Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное
учреждение высшего профессионального образования
ОМСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
им. Ф.М. ДОСТОЕВСКОГО

А.К. Гуц, Т.В. Вахний

Теория игр и защита компьютерных систем

2013

УДК 519.83+681.3.067
ББК 22.19я73
Г 977

Рекомендовано к изданию
редакционно-издательским советом ОмГУ
Рецензенты:
д-р техн. наук, проф. В.А. Филимонов
канд. физ.-мат. наук, доц. Н.Ф. Богаченко

Гуц, А.К.

Г977 Теория игр и защита компьютерных систем: учебное пособие / А.К. Гуц, Т.В. Вахний. – Омск: Изд-во ОмГУ, 2013. – 160 с.

ISBN 978-5-7779-1655-6

Книга посвящена вопросам применения теории игр в сфере защиты информации, размещенной в компьютерных системах. Излагаются элементы матричных, биматричных, позиционных и стохастических игр. Описывается игровой подход при защите от DDoS-атак, противодействии несанкционированному доступу к информации, а также решении проблемы оптимального размещения конфиденциальной информации на серверах и анализе безопасности компьютерных систем.

Для студентов и аспирантов факультетов компьютерных наук и математических факультетов, специализирующихся в области информационной безопасности.

УДК 519.83+681.3067
ББК 22.19я73

ISBN 978-5-7779-1655-6

Оглавление

Введение

Защита информационных систем – одна из важнейших задач любой службы безопасности любой организации и любого предприятия.

Информационная система – это совокупность содержащейся в базах данных информации и обеспечивающих ее обработку информационных технологий и технических средств[1].

Любая современная информационная система предполагает определенную степень ее автоматизации. Информационные системы, в которых автоматизация не является полной, т. е. требуется постоянное вмешательство персонала, называются *автоматизированными*. Поскольку автоматизация предполагает наличие и использование компьютеров (средств вычислительной техники), вместо термина «автоматизированный» употребляется термин «компьютерная».

Вследствие этого понятия «компьютерная информационная система», «автоматизированная информационная система» и просто «компьютерная система», «информационная система» считаем синонимами.

Компьютерные системы многих предприятий очень часто становятся объектами, на которые направлены помыслы злоумышленников. Посредством несанкционированного доступа злоумышленник может похитить информацию, а посредством

[1] Федеральный закон Российской Федерации от 27 июля 2006 г. № 149-ФЗ «Об информации, информационных технологиях и о защите информации».

компьютерной атаки либо уничтожить ее, либо временно ограничить доступ к ней. В любом случае предприятия несут как финансовые потери, так и моральные, как, например, банки, уровень доверия к которым падает, если они стали жертвами атаки хакеров.

Для выбора средства эффективной защиты от различного рода компьютерных атак можно использовать методы теории игр.

Теоретико-групповой подход к исследованию взаимоотношений злоумышленника и администратора предполагает, что они являются игроками в некоторой игре, где каждая сторона делает свои шаги, выбирая ту или иную стратегию[2] поведения, стремясь оптимально обеспечить свой интерес. Предпочтительнее было бы говорить о максимальном обеспечении своего интереса, но противоположная сторона мешает это сделать. Поэтому приходится говорить о поиске оптимальной стратегии поведения как по отношению к администратору, так и по отношению к злоумышленнику, ведь обе стороны, как правило, стараются свести к минимуму свои неудачи. Во всяком случае так ведет себя опытный и осторожный хакер.

Целью теории игр является выработка естественных представлений об оптимальности ситуаций и стратегий игроков, предсказание их существования в игре и указание способа их нахождения и перечисления [15].

[2] *Стратегия* (греч. strategos – войско веду) – набор явно выраженных шагов мышления и поведения, позволяющий достичь конкретного результата.

