасность своего ПО".

Я поинтересовался, было ли SDL небольшим улучшением того, о чем он прочел, или SDL - это то, что он построил с нуля, без предварительных упоминаний где-то еще. Он ответил: "Все строят свою деятельность, основываясь на работе других, но по большей части SDL - это результат действий и обучения. То, что останавливает работу, и то, что не работает или совершенно не прагматично выбрасывается. Иногда, мне кажется, что некоторые академические модели вообще никогда не проверялись в производственной среде, где есть дедлайны, требования к производительности, время выхода на рынок, экономические соображения, требования обратной совместимости, и так далее.

В то время была широко распространена точка зрения, что если просто улучшить общее качество кода, то это также улучшит непосредственно его безопасность. Но я не видел этому еще ни одного эмпирического доказательства. Можно написать функциональный SQL-код, который пройдет все функциональные тесты, но при этом может быть пронизан уязвимостями для SQL-инъекций. Если вас никогда не готовили к SQL-инъекциям, то вы увидите лишь идеальный код - он делает то, что должен. Безопасная система должна делать только то, что нужно, и ничего больше. Уязвимость к SQL-инъекциям - это "дополнительный функционал", который делает код небезопасным".

Я спросил его, какую роль сыграло внедрение практики SDL в Microsoft. Он рассказал мне: "Это была синергия многих различных вещей, в которой участвовали как я, так и другие специалисты. Все началось в конце 2001-го, когда команда .NET провела мероприятие "security stand down", на котором решили рассмотреть текущие проблемы безопасности и потенциальные риски. Мы многому там научились и добавили много новых способов защиты. Я помню, у нас были футболки, на которых была напечатана дата мероприятия, а потом грянул сильный снегопад и его отменили… получилось иронично - мы искали новые способы обезопасить код, и, при этом, на всех наших футболках была неправильная дата. Но, благодаря этому мероприятию мы многому научились и, в конце концов, пришли к SDL. Вышла книга, авторами которой были я и Дэвид, после нее многие стали сильнее задумываться о безопасности кода. В 2001-ом Microsoft атаковало огромное количество хакеров и вредоносных программ. Черви Code Red и Nimda нанесли серьезный урон. После этого Билл Гейтс спросил о природе программных уязвимостей и почему они до сих пор пристуствуют. Меня выбрали участником команды, которая встречалась с Биллом Гейтсом. Я передал ему раннюю копию нашей книги *Writing Secure Code*, и после той встречи, Билл, в конечном итоге, написал свое знаменитую заметку "Надежные вычисления" (https://www.microsoft.com/mscorp/execmail/2002/07-18twc.mspx). В этой заметке Билл упомянул нашу книгу, после чего ее продажи моментально взлетели! В итоге я стал работать в только что сформированном подразделении Microsoft, которое называлось "Надежные вычисления". После этого мероприятия по безопасности стали проводится чаще (посвященные

Windows, SQL-серверу и многим другим продуктам Microsoft). Благодаря всему этому создавалась и обновлялась концепция SDL, а затем совершенствовалась и становилась более эффективной. И она продолжает ежегодно обновляться".

Я спросил правда ли, что он и Microsoft предоставили больше информации и инструментов по безопасному написанию кода, чем любая другая организация. Он сказал: "Совершенно точно, да! Но, что более важно, это те инструменты и техники, которые мы сами используем в наших продуктах, в миллионах строк кода каждый день. Это не академический учебник. Это то, что использует одна из самых крупных компаний в мире. И почти всем этим мы делимся".

Я спросил, если программисты по всему миру лучше подготовлены в информационной безопасности, то почему в мире не становится меньше публично анонсируемых уязвимостей. Он ответил: "Ну, само собой, сейчас больше ПО, и в нем больше строк кода. Но реальная проблема в том, что программистов все еще не обучают безопасному написанию кода, и они все еще не понимают основных угроз безопасности. В большинстве случаев образование оторвано от реальности. На днях я изучал учебную программу по информационной безопасности в одном университете, и почти 50% занятий фокусируются на низкоуровневых сетевых угрозах. Не было лекций по облачной безопасности и безопасному написанию кода. Наши колледжи все еще выпускают программистов, которые многого не знают об информационной безопасности или безопасном написании кода. Мне очень грустно от осознания того, что эти выпускники будут создавать критически важные системы, подключенные к интернету. Я по прежнему нахожу самые простые ошибки в коде людей. Когда я демонстрирую проблему нарушения целостности данных в памяти или уязвимость к SQL-инъекции - самые известные, самые базовые вещи - на меня смотрят так, как будто я показываю магию или делаю что-то особенное. Очень сложно найти нового программиста, который действительно понимает основы информационной безопасности, и я радуюсь, если кандидат хотя бы беспокоится об этом. Если глаза программиста широко открыты, когда я говорю о проблемах информационной безопасности, то я уже счастлив. Вы удивитесь, насколько много существует выпускников, для которых безопасность не имеет значения, и основная причина в том, что их не учат работать над безопасностью. Точнее их учат неправильным вещам, таким, как концентрация на сетевой безопасности или незначительных деталях. В школе ученики подробно изучают RSA-алгоритм, но их не учат тому, для чего его нужно использовать, какие проблемы он решает, и когда его нужно применять. Знать, как что-то использовать для решения реальных проблем безопасности гораздо важнее, чем знать, как что-то работает. Любой может запомнить протокол, но нам нужны люди, которые знают о рисках и думают о решении проблем. Преподаватели некоторых учебных заведений, такие, как Мэтт Бишоп из Калифорнийского университета в Дейвисе, обучают тому, что действительно нужно, но Мэтт и

остальные совершают героические усилия, благодаря которым это становится возможным. Он и профессора, которые похожи на него – настоящие герои".

Я поинтересовался, что может сделать сам программист, учитывая, что наши колледжи не дают соответствующей подготовки. Он сказал: "Всегда учитесь. У меня в календаре специально выделен час, где написано "Учись". И в течение этого часа я читаю/пишу код/экспериментирую с чем-то, чего не знаю. Я делаю это каждый день. Сколько себя помню, я всегда так делал. Во-вторых, если вы не получили достаточной подготовки по информационной безопасности, изучите эту тему. Зайдите на сайт CVE (http://cve.mitre.org/cve/), прочитайте о недавних багах, прочитайте внимательно. Затем напишите код с такой же уязвимостью и выясните, что нужно, чтобы предотвратить эту уязвимость, как на техническом уровне, так и на уровне процесса написания. Для начала, как появилась эта уязвимость, и как она оказалась в коде? И используйте эти уроки для того, чтобы в вашем коде не было таких багов".

Я спросил, как большинству компаний обезопасить свой код, что нужно делать помимо следования текущим советам SDL и использования доступных инструментов. Он ответил: "Нужно добиться того, чтобы программисты понимали реальные угрозы, а только о теоретические. И встроить процесс безопасности в процесс разработки так, чтобы плохой и небезопасный код даже не мог попасть в процесс разработки. Мы, в Microsoft называем это Воротами Качества. Хороший пример (небезопасного написания) - когда человек, который пишет код, думает, что у всех IP-адресов четыре октета. Это значит, что данный код не будет работать с чистым протоколом IPv6. Такой код даже не может пройти нашу проверку, потому что программа, которая запускается автоматически, находит эту проблему, и отклоняет сохранение кода на сервере. Вот, что мы подразумеваем под Воротами Качества. Но по безопасности, повторю, что в коде не должно быть уязвимостей к SQL-инъекциям, угроз безопасности памяти.

Если бы я выбирал самые основные подходы к безопасности, то это были бы:

- Разработчики должны научиться никогда не доверять введенным данным и проверять их на корректность, желательно используя протестированную и надежную библиотеку. Если вы считаете, что данные будут занимать не больше 20 байт, то поставьте ограничение в 20 байт. Если вы ожидаете какое-то число, то убедитесь, что это именно число и так далее.
- Менеджеры по дизайну/архитектуре/программированию должны научиться моделировать угрозы и проверять правильность защиты системы.
- И тестировщики должны доказывать, что разработчики ошибаются, создавая или покупая инструменты, создающие вредоносные и/или неверные данные. Цель в том, чтобы найти ошибки разработчиков, если они есть!

Безопасность ПО это не только то о чем я рассказал. Но на мой взгляд - это фундаментальные навыки безопасности, которыми должны обладать все разработчики ПО”.

Подробнее о Майкле Ховарде

Подробнее о Майкле Ховарде вы можете найти на этих ресурсах:

- Книги Майкла Ховарда: https://www.amazon.com/Michael-Howard/e/B001H6GDPW/ref=dp_byline_cont_book_1
- Блог Майкла Ховарда: https://blogs.msdn.microsoft.com/michael_howard/
- Твиттер Майкла Ховарда: https://twitter.com/michael_howard

Глава 8. Профиль: Гэри МакГроу

Когда я позвонил Гэри МакГроу, для того чтобы взять у него интервью, он сказал, что только что разговаривал с католическим монахом, который проходил мимо его дома в долине Шенандоа, Вирджиния. Через несколько секунд после начала разговора, монах заговорил о хитросплетениях информационной безопасности. Такие странные, неестественные парадоксы происходят с МакГроу всю его жизнь. Он начал программировать в 16 лет, в 1981-ом, на своем первом компьютере Apple II+. Он пошел в колледж, на факультет философии, и одновременно с этим получил музыкальное образование. Он даже дважды выступал в Карнеги-холл. Сегодня, будучи одним из лучших мировых экспертов в области информационной безопасности, он любит готовить, ухаживать за садом и смешивать новые коктейли.