Глава 1

Проблемы защиты компьютерных систем

Компьютерные системы в XXI веке становятся самым распространенным ресурсом, на котором хранится самая различная информация. Подключение этих систем к сети Интернет приводит к тому, что хранящаяся информация становится объектом нападения самых различных злоумышленников, от отдельных хакеров, горящих желанием «пробить» защиту ресурса, до организованных преступных сообществ, а также разведовательных и военных служб различных государств.

Существуют многочисленные способы создания угрозы безопасности информационным компьютерным ресурсам. Это атаки типа DoS (Denial Of Service – отказ от обслуживания)[1] (SYN/ACK, DDoS-атаки), ICMP-атаки, использующие уязвимости в реализации протокола ICMP. Другие угрозы, такие

[1]На атакуемый сервер одновременно с множества компьютеров поступает огромное количество «ложных» запросов (бессмысленных или сформированных в неправильном формате и др.). В итоге сервер вынужден затрачивать все свои ресурсы на обработку этих запросов DDoS атаки, что приводит к невозможности обслуживания обычных пользователей или к прекращению нормального функционирования компьютерной системы.

как заражение компьютерной системы вирусами или червями, различные взломы системы могут повредить целостность данных, хранящихся на компьютерах. А угрозы, подобные программам-шпионам, троянским коням, нарушают конфиденциальность информации.

Традиционный защитный механизм включает брандмауэры, систему обнаружения вторжения (IDS) и антивирусные программы. Некоторые из этих стратегий защиты разработаны только для определенных угроз. Брандмауэры прежде всего разработаны, чтобы исследовать заголовки IP, TCP и пакеты UDP. Брандмауэры слабы, когда нападение нацелено на прикладной уровень. Антивирусные программы необходимы для защиты против вируса или червя нападения, но не очень эффективны, когда предполагают, что они способны предотвратить взлом.

Очевидно, что приобретение и установка всевозможных средств защиты информационного компьютерного ресурса, помимо того, что ведет к материальным затратам, требует наличия инструкций по эффективному их использования. Необходимо вооружить системного администратора, службу безопасности компьютерной сети оптимальной стратегией обеспечения защиты хранящейся информации.

1.1. Теория игр

Противостояние администратора и злоумышленника – это *конфликт*, т. е. противоборство двух сторон с противоположными интересами, которые каждая сторона пытается удовлетворить, используя ту или иную стратегию действий, и при которых невозможно прийти к удовлетворяющему обе стороны соглашению по поводу находящегося в распоряжении администратора информационного ресурса.

Формализация содержательного описания конфликта представляет собой математическую модель, которую называют *игрой*, а участников конфликта – *игроками*.

Таким образом, как подчеркивает Е.В. Яроцкая[2]:

> Игра – упрощенная формализованная модель реальной конфликтной ситуации.

Теория игр – это математическая теория, которая способна вырабатывать и находить оптимальные стратегии, инструкции по организации системы защиты компьютерной системы. Игровая модель конфликта применяется для оценки и прогнозирования достигнутого уровня защиты информации с учетом выбора наиболее оптимальной стратегии поведения игроков. Предполагается, что при многократном повторении игры реализация оптимальных стратегий обеспечит администратору максимально возможную защиту компьютерной системы.

Теория игр позволяет:

– представить задачу по защите компьютерной системы в математическом виде, позволяющем воспользоваться разработанными критериями нахождения оптимальных стратегий защиты, придерживаясь которых, администратор способен устранить или по крайней мере свести к минимуму наносимый злоумышленником ущерб хранящейся информации;

– наиболее точным образом оценить затраты на обеспечение безопасности информационного ресурса;

– принимать решения в условиях неопределенности;

– предсказывать поведение конфликтующих сторон и предлагать набор действий, которыми стоит воспользоваться в любой допустимой ситуации.

Вместе с тем теория игр направлена на поиск решения, оптимального или рационального в среднем. «Она исходит из принципа минимизации среднего риска. Такой подход не вполне адекватно отражает поведение сторон в реальных конфликтах, каждый из которых является уникальным» [26, с. 224].