Я спросил МакГроу, как он, являясь студентом факультета философии в Университете Вирджинии, стал интересоваться информационной безопасностью. Он сказал, что его интересовала философия сознания, благодаря которой он начал ходить на курс под названием "Компьютеры, Сознание и Мозг", в том же Университете Вирджинии, на котором преподавал Пол Хамфри. МакГроу думал, что идеи, которые преподавал профессор, были неверными и из-за этого он начал больше задумываться о философии сознания и искусственном интеллекте. В итоге, во время занятий, он начал использовать идеи светила индустрии, обладателя Пулитцеровской премии, Доктора Дугласа Хофштадтера, и это изменило весь его карьерный путь. Он не ходил на уроки информатики, пока не закончил школу, но любил программировать будучи подростком, с 1981-го. В университете Индианы, будучи студентом Хофштадтера, он получил две докторские степени по когнитивной науке и информатике. В итоге, он даже написал 10-ю главу в самой первой книге, из всех, продаваемых на сайте Amazon. Это была книга Хофштадтера *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought* (https://www.amazon.com/Fluid-Concepts-Creative-Analogies-Fundamental/dp/0465024750).

После колледжа он присоединился к компании из семи человек, которая, впоследствии, стала называться Cigital (https://www.cigital.com/). Компания Cigital выиграла большой грант от DARPA (Управление перспективных исследовательских проектов Министерства обороны США) на исследования в информационной безопасности, и МакГроу наняли для этой работы. В 2016-ом, когда Cigital продали Synopsys, в ней было уже 500 сотрудников. Сейчас в подразделении информационной безопасности более крупной компании

работает 1000 человек, их задача - значительно улучшать программное обеспечение.

В первый раз я обратил внимание на МакГроу, когда он и Эд Фелтен начали рассматривать безопасность языка программирования Java. Они написали книгу, и нашли десятки уязвимостей безопасности. В то время это немного шокировало, потому что Sun Microsystems изначально создавали Java, как очень безопасный язык программирования, потому что знали, что он будет работать в интернете и станет целью для постоянных хакерских атак. Язык Java появился в 1995-ом, и Sun с самого начала заявляли, что это очень безопасный язык программирования. Его создавали такие кудесники программирования, как Гай Стил и Билл Джой. Большинство экспертов в информационной безопасности задавались вопросом, действительно ли этот язык был так безопасен, как обещали Sun, или это очередное громкое обещание. Оказалось, что это было просто обещание. После обещаний, Java выпустили настолько забагованное программное обеспечение, которого еще не видел мир, МакГроу был одним из лучших по нахождению багов Java. МакГроу и Фелтен с самого начала анализировали Java на баги безопасности. МакГроу встретился со своим будущим соавтором Эдом Фелтеном на конференции, и результатом этой встречи стала первая из многих книг (https://www.amazon.com/Gary-McGraw/e/B000APFZ2S/ref=sr_tc_2_0?qid=1484584085&sr=1-2-ent). Многие книги МакГроу становились бестселлерами на сайте Amazon (одна из них, *Exploiting Software*, заняла 16-ое место среди всех книг). Это огромное достижение для книг по компьютерной тематике, и тем более для книг по информационной безопасности.

МакГроу продолжал думать о безопасности ПО, и где можно обучиться созданию более безопасного ПО. Ему было интересно, как остальные, "обычные" программисты смогут создать безопасное ПО, если даже у лучших кудесников этого ремесла (таких, как Билл Джой и Гай Стил) не получилось. Ему было интересно, что именно было не так в процессе, и почему так происходит. Он понял, что все ПО и программы должны быть разработаны и написаны с изначальным пониманием безопасности. Примерно в то же время ему пришла в голову концепция "Trinity of Trouble" (Троица проблем), в которой он объяснял, почему безопасность ПО остается интересной и сложной задачей. По сути, если речь идет о сетевом, сложном и рассчитанном на длительный срок работы ПО, то оно всегда будет интересно с точки зрения безопасности, но при этом, реализовать ее будет сложно. У Java, к сожалению, были сильно выражены все три эти черты, но самой ярко выраженной была сложность.

После того, МакГроу в 1999-ом в соавторстве написал книгу *Building Secure Software*, он побывал во многих компаниях, включая Microsoft, где Майкл Ховард, о котором пойдет речь в Главе 7, работал с Джейсоном Гармсом в только что сформированном подразделении Secure Windows Initiative. Он помнит всех

менеджеров продуктов Microsoft, которые были на его выступлениях, и что в Microsoft действительно готовы делать безопасное ПО.

Спустя еще несколько лет практики обеспечения безопасности ПО в полевых условиях (включая, как разработку сервисов, так и разработку технологий), МакГроу стал одним из создателей Building Security in Maturity Model (BSIMM). Сейчас BSIMM используется более, чем в 100 крупных фирмах для оценки, отслеживания и понимания своего прогресса в области безопасности ПО.

Я спросил, чем отличается SDL Майкла Ховарда и BSIMM, так как обе эти модели созданы для производства более безопасного ПО. Он сказал: "SDL - это практические методы. BSIMM - это оценочный инструмент, который можно использовать для оценки, сравнения и сопоставления других практических методов, подобных SDL. SDL - не единственная методология, хотя она и очень хорошо работает, и Microsoft написали и поделились ей. Майкл Ховард, которого я очень люблю, создал системный подход для огромной организации с десятками тысяч программистов. Он показал, что безопасное ПО может быть и масштабным, что очень важно".

Как и всех людей, представленных в этой книге, я спросил МакГроу, что по его мнению является самой большой проблемой информационной безопасности. В своем ответе он повторил Майкла Ховарда, у которого я ранее брал интервью. Он сказал: "Несмотря на то, что существуют тонны замечательных ресурсов по созданию и разработке более безопасных систем, в мире существует недостаточно программистов, которые хорошо разбираются в теме безопасности. Несмотря на то, что некоторые колледжи и центры обучения готовят хороших специалистов, большинство делают это ужасно, если они вообще что-то делают".

Он уверен, что предмет информационной безопасности все еще плохо проработан. Его любимая книга по безопасности и правильной разработке ПО - это книга Росса Андерсона *Security Engineering* (https://www.amazon.com/Security-Engineering-Building-Dependable-Distributed/dp/0471389226/) Он сказал, что эта книга ему нравится больше, чем 12 своих собственных: "Я думаю, это лучшая книга по безопасности на планете".

У МакГроу есть ежемесячный подкаст Silver Bullet Security Podcast (https://www.garymcgraw.com/technology/silver-bullet-podcast/), в котором он берет интервью у светил индустрии и экспертов. Он только что отметил свое десятилетие в эфире 120-ым подкастом. Просматривая списки людей, у которых он берет интервью, я обнаружил, что о многих из них мы поговорим в этой книге. Он действительно любит историю информационной безопасности, так же, как я, и хочет учиться и делиться знаниями. Когда наше интервью закончилось, я представил, как МакГроу возвращается домой и, прогуливаясь со своей собакой вдоль реки, размышляет над новыми способами улучшения информационной безопасности. Он – человек эпохи возрождения.

Подробнее о Гэри МакГроу

Подробнее о Гэри МакГроу вы можете найти на этих ресурсах:

- Книги Гэри МакГроу: https://www.amazon.com/Gary-McGraw/e/B000APFZ2S/
- Веб-сайт Гэри МакГроу: https://www.garymcgraw.com/
- Подкаст Гэри МакГроу Silver Bullet Security Podcast: https://www.garymcgraw.com/technology/silver-bullet-podcast/

Глава 9. Вредоносное ПО

Когда я только начал заниматься информационной безопасностью в 1987-ом, мое внимание привлекли вредоносные программы. Первые компьютерные вирусы (например, Apple's Elk Cloner и Pakistani Brain) только появлялись, хотя трояны и черви были уже давно. Компьютерные вирусы были настолько неизвестны, и так редко встречались, что эксперты из популярных СМИ считали их вымыслом. Так было до тех пор, пока вирусы не стали атаковать целые компании, и это было во время, когда интернет еще не был тем самым интернетом. В то время вредоносное ПО распространялось через dial-up и из рук в руки, так как люди копировали друг у друга программы (как законно, так и незаконно). Вредоносное ПО все еще остается одним из самых популярных методов взлома.

ПРИМЕЧАНИЕ Первое вредоносное ПО, которое меня "задело" называлось ansi bomb. На компьютере жертвы должен был быть драйвер ansi.sys (загруженный через config.sys), в те дни это была популярная конфигурация для IIBM-PC-совместимых компьютеров и дисковых операционных систем (DOS).

Виды Вредоносного ПО

Традиционные виды вредоносного ПО - это вирусы, черви и троянские программы. Компьютерный вирус - это самокопирующаяся программа, которая при запуске, ищет другие программы (или, если это макровирус - данные), чтобы их заразить. Компьютерный червь - это самокопирующаяся программа, которая, как правило, не изменяет данные или другие программы. У нее есть свой набор инструкций, и она просто переходит от устройства к устройству и от сети к сети, взламывая программные уязвимости. Троянские программы маскируются под реальные программы, и обманом вынуждают устройство или пользователя запустить их. Сегодняшнее вредоносное ПО - это комбинация двух и более видов. Например, сначала оно может распространяться, как троян, с помощью него получают первоначальный доступ, а затем он начинает использовать собственный код для самокопирования и дальнейшего распространения.