Кроме того, в теории игр фигурируют как основные исходные понятия полные списки стратегий поведения каждого

[2] Доцент Томского политехнического университета.

игрока, которые известны всем сторонам конфликта. В реальном конфликте противоборствующие стороны часто скрывают доступные им способы достижения своих целей. Поэтому, даже если администратор компьютерной сети, как ему кажется, и составил полный список стратегий хакера, всегда остается возможность столкнуться с совершенно неожиданным поведением злоумышленника.

1.2. Риски безопасности компьютерной системы

При решении проблемы защиты компьютерных систем важно уметь оценивать рискованность имеющейся системы защиты информационного ресурса. Делается это с помощью различных способов оценки риска безопасности компьютерной системы.

1.2.1. Пример количественной оценки риска безопасности системы

При количественной оценке рисков принято выделять следующие составляющие:

– стоимость информационного ресурса R,
– угроза B,
– уязвимость U,
– средства защиты Z,
– фактор угрозы F,
– единичное ожидание потерь ЕОП,
– годовая оценка инцидента ГОИ,
– ожидаемые годовые потери ОГП.

Предположим, мы имеем компьютерную сеть с числом персональных компьютеров (ПК) = 600. В таком случае R – это стоимость ресурса «рабочая станция пользователя». При установлении суммы стоимости ресурса нет смысла включать в нее деньги, потраченные на его приобретение. Следует оценивать

как минимум предполагаемые затраты на ее сопровождение, восстановление данных, потери от простоя. Примем, что $R =$ 10 000 руб.

За B примем вирус, полученный через Интернет, приведший к временной недоступности ПК, утрате, искажению данных и т. п.

Далее, имеем:

U – это недостатки (ошибки) в Internet Explorer;

K – своевременно установленные патчи;

F – процент потерь в случае реализации угрозы на данном ресурсе. Предположим, что $F = 0{,}5$ ($= 50$ %), т. е. в случае реализации угрозы мы ожидаем потерять половину от стоимости ресурсов одной рабочей станции (потенциальные потери от простоя ПК, украденной информации и т. п.).

Иначе говоря, единичное ожидание потерь, т. е. потери при взломе одной рабочей станции

$$\text{ЕОП} = RF = 10\ 000 \times 0,5 = 5\ 000 \text{ (руб)};$$

Годовая оценка инцидента ГОИ – это число, отражающее частоту проявления угрозы в год. Положим, что число подверженных атаке рабочих станций равно 50 из 600, частота появления таких супервирусов – 2 в год. Тогда

$$\text{ГОИ} = 50 \times 2 = 100.$$

Следовательно, ожидаемые годовые потери

$$\text{ОГП} = \text{ЕОП} \times \text{ГОИ} = 5\ 000 \times 100 = 500\ 000 \text{ (руб)}.$$

Теперь можно задаться вопросом, стоит ли приобретать средства защиты системы Z? Как ни жалко расставаться с весомой денежной суммой, но если этого не сделать, то вполне реальны ежегодные потери в 500 000 руб от заражения вашей компьютерной системы парой новых супервирусов.

1.2.2. Оценка серьезности сетевой атаки

Оценка риска безопасности компьютерной системы может оцениваться через определение степени опасности атаки на си-

стему.

Атаки разной степени опасности требуют разного уровня реагирования. А.М. Астахов предложил нижеприведенную методику оценки степени опасности атаки.

Опасность атаки определяется величиной OA, означающей ее осуществление. Величина OA, во-первых, определяется вероятностью PA успешного осуществления атаки и, во-вторых, величиной возможного ущерба.

Величина возможного ущерба определяется степенью защищенности ZR компьютерной системы, против которой направлена атака. Чем меньше система защищена, тем большим может быть ущерб.

Вероятность успешного осуществления атаки PA определяется эффективностью методов и величиной уязвимости системы защиты, используемых для ее осуществления. Величина уязвимости определяется эффективностью контрмер системного SCM и сетевого уровня NCM, используемых для противодействия данному виду угроз.

Формула для определения уровня серьезности атаки выглядит следующим образом:

$$OA = ZR + PA - (SCM + NCM), \tag{1.1}$$

где

ZR – степень защищенности сетевого ресурса;

PA – вероятность успешной атаки;

SCM – эффективность реализованных контрмер системного уровня;

NCM – эффективность реализованных контрмер сетевого уровня.

Формула (1.1) может использоваться для определения величины рисков, связанных с атаками, выявленнными при помощи IDS, при анализе результатов мониторинга сетевого трафика. Обычно интерес представляют только те атаки, для которых величина риска превышает некоторое установленное значение.

Для определения уровня серьезности атаки OA используется числовая шкала от -10 до $+10$.

Атака серьезна, риск велик, если $OA > 0$ и близко к $+10$. Атака незначительна, если $OA < 0$ и близка в -10.

$OA \in [-10, 10]$ – величина риска, связанного с осуществлением сетевой атаки.

Защищенность сетевого ресурса ZR определяется по 5-балльной шкале исходя из назначения данного сетевого ресурса и выполняемых им функций. На практике обычно ориентируются на следующую шкалу:

Назначение сетевого ресурса	**Балл**
Межсетевой экран, DNS-сервер, маршрутизатор	5
почтовый шлюз	4
UNIX рабочая станция	2
Персональные компьютеры с ОС Windows XP	1

Вероятность успешного осуществления атаки PA и вид ущерба определяется по следующей шкале:

Атакующий может получить права суперпользователя на удаленной системе	5
Отказ в обслуживании в результате осуществления сетевой атаки	4
Получение прав непривилегированного пользователя на удаленной системе, например, путем перехвата пароля, передаваемого по сети в открытом виде	3
Раскрытие конфиденциальной информации в результате осуществления несанкционированного сетевого доступа, например, атака "null session"на Windows системы	2
Вероятность успешного осуществления атаки очень мала	1

Эффективность реализованных контрмер системного уровня SCM можно оценить по следующей шкале:

Современная ОС, установлены все программные коррекции (пакеты обновления), используются дополнительные (наложенные) сетевые средства защиты (например, "tcp wrappers"или "secure shell")	5
Устаревшая версия ОС, не установлены некоторые программные коррекции	3
Отсутствуют специализированные средства защиты, отсутствует политика управления паролями, пароли передаются по сети в открытом виде	1

Эффективность реализованных контрмер сетевого уровня NCM можно оценить по следующей шкале:

Межсетевой экран, реализующий принцип минимизации привилегий, является единственной точкой входа в сеть	5
Межсетевой экран и наличие дополнительных точек входа в сеть	4
Межсетевой экран, разрешающий всё, что явным образом не запрещено (разрешительная политика управления доступом)	2

Как уже было отмечено, данная методика оценки рисков, связанных с осуществлением сетевых атак, используется в SANS/GIAC при анализе подозрительных фрагментов сетевого трафика (detects), обнаруженных при помощи сетевых IDS.

1.3. Методика применения теории игр в сфере защиты компьютерных систем

Методика применения теории игр в сфере защиты компьютерных систем состоит из следующих этапов:

• Постановка теоретико-игровой задачи, которая заключается в том, что задача организации защиты компьютерной системы представляется в терминах и понятиях теории игр. Определяются игроки, их число и стратегии, платежные функции. Очевидно, что игроки – это системные администраторы и злоумышленники. Стратегии администраторов состоят из программно-аппаратных средств защиты, находящихся в их распоряжении, а стратегии злоумышленников – это способы взлома серверов и способы атак на них;

• Выбор и построение теоретико-игровой модели (типа игры) конфликта (игры). Иначе говоря, решается вопрос, о какой игре идет речь – последовательной или параллельной, бескоалиционной или коалиционной и т. д.;

• Решение игры (нахождение оптимальных стратегий),

• Анализ решения и его реализация в организации защиты компьютерной системы.