Вредоносное ПО достаточно эффективно. Тысячи различных вредоносных программ в течение нескольких часов заразили целые сети по всему миру. Сотни вредоносных программ за один день заразили значительную часть компьютеров, подключенных к интернету. Рекорд по скорости все еще принадлежит червю из 2003-го SQL Slammer (https://en.wikipedia.org/wiki/SQL_Slammer), который

заразил большинство подключенных к интернету, уязвимых SQL-серверов, примерно, за 10 минут. Соответствующий патч вышел через пять месяцев, но тогда никто не делал патчи вовремя. Сегодня большинство вредоносных программ - это трояны, которым нужно, чтобы конечный пользователь совершил действие (например, открыл ссылку или прикрепленный файл), чтобы начать наносить вред, хотя пользователь или устройство может и вовсе не преднамеренно (случайно) запустить эту программу. Это зависит от способа распространения и сценария работы самого вредоносного ПО.

Количество Вредоносных Программ

Сегодня на планете существуют буквально сотни миллионов различных вредоносных программ, и неизмеримое количество новых появляется каждый год. Большинство вредоносных программ - слегка измененные варианты тысяч различных вредоносных программ. Тем не менее, все варианты должны быть обнаружены защитными программами, которые часто используют сочетание цифровых подписей (уникальный набор байт для каждой вредоносной программы или семейства программ) и обнаружение по поведению. Защитная программа должна уметь быстро сканировать десятки миллионов файлов, распознавать сотни миллионов вредоносных программ, и делать это без значительного снижения производительности устройства, на котором она установлена. Это очень сложно реализовать, и даже, если все сделано с максимальной точностью, ее может обойти новая вредоносная программа, в которой изменен всего один байт.

ПРИМЕЧАНИЕ Защитные программы чаще всего называют антивирусами, хотя они обнаруживают и удаляют различные виды вредоносного ПО. Это произошло из-за того, что эти программы обрели популярность, когда большую часть вредоносного ПО составляли вирусы.

Большая Часть Создана в Преступных Целях

В наши дни одной из самых масштабных и вызывающих беспокойство тенденций является то, что большая часть вредоносного ПО создается исключительно в преступных целях. Примерно, до 2005-го, основную часть вредоносных программ писали тинейджеры и молодые люди, просто, чтобы доказать, что они могут написать вредоносное ПО. Того, что оно работало и самокопировалось было достаточно. Конечно, было несколько программ, непосредственной целью которых было нанесение вреда, но, в целом, они больше надоедали, чем представляли опасность.

Сейчас, практически все вредоносное ПО создается непосредственно с преступными намерениями. Большинство пользователей вредоносного ПО так или иначе хотят украсть деньги, будь то непосредственно воровство финансовых средств, кража идентификационных данных или паролей. В наши дни очень популярны "программы-вымогатели", которые шифруют ваши данные и просят деньги за расшифровку. Другие программы крадут игровые ресурсы или электронную валюту, или совершают несанкционированную торговлю на биржах. Рекламное ПО проникает на ваш компьютер и заставляет вас смотреть рекламу (иногда конкретно чью-то), которую, по-другому, вы бы не увидели, или скрытно заставляют ваш компьютер заходить на определенные сайты для увеличения количества посетителей, чтобы преступным путем увеличить доход от рекламы. Некоторое вредоносное ПО используется для взлома и кражи конфиденциальной информации. Также оно может использоваться для совершения распределенных атак типа "отказ в обслуживании" ("DDoS-атаки" описаны в Главе 28). Прошли те дни, когда вредоносные программы делали, в основном, подростки-хулиганы, которые писали милые надписи на вашем экране, включали из колонок Yankee Doodle Dandy, или просили вас помочь "легализовать марихуану" (например, вирус Stoned boot). Сегодня вредоносное ПО делают профессионалы!

Зачастую вредоносное ПО создается одним человеком, а продается и покупается другими. Во многих случаях, тысячи компьютеров, взломанных определенной вредоносной программой, собирают вместе, и получаются "ботнеты". Такие ботнеты можно арендовать или купить, а потом дать им команду атаковать определенный сайт или совершить какое-то действие сразу в нескольких регионах. Часто вредоносное ПО, которое первым взламывает компьютер, используется как "загрузчик". Оно получает изначальный доступ и изменяет систему, чтобы в будущем гарантировать появление другой вредоносной программы или предоставление хакеру полного контроля. Затем оно скачивает другую вредоносную программу с новыми инструкциями. Этот процесс может повторяться десятки раз, пока не скачаются и выполнятся конечная программа и инструкции. Таким образом, большая часть вредоносного ПО постоянно обновляется и остается невидимым для защитных программ. Вредоносные программы даже продаются с круглосуточной поддержкой и гарантией от обнаружения, а у их разработчиков есть свой рейтинг покупателей.

Каждый год результатом работы таких программ становятся кражи или нанесение ущерба в сотни миллионов долларов. Любой, кто давно работает в информационной безопасности и борется с вредоносными программами более десяти лет, мечтает, чтобы единственной проблемой были подростки-хулиганы.

Защита от Вредоносного ПО

Существует множество способов защиты от взлома вредоносной программой, многие из которых также хорошо работают против некоторых хакерских атак.

Полностью Пропатченное ПО

Вредоносной программе намного сложнее взломать ту систему, на которой установлены все патчи, чем ту, на которой их нет. В наши дни "наборы эксплойтов" хранятся на взломанных веб-сайтах, и когда пользователь их посещает, такой набор эксплойтов сначала ищет непропатченные уязвимости, и если их нет, пытается обманом заставить пользователя запустить троянскую программу. Если на системе не установлены патчи, то, часто вредоносная программа может скрытно запуститься, а пользователь даже не будет ни о чем подозревать.

Подготовка Пользователей

Вредоносному ПО сложно взломать систему, на которой установлены все патчи, без вовлечения конечного пользователя. В случаях, когда вредоносная программа или набор эксплойтов не смогли найти уязвимости, как правило, используется социальная инженерия. Обычно отправляется сообщение конечному пользователю, что он должен что-то запустить или открыть. Подготовка пользователей к встрече с основными техниками социальной инженерии - это отличный способ снизить результативность вредоносного ПО.

Защитные программы

Защитное ПО (часто называемое антивирусами) необходимо практически для каждой компьютерной системы. Даже лучшие защитные средства могут пропустить вредоносную программу, и ни одно из них не может быть на 100% эффективным, но использование компьютера без них - это как вождение с протекающей тормозной жидкостью. Вы можете недалеко уехать, но, в конечном итоге, произойдет трагедия. В то же время, нельзя верить производителям антивируса, которые гарантируют 100% обнаружения. Это всегда ложь.

Программы контроля приложений

Программы контроля приложений (также известные, как программы "белого списка" и "черного списка") хорошо справляются с задачей остановки вредоносного ПО, когда используются в режиме белого списка. В режиме белого списка могут работать только разрешенные и прошедшие проверку программы. Это останавливает большинство вирусов, червей и троянов. Программы контроля приложений может быть сложно использовать из-за их природы, так как каждая программа и исполняемый файл должны получить разрешение на запуск. При этом не все виды вредоносных программ или хакерских действий могут быть остановлены, особенно те виды, которые используют встроенные, реальные программы или средства создания сценариев. Однако, программы контроля приложений - это эффективные инструменты, и они постоянно улучшаются. Лично я считаю, что на каждой системе, которая считается "очень безопасной" должна быть установлена и запущена программа с белым списком приложений.

Ограничение Зоны Безопасности

Файрволы и другие локальные и сетевые виды ограничения зоны безопасности (такие, как VLAN, роутеры, и так далее) действительно способны не давать вредоносному ПО даже возможности взломать компьютер. Большинство операционных систем имеют встроенные файрволы, но, как правило, они не настроены или не включены по умолчанию. Применение файрвола может значительно снизить риск получения вреда, особенно, если присутствует непропатченная уязвимость. Более подробно файрволы описаны в Главе 17.

Система Обнаружения Вторжений

Программы или устройства сетевого обнаружения/предотвращения вторжений (NID/P) и хостового обнаружения/предотвращения вторжений (HID/P) используются для обнаружения и остановки вредоносного ПО в сети или на локальном хосте. Обнаружение вторжений описано в Главе 14. Но, как и традиционные защитные программы, NID и HID не на 100% надежны, и нельзя полагаться только на них, как на единственные средства обнаружения и предотвращения работы вредоносного ПО.

Вредоносные программы долгое время были частью угрозы компьютерной безопасности, и продолжат оставаться одной из главных угроз. В конце 1990-ых,

когда антивирусные сканеры стали работать лучшие, я был уверен, что к 2010-му вредоносные программы вымрут и останутся в прошлом. Тогда существовали лишь сотни вредоносных программ. Сейчас, когда существуют сотни миллионов различных вариантов этих программ, я понимаю, насколько оптимистично (и наивно) я мыслил.

В главах 10 и 11 речь пойдет про Сьюзан Брэдли и Марк Руссинович, которые на протяжении десятилетий успешно сражаются с вредоносными программами.

Глава 10. Профиль: Сьюзан Брэдли

Я встретил Сьюзан больше 15-ти лет назад, когда меня выбрали одним из первых Самых Ценных Профессионалов Microsoft (Most Valuable Professionals). Как известно, Microsoft MVP присуждается независимым лидерам сообщества, которые продемонстрировали глубокие познания в одной или нескольких сферах информационных технологий Microsoft, и активно взаимодействуют с конечными пользователями, например, постоянно ведут блог, делают новостные рассылки или пишут колонки. Сразу было понятно, что Брэдли - MVP из MVP. Она очень умная, трудолюбивая, и всегда помогает не только конечным пользователям, но и другим MVP (Мы тоже конечные пользователи). Первые раз она получила MVP в 2000-ом за ныне прекращенный проект Microsoft Small Business Server (SBS), но у нее огромный опыт в технической части Windows. Она каждый раз получала ежегодную премию MVP (https://mvp.microsoft.com/en-us/PublicProfile/7500?fullName=Susan%20Elise%20Bradley) и продолжила ее получать, когда ее стали называть Cloud & Datacenter Management MVP.