Если решение игры – чисто формальная сторона вопроса, поскольку на этом этапе используются всевозможные теоремы, наработки и программные продукты, созданные и накопленные поколениями специалистов по теории игр, то анализ решения игры и его использование при организации защиты компьютерной системы – задача, требующая достаточно серьезной работы по осмыслению того, как чисто математические формулы будут превращены в практический механизм, осуществляющий защиту информации, размещенной в компьютерной системе.

1.3.1. Как использовать найденные стратегии

Если решение игры однозначно дает конкретную, *чистую* стратегию, т. е конкретный способ защиты, то в принципе достаточно ему следовать. Но если решение выдает набор стратегий, каждой из которых соответствует вероятность, то возникает большая проблема по практическому воплощению такой рандомизированной, *смешанной* стратегии. Действительно, нельзя найденную вероятность, скажем $1/3$, понимать как рекомендацию задействовать соответствующую ей стратегию каждый третий день или час, ведь хакер может не посчитаться с таким графиком защиты сервера.

Как в таком случае использовать смешанную стратегию? Случайный выбор стратегии защиты можно осуществлять следующим образом [46, с. 39].

Пусть имеются три стратегии, для которых найдены вероятности $p_1 = 0, p_2 = 0,35, p_3 = 0,65$. Составляются два промежутка $[0, 0,35]$ и $(0,35, 1]$. Берется случайная величина со значениями на отрезке $[0, 1]$ с равномерным распределением. Если в испытании ее значение попадает в первый или второй промежуток, то для защиты сервера берется вторая или, соответственно, третья стратегия.

Такой подход не является безответственным, а отвечает

объективной закономерности рассматриваемого конфликта, в котором реально имеет место случайный характер выбора ходов игроками, что и отразилось в том, что решение задачи дало не «чистую» стратегию, т. е. одну единственную, а «смешанную», состоящую из трех возможных с набором их вероятностей. «Применение случайного механизма означает гибкую, подвижную, неожиданную для противника тактику, целесообразность которой всегда очевидна» [46, с. 40].

Однако нетрудно представить реакцию владельца компьютерной системы, которому системный администратор скажет, что при принятии решения о выборе стратегии защиты он бросает кости. И если при этом администратор начнет еще говорить что-то наукобразное об объективности применения случайного механизма, то скорее всего вопрос о его увольнении прийдет на ум владельцу гораздо раньше, чем желание задуматься о диалектике случайного и необходимого.

Поэтому на практике используются обе стратегии защиты, поскольку трудно реализовать задействование той или иной стратегии на каждом такте работы компьютерной системы в пропорции $p_2 : p_3$. В силу того, что обе стратегии появились как решение игры, в которой имеются, как правило, и многие другие стратегии защиты, помимо этих двух, то задействование сразу обеих статегий защиты не является чрезмерно избыточным.

Решение о применении найденной стратегии игрок принимает в условиях неопределенности. В лучшем случае он знает список стратегий своего противника, знает в результате найденного решения игры, какая его стратегия и какая стратегия противника оптимальны, но не знает, какую все-таки стратегию применит противник.

Поэтому рекомендуется до построения модели собрать как можно больше информации о способах взлома компьютерных систем, о финансовых возможностях и интересе злоумышленника, о его психологическом состоянии и т. д. Это позволит выбрать более адекватно отражающую ситуацию модель конфликта, т. е. выбрать тип игры, найти более точные выраже-

ния для функций выигрышей сторон, определиться с тем, что будет пониматься под оптимальностью стратегий.

В гл. 4 показано, как администратор может сочетать подсказку, даваемую теорией игр, касательно выбора оптимальной стратегии, и учитывать при этом все доступные на рынке программные и аппаратные средства защиты, опираясь при принятии решения на специальную компьютерную программу.

1.3.2. Критерии оптимальности

В книге под оптимальными понимаются стратегии, удовлетворяющие *максиминному критерию фон Неймана* либо ситуации *равновесия Нэша*.

Существуют и другие критерии оптимальности, удовлетворяя которым стратегии рассматриваются как оптимальные.

Часто используется *критерий оптимума по Парето*, согласно которому игроки должны стремиться к ситуациям, в которых их выигрышы максимальны, или, другими словами, к ситуациям, любое отклонение от которых повлечет уменьшение выигрыша хотя бы одного игрока.

Критерий доминирования в матричных играх предлагает игроку выбирать ту стратегию, которая доминирует среди остальных в так называемой платежной матрице.

Критерий Сэвиджа для принятия решения использует не только возможные выигрыши, заданные в платежной матрице, но и проигрыши, данные матрицей рисков.

Равновесие Штакельберга – это такой (набор) стратегий игроков, что первый игрок (лидер) с учетом целей партнеров адекватно прогнозирует равновесия Нэша, складывающиеся после его хода, и оптимизирует свою стратегию соответственно, а остальные поступают согласно его прогнозу (С.Г. Коковин).

Но важно не только ввести в теорию тот или иной критерий оптимальности, нужно доказать, что для рассматриваемого типа игры соответствующие оптимальные статегии существуют. Причем для практического использования данного

критерия небходимо задать алгоритм, посредством которого оптимальные стратегии будут найдены (вычислены).

1.4. Проблемы выбора критерия оптимальности

С помощью теории игр можно, например, эффективно определить, при каких условиях двум «недружественно» настроенным партнерам целесообразно сотрудничать и добиваться оптимальных для них результатов – достичь оптимальной ситуации в соотношении «выигрыш / выигрыш». Каждая сторона должна считаться с соперником и в чем-то жертвовать своими интересами в соответствии с нэшевской рациональностью.

Однако в случае защиты компьютерной системы над администратором «нависают» статьи Уголовного Кодекса РФ, и в силу этого не следует ожидать от него рационального отношению к выбору равновесной стратегии. Тем не менее при защите компьютерных сетей теория игр позволяет проверить эффективность разработанной стратегии защиты охраняемой компьютерной системы и спрогнозировать ситуации, которые могут возникнуть в процессе обеспечения компьютерной безопасности, вверенного информационного ресурса.

Это говорит о том, что в случае защиты компьютерных систем от хакеров или в случае кибервойн, ведущихся государствами, следует использовать не только ситуации равновесимя Нэша, но и другие. Так, в гл. 8 используется критерий оптимальности по Парето.

Тем не менее, как показано в гл. 14, ситуации равновесия Нэша вполне пригодны для организации отражения атак злоумышленников.

Глава 2

Элементы теории игр

Конфликт – это ситуация, в которой сталкиваются интересы двух сторон и в которых каждая сторона пытается достичь наибольшей выгоды для себя, сводя к минимуму свои потери. Человеческая практика наполнена всевозможными конфликтными ситуации, которые требуют разрешения.

Для решения этих задач естественно обратиться к всесильной математике. Математическая модель конфликтной ситуации называется *игрой*, а математическая теория, помогающая принимать рациональные решения в конфликтной ситуации, – *теорией игр*.

2.1. Игры и их классификация

Игра состоит из игроков, которые поочередно или одновременно делают *ходы*. Игра осуществляется по установленному набору *правил игры*. Правила игры описывают [23, с.14]:

– что разрешается или что требуется делать лицу, называемому игроком, при всех возможных обстоятельствах;

– правила определяют сведения, которые получает каждый игрок;

– момент окончания игры, сумму, которую уплачивает или получает каждый игрок, и цель каждого игрока;

– число ходов, число игроков и платёж (*платёжные функции*, или *функции выигрышей* игроков), который является количественной оценкой результатов игры.