Если вы не знаете, чем был Small Business Server, возьмите самые крупные и сложные продукты Microsoft (включая Active Directory, Exchange, SQL, Outlook, и так далее), объедините их в одно ПО для маленького бизнеса, и скажите, что с ним легко работать. Я заработал много денег, консультируя клиентов, которые быстро поняли, что это, оказывается, не так легко. Брэдли оказала мне техническую поддержку, когда я застрял на проблеме, которую не смог бы решить сам. В конце концов, мы встречались на нескольких национальных конференциях по информационной безопасности, на которых мы оба выступали, и породнились из-за нашего общего бухгалтерского прошлого. Мы оба CPA (Certified Public Accountant - Сертифицированный аудитор), хотя она партнер в CPA-фирме, в то время как только могу называть себя “CPA”. Она написала очень важные главы для некоторых книг, у нее есть сертификат SANS Global Information Assurance Certification (GIAC), и она также является автором новостной рассылки *Windows Secrets* (http://windowssecrets.com/).

Я спросил Брэдли, как она попала в информационную безопасность, и она сказала: “Я начинала в индустрии, для которой, по определению, важны деньги, личная информация и конфиденциальность - финансовый учет. Эта сфера, благодаря которой транзакциям, на которые мы полагаемся, действительно можно доверять, тесно связана с информационной безопасностью. Мы должны быть уверены, что то, что мы вводим на клавиатуре (или в наши дни диктуем голосом и так далее) не изменится, когда дойдет до нужного репозитория. Я начала общаться с небольшими бизнес-проектами и не только из-за собственных

нужд, но и по поводу установки патчей. У меня был серверный продукт, в котором была целый микс из продуктов, и на все эти продукты нужно было устанавливать патчи. И тогда не существовало простого способа их установки. Тогда никто даже не выпускал патчи для своих продуктов. Потом появился червь SQL Slammer (в 2003-м) и он повлиял на весь мир. Против него долго не делали патчей, примерно полгода. Выпускать патчи было нелегко. Поэтому я научилась делать патчи для своего микса из продуктов, и узнала, что другие владельцы бизнеса оценили мои знания и мои навыки. С тех пор я делаю то, что делаю".

Брэдли продолжает работать с малыми бизнес-проектами, и из ее постов и общения с ней, я знаю, что она также активно помогает клиентам восстановиться после воздействия программ-вымогателей или избежать его. Я спросил ее, что она порекомендует клиентам, которые хотят восстановиться после встречи с такими программами или как не заразить свои компьютеры. Она ответила: "За несколько лет стало совершенно очевидно, что программы-вымогатели - это большая проблема не только для потребителей, но и для малого бизнеса. Найти правильную информацию сложно и для этого надо приложить много усилий, поэтому мы с моей подругой, Эми Бабинчак, которая тоже MVP, (с 2006-го), объединили усилия три года назад, и решили разобраться с программами-вымогателями. Мы создали набор инструментов Ransomware Prevention Kit (http://www.thirdtier.net/ransomware-prevention-kit/). В нем есть все, что вам нужно знать. Это хороший набор информации и инструментов, таких, как настройки и сценарии групповой политики, а теперь мы также добавляем видео, чтобы помочь людям. Это не бесплатно. Цена начинается от $25.00. Изначально все средства шли в фонд поощрительных стипендий для женщин (http://www.thirdtier.net/women-in-itscholarship-program/). Тетя Эми дала ей взаймы денег, чтобы та могла получить первый IT-сертификат, и Эми думает, что она бы не была сейчас такой успешной без так необходимых в то время денег. Она хочет вернуть долг. Фонд поощрительных стипендий покрывает расходы женщин на IT-экзамены, если они успешно их сдадут. Первоначально Эми хотела набрать в фонд $10 000, и набрали их за девять месяцев. Сегодня на стипендии уходит значительная часть фонда, но не 100%. Нужно огромное количество времени, чтобы постоянно обновлять такой набор инструментов, но Эми делает все, чтобы он постоянно обновлялся и у каждого покупателя была самая последняя версия, это огромная работа".

Я спросил Брэдли, что, по ее мнению, является самой большой проблемой информационной безопасности. Она ответила: "Мы наступаем на те же грабли и не исправляем причины. Взять, например, современное безразличное отношение к утечке данных. Из-за того, что эти утечки не сильно влияют на бизнес, это допустимый риск, который оправдан в соответствии стандартам PCI (Описанным в Главе 37, "Конфиденциальность и Стратегия"), люди просто сверяют угрозы по чеклисту, это допустимый риск, но мы не отступаем, и думаем, как улучшить безопасность потока данных. Часть проблем заключается в том, что технологии

постоянно меняются. Но не меняются основные проблемы. Вчера (в прошлом веке) у нас были большие ЭВМ. Затем появились ПК, серверы и сети - распределенная модель ПК. В то время специалисты и консультанты просто делали серверы и не волновались о том, как их обезопасить. Сейчас мы переходим к облачной модели. У всех должно быть облако! И я вижу, как совершаются те же основные ошибки. Люди перемещают серверы в облачные хранилища или используют для бизнеса облачные сервисы, но никто не понимает, как их защитить. Мы снова совершаем те же ошибки, только теперь все становится сложнее, потому что клиент не всегда контролирует безопасность, и следы преступлений уходят все дальше. Нам нужно сосредоточиться на основных проблемах, потому что технологии всегда меняются".

Если вы хотите в совершенстве освоить Microsoft Windows, то, вам обязательно нужно прочитать все, что пишет Сьюзан Брэдли.

Подробнее о Сьюзан Брэдли

Подробнее о Сьюзан Брэдли вы можете найти на этих ресурсах:

- Microsoft MVP блог Сьюзан Брэдли: http://blogs.msmvps.com/bradley/
- Сьюзан Брэдли на Windows Secrets: http://windowssecrets.com/author/susan-bradley/

Глава 11. Профиль: Марк Руссинович

Никто полностью не знает Microsoft Windows. Там написаны десятки миллионов строк кода. Но за более, чем два десятилетия, у Марка Руссиновича получилось близко подойти к этим знаниям. Он технический директор Microsoft Azure. Первые лица компании (такие, как CEO, CIO, и так далее), как правило, редко глубоко разбираются в технических деталях, но Руссинович один из тех, кто разбирается. Рядом с ним редко можно встретить людей, которые умнее, чем он или больше знают о каких-то особенностях. Он очень счастлив, когда смотрит на код. Я сказал ему об этому во время нашего интервью, и он ответил: "Тонкости технологий – это то, что мной движет!".

Я знаю Руссиновича почти два десятилетия. Долгое время у него было две компании, выпускающие ПО, Winternals - коммерческая компания и Sysinternals - компания, выпускающая бесплатное ПО. Обе компании и их ПО были очень популярны среди технарей. В конце концов, когда он пошел работать в Microsoft, они купили обе компании. Зайдите на http://www.sysinternals.com, там есть классные утилиты, которые он создал, и которые Microsoft все еще поддерживают и обновляют. Руссинович всегда был технарем из технарей, и он не боится разногласий, когда узнает правду во время своих технических расследований.

Я даже отчетливо помню, как обедал с ним в ресторане в 2005-ом (в то время никто из нас не работал в Microsoft), когда скандальное открытие руткита от Sony BMG было во всех новостях. Руссинович обнаружил, что, когда пользователи вставляли музыкальный диск от Sony в компьютер под управлением Windows, то с этого диска незаметно устанавливались две части программного обеспечения Digital Rights Management (DRM). Это ПО было нелегко удалить и, частично, оно все равно устанавливалось, даже, если человек не принимал лицензионное соглашение конечного пользователя (EULA). Оно мешало внутренним операциям Microsoft с компакт-дисками, и что еще хуже, в нем были уязвимости, которыми, в конечном итоге, воспользовались вредоносные программы.

Руссинович тестировал свою программу обнаружения руткитов, Rootkitrevealer, когда наткнулся на программу от Sony. Руткит изменяет процессы внутри операционный системы для того, чтобы скрытно работать. Он сказал, что DRM-программа от Sony – это именно вредоносный "руткит", что было очень громким заявлением в то время. Он обвинил огромную корпорацию в том, что они поступают неэтично. Его оригинальный пост об этом можно найти по ссылке: https://blogs.technet.microsoft.com/markrussinovich/2005/10/31/sony-rootkits-and-digital-rights-management-gone-too-far/.

Все СМИ рассказывали эту историю, и репутация Sony была сильно испорчена. Сначала Sony заявляли, что их программа работает нормально и не делает ничего противозаконного, но после нескольких дней общественного гнева, Sony были вынуждены признать, что это было неправильно и предложили программу для удаления. В конце концов, они отозвали соответствующие диски и предложили компенсацию. К сожалению, они плохо справились, как с ответственностью, так и с деинсталлятором. Последовавшие судебные иски сопровождались правительственными расследованиями. Подробно обо всем скандале вы можете прочитать по ссылке https://en.wikipedia.org/wiki/Sony_BMG_copy_protection_rootkit_scandal. В результате расследования Руссиновича и гнева общественности, все производители стали осторожнее, и скрытная установка ПО свелась к минимуму.

И это только небольшая часть того, что сделал Руссинович. На конференциях он постоянно проводит лекции и делает детальные технические презентации. Те из нас, кто пытается соревноваться с ним в выступлениях, оцениваемых аудиторией, знают, что второе место - это лучший результат. Он написал или был соавтором многих книг (https://www.amazon.com/Mark-E.-Russinovich/e/B001IGNICC/), включая самый продаваемый триллер о кибербезопасности. Тот факт, что некоторые из его историй о киберар-армагеддоне на самом деле могут произойти или уже происходят, пугает не меньше, чем рассказы Стивена Кинга. В 1994-ом он получил докторскую степень в компьютерной инженерии в Университете Карнеги-Меллона, и присоединился к Microsoft в 1996-ом.