Действия каждого игрока, его поведение, т. е. выбор *хода*, определяется набором правил, которые составляют *стратегию* игрока. Делая ход, игрок выбирает стратегию из множества возможных своих стратегий. Выигравший игрок получает *выигрыш*.

Выигрыш – это оценка ожидаемых результатов всех возможных сочетаний стратегий одного игрока со стратегиями другого.

Оптимальной стратегией называют такую стратегию, при которой достигается максимальный ожидаемый средний выигрыш при многократном повторении игры. При этом под «максимальным ожидаемым средним выигрышем» понимается тот выигрыш, который дает выбор стратегии на основе принятого *критерия оптимальности*.

Игры классифицируют по числу игроков – игра двух лиц, трех лиц и т. д. Различают игры с *нулевой суммой платежей* и *игры с ненулевой суммой платежей*.

Если игроки производят расчеты только между собой, то такая игра называется *игрой с нулевой суммой*.

Игра называется *конечной*, если в ней каждый игрок имеет конечное число стратегий. Прочие игры называются *бесконечными*.

Игра называется *игрой с полной информацией*, если в ней игроки знают все ходы, сделанные до текущего момента, равно как и возможные стратегии противников. Это позволяет им в некоторой степени предсказать последующее развитие игры. Полная информация не доступна в параллельных играх, так как в них не известны текущие ходы противников.

Игра называется *игрой с неполной информацией*, или *байессовой*, если перед началом игры игроки не располагают всей

информацией о стратегиях, платежных функциях или иной важной для принятия решений информаций, касающейся других игроков.

Игра называется *параллельной*, или *статичной*, если либо игроки ходят одновременно, либо по крайней мере они не осведомлены о выборе других игроков до тех пор, пока все не сделают свой ход. Параллельные игры являются *одноходовыми*: все игроки делают только один ход.

Параллельные игры имеют *нормальную*, или *стратегическую, форму*, т. е. описываются платёжной матрицей. Примером параллельной игры являются матричные игры (см. § 2.2).

Игра называется *последовательной*, или *динамической*, если игроки могут делать ходы в заранее установленном либо случайном порядке, но при этом они получают некоторую информацию о предшествующих действиях других. Не предполагается, однако, что эта информация является полной. Последовательные игры являются *многоходовыми*.

Последовательные игры имеют *экстенсивную*, или *расширенную, форму*, т. е. представляются в виде *ориентированного дерева*, где каждая вершина соответствует ситуации выбора игроком своей стратегии. Каждому игроку сопоставлен целый уровень вершин. Платежи записываются внизу дерева, под каждой листовой вершиной (см. § 9.1, 9.2). Примером последовательной игры являются позиционные игры (см. гл. 9).

Игра называется *коалиционной*, или *кооперативной*, если игроки могут объединяться в группы, беря на себя некоторые обязательства перед другими игроками и координируя свои действия. В группу могут объединится злоумышленники $P_1, ..., P_{N-1}$ с целью произведения скоординированной атаки на некоторый информационный ресурс P_N.

Игра называется *бескоалиционной*, если все ее игроки действуют независимо друг от друга, без взаимного сотрудничества или обмена информацией.

Рис. 2.1. Классификация игр

2.1.1. Чистые стратегии игроков

Что в реальности понимается под *стратегией* игрока? Как правило, каждый игрок составляет перед партией план проведения игры с начала до конца. Он должен быть полным и охватывать все возможные варианты проведения игры.

Стратегия игрока – это подробное описание того, как должен поступать игрок (какое должен выбирать решение) во всех возможных ситуациях, сложившихся в процессе игры.

Таким образом, каждый игрок P имеет в своем распоряжение конечный набор своих стратегий:

$$x_1, x_2, ..., x_n.$$

Каждая стратегия x_i – это конкретное действие игрока (ход), которое он производит в зависимости от сведений, которые, возможно, он может получить в ходе игры. Такие стратегии называют *чистыми стратегиями*. В действительности нужная информация, которая позволяет однозначно выбрать конкретную стратегию x_i, может и не поступить, и тогда приходится производить выбор стратегию с вероятностью.