Руссинович является одной из самых важных фигур в Microsoft, благодаря нему в компании произошло огромное количество масштабных технологических прорывов. Он сыграл важную роль в ускорении и обеспечении безопасности последних операционных систем Microsoft, и сейчас стоит во главе их облачных сервисов. Помимо того, что он технический директор Microsoft Azure, он Microsoft Technical Fellow, а это звание присуждается только тем людям, которые оказали значительное влияние на компанию и на мир в целом. Ирония в том, что два десятилетия назад, в 1997-ом, Microsoft чуть не уволили Руссиновича из компании, в которой он тогда работал.

Руссинович был сотрудником Open Systems Resources, и делал ПО для Windows NT 3.51. В процессе более подробного изучения внутренних компонентов Windows, он обнаружил, что изменив один ключ реестра, можно было сделать из Windows NT серверную ОС. Он уточнил: “На самом деле было два ключа реестра: один назывался ProductType, а второй был зашифрован, и использовался для обнаружения вмешательства в первый. Этот ключ реестра менял 12 других параметров системы, что, по сути, превращало Windows или в серверную ОС или в ОС для рабочей станции. И я написал статью об этом (http://windowsitpro.com/systems-management/inside-windows-nt-registry) в

журнале *Windows IT Pro*". Статью можно найти по ссылке: http://archive.oreilly.com/pub/a/oreilly/news/differences_nt.html.

Я хорошо помню эту статью. Тогда я только начал писать на том же профессиональном уровне, и, один из редакторов журнала, пригласил меня в качестве технического редактора для этой статьи. Уже тогда, перед публикацией, все понимали, что в Microsoft будут недовольны этой статьей, потому что версии Windows NT для рабочих станций и серверов позиционировались, как два совершенно разных (но схожих) продукта, и ценник на вторую версию был значительно выше.

Помню я услышал, что Руссиновича уволили из-за публикации этой статьи. Я спросил его, было ли это правдой. Он ответил: "Ну, они совершенно точно не обрадовались, но они меня не увольняли. Они надавили на Open System Resources, и из-за этого давления, я вынужден был уйти. Тогда я перешел в IBM Research, но у меня всегда были друзья в Microsoft, и со многими людьми из этой компании также были дружеские отношения. Меня все равно пригласили на презентацию о внутреннем устройстве Microsoft Windows, и я все еще писал книги о внутреннем устройстве Windows. В конце концов, меня несколько раз приглашали на работу в Microsoft. Они приобрели меня вместе с компаниями Winternals и Sysinternals, в каждой из которых в то время было по 85 сотрудников". Остальное уже история.

Сегодня Руссинович помогает разрабатывать техническое направление Microsoft Azure, добавляя в эту платформу новые возможности, делая ее быстрее и безопаснее. В последнее время он много работает с микросервисами и контейнерами. Контейнеры - парадигма виртуальных машин, популяризированная проектом Docker (https://www.docker.com/). Контейнеры появились из ниоткуда, и некоторые опасались, что они могут угрожать "большим парням" в области виртуальных машин (таким, как Amazon, Google, Microsoft и VMware). Вместо этого, под руководством Руссиновича, Microsoft переняли идею контейнеров, и теперь Azure является одним крупнейших поставщиков контейнеров в мире.

Я спросил его, помогают ли контейнеры информационной безопасности или наоборот. Он сказал: "Это зависит от того, что вы называете контейнером и от сценария сборки образа. В некоторых случаях они слегка ослабляют информационную безопасность. Контейнеры эффективны в динамичном состоянии, что усложняет атакующим возможность закрепиться, т.к. все их труды, могут быть с легкостью уничтожены. Но в то же время, если уязвимость, позволившая им проникнуть в первый раз, все еще присутствует, то они легко могут вернуться, или, если в первый раз они смогли проникнуть через статичный сервер, например SQL-сервер, то перезагрузка контейнера их не остановит. Один из недостатков контейнеров, особенно в образах проекта Docker, это наслаивание контейнеров, которое происходит при созданиии одного приложения или сервиса. И, если вам нужно установить патч или обновить код

в образе Docker, то могут существовать зависимости, которым потребуется, чтобы все связанные слои были пересобраны. Нужно проделать кратно больше работы для того, чтобы установить патч, исправить или пересобрать образ. Это один из минусов - усложнение".

В конце нашего интервью я спросил у Руссиновича, что бы он порекомендовал тем, кто рассматривает карьеру в информационной безопасности. Его ответ, по сути, полностью повторил его карьеру и успех. Он порекомендовал: "Вы должны стать экспертом во всех системах, которые собираетесь защищать, вам нужно понять, как они работают, их внутреннее устройство, включая идентификацию, принципы работы, мониторинг и сегментацию сети. Первый шаг - это глубоко ознакомиться с ПО или платформой. Во-вторых, убедитесь, что вы смотрите на интересующий вас продут с разных точек зрения. Каждая из точек зрения немного отличается друг от друга, и, если вы будете искать и понимать различные точки зрения, то вы будете лучше разбираться в том, что хотите защитить."

Подробнее о Марке Руссиновиче

Подробнее о Марке Руссиновиче вы можете найти на этих ресурсах:

- Марк Руссинович на википедии: https://en.wikipedia.org/wiki/Mark_Russinovich
- Книги Марка Руссиновича: https://www.amazon.com/Mark-E.-Russinovich/e/B001IGNICC/
- Веб-сайт Марка Руссиновича: http://www.trojanhorsethebook.com/
- Твиттер Марка Руссиновича: @markrussinovich
- Microsoft блог Марка Руссиновича: https://blogs.technet.microsoft.com/markrussinovich/
- Классные утилиты Sysinternals: https://technet.microsoft.com/ru-ru/sysinternals

Глава 12. Криптография

Огромное количество технологий, которые являются основой информационной безопасности, используют криптографию. Криптографию придумали уже давно, и она будет еще долго существовать после того, как мы покинем Землю и уйдем на другие обитаемые планеты. Лично для меня криптография - это любимый жанр информационной безопасности, и даже при том, что я почти три десятилетия увлекался шифрованием, я не считаю себя экспертом в криптографии.

Что Такое Криптография?

В цифровом мире, криптография – это использование 0 и 1 для расшифровки или проверки цифрового контента. В криптографии используются математические формулы, а также используются тех самые 0 и 1 (которые называются *криптографическими ключами*), с помощью которых удается не допускать к просмотру личных данных посторонних людей. Кроме этого криптография так же используется для подтверждения личности человека или для подтверждения достоверности информации.

Самый простой пример шифрования, который приходит мне в голову - преобразование содержимого открытого текста (не зашифрованного) в зашифрованный вид, с помощью сдвига букв вперед по алфавиту на одну. (так, например, A становится B, B становится C, C становится D, и так далее до тех пор, пока Z не станет A). Таким образом слово FROG станет GSPH. Для того чтобы увидеть исходный текст дешифровщик делает все в обратном порядке. В этом примере шифр (если его можно так назвать) - это математика, в данном случае + или - (сложение или вычитание), а 1 - это ключ. Подобные шифры использовали сотни лет для передачи секретных сообщений (и в кольцах-дешифровщиках из коробок с хлопьями), хотя они были недостаточно сложны не всегда удавалось скрыть сообщение от посторонних.

В современном цифровом мире длина ключей шифрования составляет 128 бит (128 единиц или нулей), или больше. В зависимости от шифра, длина ключа может быть больше, хотя, если алгоритм устойчив ко взлому, то, как правило, самая большая длина ключей составляет 4096 бит. Если вы видите более длинный ключ, то это, скорее всего, признак слабого алгоритма или человека, который не очень хорошо разбирается в криптографии (или пытается продать “средство от всех болезней” тем, кто не разбирается).

Почему Злоумышленник Не Может Просто Разгадать Все Возможные Ключи?

Люди, неразбирающиеся в криптографии не понимают, почему злоумышленник не может просто перепробовать все возможные комбинации из 1 и 0, в зависимости от размера ключа. Разве на самом быстром компьютере нельзя разгадать все возможные комбинации? Если коротко, то нет. Даже современный ключ в 2000 бит устойчив к "брутфорсу". Не хватит не то, что одного мощного компьютера, но даже, если взять все компьютеры в мире, не только те, которые есть сейчас, но и которые будут в обозримом будущем, их мощности все равно будет недостаточно (по крайней мере до тех пор, пока квантовая криптография не станет широко распространенной). Следовательно, все ("чистые") случаи взлома шифрования опираются на подсказки в содержимом или слабость алгоритма. Криптографический алгоритм трудно (как минимум) правильно понять, и алгоритм, который поначалу кажется идеальным, может быть полон недостатков, которые значительно ускоряют взлом. Вот почему стандарты шифрования и размеры ключей постоянно меняются, старые шифры становится легче взломать, и появляются новые.

Симметричные и Асимметричные Ключи

Если для шифрования и расшифровки применяется один и тот же ключ (как в примере выше, с 1), то такой ключ называется "симметричным". Если для шифрования и расшифровки применяются разные ключи, то это "ассиметричный" ключ. Ассиметричные шифры также известны, как системы с открытым ключом, где у одной стороны есть закрытый ключ, который известен только ей, но у всех остальных есть "открытый" ключ, и, пока никто не знает закрытый ключ, все безопасно. Однако, симметричное шифрование, как правило, быстрее и безопаснее для ключей определенной длины.

Популярность криптографии

В наши дни многие системы шифрования настолько хорошо изучены и проверены, что превратились в целую индустрию, если не в мировой стандарт.

Популярные симметричные ключи шифрования включают Data Encryption Standard (DES), 3DES (Тройной DES) и Advanced Encryption Standard (AES). Первые два примера больше не используются. Последний AES, считается очень

устойчивым и самым популярным алгоритмом симметричного шифрования на сегодняшний день. Как правило, размеры симметричных ключей составляют от 128 до 256 бит, но со временем, их длина увеличивается. Увеличение на один бит, например, со 128 до 129 бит, как правило, удваивает надежность ключа с тем же шифром.

Популярные ассиметричные шифры используют алгоритмы Диффи-Хеллмана (о Хеллмане пойдет речь в следующей главе), Ривест-Шамир-Адлеман (RSA), и Эллиптическая криптография (ECC). ECC - это новый способ, и его только начинают использовать. В основном, длина асимметричного ключа варьируется от 1024 до 4096 бит, хотя сегодня минимальной длиной ключа, использующего алгоритмы Диффи-Хеллмана и RSA считается 2048 бит. ECC использует меньшие размеры ключей, начиная от 256 бит. Сегодня достаточно устойчивым считается ключ, длиной 384 бита. В целом, асимметричные шифры используются, чтобы безопасно передавать симметричные ключи, которые составляют основную часть шифрования, между источником и получателем.

Криптографические Хэш-функции

Криптография довольно часто используется для проверки подлинности и для проверки контента. В обоих случаях используется алгоритм шифрования, известный, как криптографические хэш-функции. С таким подходом незашифрованный контент, который нужно проверить, применяется к ключу (который, повторюсь, состоит из 1 и 0) с помощью формулы, и в результате получается уникальный результат (хеш-сумма или хеш). Подлинность или содержимое могут быть хешированы в любой момент времени, а затем опять перехешированы, и в итоге сравниваются две хеш-суммы. Если хэш не изменился, то контент так же остался без изменений.

Наиболее распространенные хеш-функции - это Secure Hash Algorithm-1 (SHA-1), SHA-2 и SHA-3. Впоследствии у SHA-1 нашли криптографические уязвимости (которые также есть у SHA-2), так что от SHA-1 отказались. В данный момент алгоритм SHA-2 становится самым популярным, но некоторые эксперты по шифрованию уже рекомендуют использовать SHA-3.

Для обеспечения наилучшей защиты, большинство способов шифрования используют симметричные и асимметричные ключи, а также алгоритмы хеширования. Во многих странах, например, в США, есть определенные стандарты, которые анализируют и разрешают правительству использовать различные системы шифрования. Во многих случаях, официально разрешенные системы, также начинают использоваться и в других странах. В США, Национальный институт стандартов и технологий (http://www.nist.gov) вместе с Агентством национальной безопасности (http://www.nsa.gov) проводят открытые

соревнования, в которых кодировщики со всего мира представляют для отбора свои собственные системы шифрования. Все проходит достаточно открыто, и часто выбирают даже проигравших. К сожалению, АНБ и НИСТ, как минимум, дважды обвиняли в том, что они намеренно принимают более слабые стандарты (в частности за DES и Dual_EC_DRBG, в последнем была лазейка). В результате создалось напряжение, и многие больше не доверяют заявлениям НИСТ и АНБ о надежных способах шифрования.

Использование Криптографии

Криптография является основой современного цифрового мира. Криптография защищает наши пароли и биометрические данные, а также используется в цифровых сертификатах. Криптография применяется каждый раз, когда мы вводим учетные данные на компьютере и заходим на сайт, защищенный протоколом HTTPS. Она используется при проверке скачанного ПО, для защиты электронной почты, и при проверке подлинности компьютеров. Шифрование применяется для защиты жестких дисков и портативных носителей от несанкционированного просмотра, для защиты загрузочного сектора ОС от повреждения, а также для защиты беспроводных сетей. Оно используется для подписи программ, сценариев и документов. Благодаря ему, мы можем создавать в интернете частные подключения к интересующим нас компьютерам и компаниям, кроме того, оно используется почти во всех кредитных картах и финансовых транзакциях. Хорошее шифрование - враг шпионов, тиранов и авторитарных режимов. Без преувеличения можно сказать, что без криптографии у нас не было бы того интернета, который есть сейчас, а пользователи не могли бы управлять своими компьютерами.

Атаки На Шифрование

Существует огромное количество различных атак на шифрование. В следующие разделах будут описаны самые заметные.

Математические Атаки

Многие атаки просто стараются найти «математические уязвимости». Как правило, шифр без математической слабости может выдержать брутфорс атаку равную количеству бит в ключе минус один. Таким образом, 128-битный шифр (2^{128}), такой как SHA-1, в среднем, должен выдержать 2^{127} попыток разгадать

его перед тем, как он будет взломан. Сейчас, используя метод перебора, атакующие нашли недостатки в SHA-1, существенно ослабив этот алгоритм, примерно, до 2^{57} бит. Несмотря на то, что 2^{127} считается абсолютно устойчивым значением (по крайней мере сейчас), 2^{57} можно взломать уже сегодня или можно будет легко взломать в ближайшем будущем, без использования огромных вычислительных мощностей.

Атака на Основе Открытого Текста/Шифротекста

Многие атаки заканчиваются успешно, потому что у атакующего есть зацепка. В зацепке, как правило есть информация о наборе бит или байт либо в шифротексте, либо в расшифрованном контенте, либо в закрытом ключе. Зацепка уменьшает возможное количество бит в ключе.

Атака По Сторонним Каналам

Атака по сторонним каналам часто используется против непредвиденных недостатков в реализации системы. С помощью нее легче определить ключ. Один из распространенных примеров - изменение звука работы или электромагнитного излучения процессора, при замене 1 на 0. Таким образом, при наличии очень чувствительного подслушивающего устройства, злоумышленник может определить нули и единицы, когда компьютер получает доступ к закрытому ключу. Другой пример: злоумышленник может определить на какие клавиши вы нажимаете, когда печатаете, просто потому, что он записывает звук нажатия клавиш.

Небезопасная Практическая Реализация Криптосистемы

В современном мире, подавляющее большинство успешных взломов шифрования не является результатом атак против шифра или ключа. Вместо этого, атакующие находят недостатки практической реализации криптосистемы, которые можно сравнить с ключом от двери, положенным под ковер. Даже стойкие алгоритмы не спасут от слабой практической реализации.

Существует множество других криптографических атак, тем не менее, самые основные перечислены выше. Единственная защита от криптографических атак

- это надежный алгоритм, безопасная реализация системы, и невидимый или дружественный пользовательский интерфейс. Все остальное не имеет значения.

В Главе 3 мы говорили о Брюсе Шнайере, который считается отцом современной компьютерной криптографии. В Главе 13 речь пойдет об одном из самых известных мировых экспертов в области шифрования - Мартине Хеллмане, а в 15-й Главе речь пойдет о Докторе Дороти Э. Деннинг, которая написала одну из первых книг по компьютерной криптографии.

Глава 13. Профиль: Мартин Хеллман

Я узнал довольно интересную вещь общаясь с экспертами из определенных сфер, они, как правило, отлично разбираются не только в ней. В основном, у них есть хобби, которыми они активно увлекаются, и они пытаются "взломать" разные проблемы, многие из которых не связаны с их сферой деятельности. Мартин Хеллман, один из первых создателей открытого ключа шифрования, отличный тому пример. Будучи все еще одним из лучших мировых криптографов и думая, как решить многие современные проблемы шифрования, он также любит парить на глайдерах, улучшать отношения в браке и останавливать ядерные войны… не обязательно в таком порядке.

Хеллман известен, как один из основателей открытого ключа шифрования в 1976, вместе со своими коллегами Уитфилдом Диффи и Ральфом Мерклом. В ноябре 1976 года была опубликована работа под названием "Новые направления в криптографии" (https://ee.stanford.edu/~hellman/publications/24.pdf). Получившийся алгоритм шифрования стал известен, как *Алгоритм Диффи-Хеллмана*, но Хеллман предпочитает называть его *Алгоритм Диффи-Хеллмана-Меркла*, и называл его так во время нашего интервью. Спустя, примерно год после публикации "Новых направлений в криптографии", основанных на работе Диффи и Хеллмана, Рон Ривест, Ади Шамир и Леонард Адлеман, все из Массачусетского технологического института, создали алгоритм *RSA*, а последовавшие маркетинговые усилия их компании способствовали тому, что открытый ключ шифрования моментально завладел миром, навсегда оставив их имена в истории.

Я рассказывал о том, как Хеллман с коллегами создали открытый ключ шифрования, даже не будучи уверенным, что моя версия соответствует действительности. Это невероятная сказка о трех специалистах, ни у одного из которых не было соответствующего криптографического образования, и в которых не верил практически никто, кроме них самих. В моей версии, перед внесением поправок, я рассказывал о том, как Диффи выступал в IBM с неофициальным докладом об открытом шифровании, и никого не впечатлил. По дороге к выходу, кто-то сказал ему о еще одном "сумасшедшем парне" со схожими идеями, которого зовут Хеллман. Диффи бросил все дела, проехал на машине через всю страну и встретился с Хеллманом, который, поначалу, испугался незнакомца, проехавшего через всю страну, чтобы с ним встретиться, но быстро понял, что у них общие идеи, и они сформировали партнерство, которое повлияло на историю.

Я спросил Хеллмана, насколько правдива была моя история. Он ответил: "Когда я впервые встретил Уита, я был в восторге, а не "испуган". Вот, как это было: Я работал в IBM задолго до того, как встретил Диффи, но мне пришлось уйти в Массачусетский технологический институт, а затем в Стэнфордский университет. В 1974-ом я вернулся и выступил с докладом о проблемах современной криптографии. В то время в IBM никому это не было интересно. Хотя тогда я не знал, что они только что разработали симметричный алгоритм DES, и не могли его взломать. Руководство IBM считало, что они решили все проблемы шифрования, и пришло время двигаться дальше. В том же году, спустя несколько месяцев, в IBM пришел Уит, которого я тогда не знал, и выступал с таким же докладом, и результат был тем же, с одним исключением. Алан Конхейм, возглавлявший Отдел вычислений, сказал ему связаться со мной, когда он вернется в Сан-Франциско. К тому моменту Уит уже ездил по стране и общался с криптографами, в том числе, с Дэвидом Каном, автором популярной книги о шифровании *The Codebreakers* (https://www.amazon.com/Codebreakers-Comprehensive-History-Communication-Internet/dp/0684831309). Когда Уит вернулся в Сан-Франциско, он позвонил мне, и мы организовали встречу. Он совсем меня не напугал, и встреча, которая должна была длиться 30 - 60 минут, растянулась на часы, и я даже пригласил его с женой к себе, чтобы продолжить разговор и познакомиться с моей семьей. Тогда мы общались до 11 часов вечера. Это была осень 1974-го. До того, как я встретил Уита, все мои коллеги уговаривали меня бросить работу над шифрованием. Они говорили мне, что у АНБ огромный бюджет, и что эта организация уже несколько десятилетий работает в этом направлении. Как я могу обнаружить что-то, о чем они уже не знают? И даже, если бы у меня что-то получилось, то в АНБ сделали бы все, чтобы это скрыть. Оба аргумента были весомыми, но учитывая наши награды, было очень мудрым решением сделать то, что казалось глупостью. При том, что, несмотря на все попытки отговорить меня, я по-прежнему был настойчив, но после встречи с Уитом у меня появилась настоящая мотивация. Плюс, в течение следующих лет, мы действительно хорошо работали вместе, включая работу над открытым ключом шифрования".

Я спросил Хеллмана, кому в партнерстве принадлежали те или иные идеи. Мы знаем, что Меркл, который тогда учился в Калифорнийском университете в Беркли, самостоятельно предложил идею шифрования открытого ключа - это был обмен ключами по незащищенному каналу без предварительной подготовки. Но что именно сделали Хеллман и Диффи? Он ответил: "Я не люблю разделять затраченные на работу усилия или успех. Мы вместе работали, общались и делились знаниями. Но именно Диффи первым предложил идею криптосистемы с открытым ключом. У нас уже была идея криптосистемы с "потайным ходом", где в шифре была уязвимость (то есть, потайной ход), которую могли использовать только те, кто о нем знал. Диффи пошел дальше и начал составлять концепцию о том, как будет работать открытый ключ шифрования, и

особенно о том, как сделать криптосистему с открытым ключом, в которой можно как обмениваться ключами, так и делать цифровые подписи. Он начал думать об этом в 1975-ом. Позже мы узнали, что Меркл, независимо от нас, также работал над обменом открытыми ключами. В 1976-ом я опубликовал подход реализации идеи, которая сейчас называется алгоритм Диффи-Хеллмана, и так как этот подход был ближе к концепции Меркла, чем к нашей, я всегда настаиваю, чтобы его называли алгоритм Диффи-Хеллмана-Меркла. И прямо сейчас, сидя за тем же столом, за которым поздней ночью, в мае 1976-го, я разработал этот алгоритм, я тоже на этом настаиваю".

Я спросил, как появился алгоритм RSA. Он сказал: "Я выступал в Массачусетском технологическом институте, и мы переписывались с Роном Ривестом. Незадолго до того, как был опубликован алгоритм RSA, он отправил мне черновик этой публикации. Когда я увидел его, моей реакцией было: "Мы упустили это!". Они узнали, как использовать факторизацию больших целых чисел в качестве криптосистемы с открытым ключом. Алгоритм Диффи-Хеллмана-Меркла использовал большие целые числа, но не использовал факторизацию. Работа, которую я написал вместе со своим студентом Стивом Полигом несколькими годами ранее, включала использование алгоритма RSA, но тогда мы не думали о криптосистеме с открытым ключом, так что мы упустили этот момент".

Я спросил Хеллмана, что он чувствовал, видя, как алгоритм RSA набирает популярность и приносит миллионы своим создателям, в то время, как разработка его команды почти не приносит дохода. Он сказал: "За прошедшие годы меня много раз спрашивали, что я чувствую, зная, что алгоритм RSA появился почти сразу после нашего открытия, я и Диффи были указаны в их документе, как создатели системы с открытым ключом, но их компания (RSA Data Security) отказывалась платить отчисления. Со временем, мои чувства изменились. Сначала я думал, RSA не полностью показали связь между своим алгоритмом и моей работы со Стивом Полигом. Но, со временем, я стал смотреть на вещи по-другому. RSA так хорошо постарались, продвигая криптосистему с открытым ключом, что создали целую индустрию. Я получил признание и возможности, которых без RSA у меня могло и не быть. Сейчас мы, с Роном Ривестом, хорошие друзья. На самом деле, перед этим интервью, я даже хотел ему позвонить, и кое-что спросить".

Я поинтересовался у Хеллмана, как сильно он сейчас увлекается криптографией. А также спросил, считает ли он, что квантовая криптография в итоге получит широкое применение. Он ответил: "Ты говоришь о квантовой криптографии или квантовых вычислениях? Потому что это абсолютно разные вещи. Квантовая криптография дает возможность безопасно передавать ключи или информацию, используя квантовые свойства. Для сравнения, квантовые компьютеры могут взломать все существующие системы с открытым ключом. Я не уверен, когда они появятся, и появятся ли вообще. Это, как холодный ядерный

синтез, о котором за последние 50 лет, говорят на протяжении 20. Но когда-нибудь квантовые компьютеры могут появится. И у меня есть несколько возможных решений. Нам нужно использовать два метода шифрования и цифровых подписей, таким образом, если взломают один, другой все еще будет работать. Например, у нас есть метод шифрования с открытым ключом и центры распределения ключей (KDC, которые используются в PGP-приложениях). Нужно использовать оба метода, тогда, если квантовые вычисления взломают систему с открытым ключом, защита KDC останется. Или подписывать документ, используя традиционный открытый ключ подписи, а также подпись Меркла (https://en.wikipedia.org/wiki/Merkle_signature_scheme). Если вас действительно волнует шифрование, и вы беспокоитесь, что используемый вами способ может быть взломан в будущем, то вам нужна двойная система для подстраховки. В АНБ есть подходящее выражение: "Ремни и подтяжки". Если вы носите и то, и другое, то вас никогда не поймают со спущенными штанами". Думаю, я получил ответ на свой вопрос.

В последней части нашей беседы мы говорили о ядерном сдерживании и улучшении отношений в браке. Хеллман с женой написали отличную книгу: (https:/www.amazon.com/New-Map-Relationships-Creating-Planet/dp/0997492309/) в которой есть и то, и другое. Перед интервью он отправил мне копию книги для ознакомления, и, честно говоря, я был одержим идеей, что один из моих героев криптографии пытается отвлечь меня от темы. Потом я прочитал ее. Замечательная книга. Я купил копии для своих детей, состоящих в браке. У Хеллмана каким-то образом получается вплести свой криптографический опыт и разочарования в книгу об улучшении отношений в браке и о том как избежать ядерного уничтожения. В 2015-ом Хеллман и Диффи получили премию Тьюринга (http://amturing.acm.org/award_winners/hellman_4055781.cfm), которая, по сути, представляет из себя Нобелевскую премию в информатике. Хеллман и его жена потратят полученные $500 000 на снижение риска ядерного уничтожения - угрозу, которая стала привлекать внимание после выборов президента США в 2016-ом. Браво!

Подробнее о Мартине Хеллмане

Подробнее о Мартине Хеллмане вы можете найти на этих ресурсах:

- Книга *A New Map for Relationships: Creating True Love at Home and Peace on the Planet*: https://www.amazon.com/New-Map-Relationships-Creating-Planet/dp/0997492309/

- Биография Мартина Хеллмана в Стэнфордском университете: http://www.ee.stanford.edu/~hellman/
- Работа Мартина Хеллмана в области шифрования: https://ee.stanford.edu/~hellman/crypto.html

Глава 14: Обнаружение вторжения/APT-атаки

Обнаружение вторжения - технология обнаружения несанкционированных действий. В компьютерном мире это означает обнаружение несанкционированных подключений, использования учетных данных или доступа к ресурсам или обнаружение попыток этих действий. Обнаружение вторжений - это, отчасти, причина по которой, практически на каждом компьютере или устройстве есть журнал событий. Обнаружение вторжения и журнал событий стали нераздельными элементами с тех пор, как в 1980-ом Джеймс П. Андерсон опубликовал прорывной документ, который назывался "Контроль и наблюдение за угрозами информационной безопасности" (http://csrc.nist.gov/publications/history/ande80.pdf).

Несмотря на то, что компьютеры генерируют множество событий, люди, и их системы оценки угроз, не извлекают из этого пользы. Для большинства компьютерных пользователей лог-файлы слишком сложны и запутанны, они не в состоянии обнаружить в них нечто вредоносное.

Наиболее наглядная информация о разнице между количеством случаев проникновения угрозы и случаев ее обнаружения представлена в ежегодном отчете компании Verizon "Data Breach Investigations Report" (http://www.verizonenterprise.com/verizon-insights-lab/dbir/). В отчете 2016-го видны следующие продолжительные тенденции, вызывающие беспокойство:

- В среднем, от изначального взлома, до кражи личной информации или учетных данных, хакеру нужно от нескольких минут, до нескольких дней.
- Большинство атакующих (от 70% до 80%) долгое время (несколько месяцев) находятся в системе, перед тем, как их обнаружат.
- Обнаружение взлома внутренними ресурсами происходит, примерно в 10% случаев.

И это, несмотря на то, что, в большинстве случаев, следы проникновения есть в журнале событий, и их можно было бы обнаружить, если бы журнал событий просматривался или правильно управлялся. Если быть точнее, то я говорю о системном журнале событий и лог-файлах на устройствах безопасности (таких, как файрволы, системы обнаружения вторжений, и так далее).

Особенности Хорошего Журнала Событий

К сожалению, большинство систем безопасности генерирует тысячи, если не миллиарды сообщений в журнале, которые никак не указывают на признаки несанкционированных действий. Или, если такие признаки указаны, то записывается событие, которое представляет очень, очень слабую угрозу (например, когда файрвол регистрирует заблокированный пакет). В конечном итоге, большинство журналов событий очень "шумные", то есть, в них больше бесполезной информации, чем полезной. Исходя из этого, журнал событий, способствующий улучшению безопасности, должен иметь следующие свойства:

- Низкий уровень шума
- Малое количество записей о ложных угрозах, то есть указывать только на настоящие признаки несанкционированных действий
- Понятное описание события
- Как можно больше деталей, чтобы понять происхождение угрозы
- Создание такого события должно запускать ответное расследование

К этим свойства должны стремиться все системы обнаружения вторжений.

Развитая Устойчивая Угроза (Целевая Кибератака / APT-атаки)

Целевые кибератаки (APT-атаки) совершаются подготовленными преступными группами, и за последнее десятилетие такие атаки применялись против большинства корпораций, военных и других организаций. На самом деле, многие эксперты по безопасности уверены, что все организации, подключенные к интернету, уже стали жертвами APT-атак, или, по крайней мере, в случае необходимости, могут в любой момент ими стать. APT-атаки совершаются профессиональными хакерами, которые отличаются от обычных хакеров следующими особенностями:

- Они стараются постоянно оставаться в системе после изначального взлома.
- Если их обнаружили, они не "убегают".
- У них есть десятки сотен способов взлома и эксплойтов, которые они могут использовать, включая уязвимости нулевого дня.
- Они всегда получают абсолютный контроль над информационной инфраструктурой.

- Часто, их целью является постоянная кража интеллектуальной собственности.
- Как правило, они находятся в “безопасной гавани” - стране, где их никогда не привлекут к ответственности за их действия. (Все верно, во многих случаях их финансирует и поощряет правительство).

Причина, по которой APT-атаки описаны в этой главе, заключается в том, что их гораздо сложнее обнаружить, используя традиционные способы обнаружения вторжения. Не невозможно, а просто очень сложно без подготовки и настройки систем обнаружения вторжений. Некоторые из новейших систем, описанных в этой главе, эффективно обнаруживают и предотвращают APT-атаки.

Виды Систем Обнаружения Вторжений

Существует два основных вида систем обнаружения вторжений: основанные на сигнатурах и основанные на поведении. Многие системы обнаружения вторжений используют оба метода.

Системы, Основанные на Поведении

Также известные, как системы обнаружения аномалий, системы обнаружения вторжений, основанные на поведении, ищут аномалии, которые являются признаком активности злоумышленников. Примером такой аномалии может быть файл, который пытается скопировать себя в другой файл (компьютерный вирус), программа, которая постоянно перенаправляет пользователя на другой URL-адрес (рекламное ПО, MitM-атака, и так далее), внезапное подключение к honeypot или копирование содержимого базы данных проверки подлинности (кража учетных данных). Основная идея обнаружения по поведению заключается в том, что существует слишком много способов взлома и вредоносного ПО, невозможно уследить за всем по отдельности, лучше отслеживать их все по поведению. И это действительно работает. Например, существуют десятки миллионов компьютерных вирусов, большую часть из которых можно обнаружить по поведению, так как они все копируют себя в новые файлы. Доктор Дороти Э. Деннинг (Глава 15) является большим сторонником систем обнаружения вторжений (IDS), она также написала важный документ об обнаружении (https://users.ece.cmu.edu/~adrian/731-sp04/readings/denning-ids.pdf) аномалий в 1986-ом.

Системы, основанные на сигнатурах

В системе обнаружения вторжений, основанной на сигнатурах, используется противоположный подход. Такие системы не отслеживают вредоносное поведение, так как оно часто меняется, а используемые программы могут вызывать ложные срабатывания. Сканеры антивирусов - идеальный пример программы, основанной на сигнатурах. Они содержат миллионы уникальных последовательностей байтов (сигнатур), которые позволяют обнаружить вредоносные программы.

Инструменты и Сервисы Обнаружения Вторжений

В общем смысле, любое программное или аппаратное средство для обнаружений вредоносных действий - это программа обнаружения вторжений. К ним относятся файрволы, honeypot'ы, защитные программы и системы, ведущие журнал событий. Некоторые эксперты предпочитают использовать только те продукты, в название которых есть слова "обнаружение вторжений".

Системы Обнаружения/Предотвращения Вторжений

Системы обнаружения вторжений (IDS) созданы с целью обнаружения вредоносной активности, и, как правило, используют сочетание методов, основанных на сигнатурах и поведении. Системы предотвращения вторжений (IPS) обнаруживают и предотвращают вредоносную активность. Многие IDS также используют IPS для защиты от негативного воздействия, поэтому термин IDS может также означать IPS. Некоторые защитники не решаются использовать автоматические меры профилактики, даже, если они доступны, потому что у многих IDS/IPS часто бывают ложные срабатывания. Но иногда, при работе с IPS с меньшим риском ложного срабатывания, автоматическую защиту включают.

Далее эти системы классифицируют на хостовые IDS/IPS (host-based IDS/IPS, HIDS/HIPS) и сетевые IDS/IPS (network-based IDS/IPS, NIDS/NIPS), в зависимости от того, используется ли система на отдельном компьютере или анализирует пакеты в сети.

Первой популярной программой HIDS, которую я помню была Tripwire (https://en.wikipedia.org/wiki/Tripwire_(company)), которая появилась в 1992-ом. Ее создали студент Университета Пердью, Джин Ким, и его профессор, Юджин Спаффорд. Не случайно, что в Университете Пердью также училась и преподавала Доктор Дороти Деннинг.

Первой суперпопулярной программой NIDS, которую я помню была бесплатная Snort (https://www.snort.org/). Мне повезло, и этой программой меня научил пользоваться ее создатель, Мартин Роеш, на лекции в SAN Institute, в 1990-ом. Сейчас эта программа также остается очень популярным коммерческим продуктом, у нее есть бесплатная и платная версии, разработанные Sourcefire.

Системы Управления Событиями

За каждым успешным обнаружением проникновения или решением для журналирования стоит система, которая распознает и собирает события с помощью одного или нескольких "сенсоров". В любой организации, где много компьютеров, для извлечения максимальной пользы, необходим анализ и сбор всех событий в одно целое. Системы управления осуществляют сбор, анализ событий, и выдают оповещения безопасности. От работы таких систем зависит эффективность или неэффективность всей системы. У каждой системы управления событиями есть множество компонентов и особенностей. Специальное издание 800-92 Национального института стандартов и технологий "Руководство по управлению журналом безопасности" (http://csrc.nist.gov/publications/nistpubs/800-92/SP800-92.pdf) считается наиболее полным руководством по эффективному управлению журналом событий. Хороший менеджер управления событиями сложен в освоении и требует много ресурсов. Соответственно, существует множество коммерческих производителей, которые готовы сделать за вас всю сложную работу. Они известны, как Security Information and Event Management (Управление информационной безопасностью и событиями безопасности / SIEM) компании или сервисы.

Обнаружение Целевой Кибератаки (APT-атаки)

Профессиональные APT-хакеры умеют проникать в компании, практически не оставляя следов. Долгие годы их обнаружение считалось сложной, а порой даже невозможной, задачей. Но, в конце концов, способы обнаружения вторжений стали намного эффективнее, и сейчас существует несколько продуктов, сервисов и компаний, которые хорошо справляются с задачей обнаружения APT-атаки.

Производители операционных систем встраивают возможности и сервисы, которые значительно лучше обнаруживают такие виды киберпреступлений. Примеры таких сервисов - это Advanced Persistent Threat (https://www.microsoft.com/en-us/cloud-platform/advanced-threat-analytics) и Advanced Threat Protection (https://www.microsoft.com/en-us/WindowsForBusiness/windows-atp) от Microsoft.

Сейчас многие компании регулярно преследуют десятки различных хакерских групп, которые проводят целевые атаки. Удается определить, что именно они делают и где именно. Многие компании предлагают услуги по быстрому обнаружению целевых атак и оповещению об их присутствии. Вероятно, самая большая разница между традиционными и новыми средствами обнаружения вторжений заключается в возможности последних собирать данные о многих компаниях по всему интернету. Одними из самых известных компаний в этой области являются CrowdStrike (https://www.crowdstrike.com), AlienVault (https://www.alienvault.com) и давний игрок рынка TrendMicro (http://www.trendmicro.com).

Очевидно, что киберпреступникам становится сложнее прятаться. В следующей главе, Главе 15, речь пойдет о пионере обнаружения вторжений, Докторе Дороти Э. Деннинг. В Главе 16 речь пойдет про Майкла Дубинского, менеджера по продукции одного из самых продвинутых сервисов обнаружения вторжений, доступных на сегодняшний день.

Продолжение книги: Вторая и Третья части