Игрок, применяя какую-нибудь стратегию x_i, не теряет свободу действий, так как эта стратегия указывает его действия в зависимости от сведений, которые он получит.

2.1.2. Смешанные стратегии игроков

Под *смешанной стратегией* s игрока P будет пониматься набор неотрицательных чисел $p_1, ..., p_n$, сумма которых равна единице и которые поставлены в однозначное соответствие чистым стратегиям $x_1, x_2, ..., x_n$ этого игрока.

Как это понимать, если выражаться на более простом языке? Если стратегия игрока описывается во фразах «надо стратегию x_1 выбирать с вероятностью p_1, стратегию x_2 – с вероятностью p_2, ..., стратегию x_n выбирать с вероятностью p_n», то мы имеем дело со смешанной стратегией.

На практике каждая из компонент p_i смешанной стратегии s показывает относительную частоту использования игроком

соответствующей чистой стратегии x_i.

Смешанную стратегию будем представлять как формальную сумму

$$s = \sum_{i=1}^{n} p_i x_i,$$

где

$$\sum_{i=1}^{n} p_i = 1, \quad \forall i (p_i \geq 0).$$

2.2. Матричные игры

Изучим *конечную игру двух лиц с нулевой суммой*. Очевидно, что в такой игре выигрыши одного игрока равны проигрышам другого игрока.

Матричная игра Γ_A – это игра, где два игрока P_1 и P_2 играют в игру с нулевой суммой, имея конечное число чистых стратегий $i \in \{x_1, ..., x_n\}$ и $j \in \{y_1, ..., y_m\}$ соответственно, и для каждой пары стратегий (ij) задан платеж a_{ij} второго игрока первому. Матрица $A = (a_{ij})$ задает выигрыш первого игрока и проигрыш второго.

Матричная игра является одноходовой. Оба игрока делают свой ход, после чего происходит распределение выигрышей. Поскольку это игра с нулевой суммой, то выигрыш игрока P_1 равен a_{ij}, а выигрыш игрока P_2 равен $(-a_{ij})$.

Таблица 2.1

Платежная матрица

$P_1 \setminus P_2$	y_1	y_2	...	y_m
x_1	a_{11}	a_{12}	...	a_{1m}
x_2	a_{21}	a_{22}	...	a_{2m}
...	...	...	...	...
x_n	a_{n1}	a_{n2}	...	a_{nm}

Игра заключается в том, что каждый из игроков, не имея информации о действиях противника, делает один ход (выбирает одну из своих стратегий). Каждая выбранная пара стра-

тегий – по одной стратегии для каждого игрока – определяет *партию игры*. Партия в свою очередь определяет платеж каждому игроку. Результатом партии является выигрыш одного игрока и проигрыш другого.

Партия игры состоит в том, что каждый игрок принимает одно решение, а именно делает свой выбор стратегии. Повторный выбор стратегий игроками – это второй шаг в игре, вторая партия в игре. Возможен и третий, и четвертый шаги. Иначе говоря, игра может быть многошаговой, т. е. состоять из нескольких партий.

2.2.1. Минимаксные стратегии

Игру Γ_A можно представить себе следующим образом. Игрок P_1 выбирает стратегию i (номер строки платежной матрицы), а игрок P_2 – стратегию j (номер столбца платежной матрицы). В случае выигрыша игрок P_1 получает от своего противника сумму a_{ij}.

Целью игрока P_1 в матричной игре является, естественно, получение по возможности большего выигрыша. Цель же игрока P_2 состоит в том, чтобы дать игроку P_1 возможно меньший выигрыш. И наоборот.

Поэтому разумное поведение игроков в матричной игре должно основываться на следующих рассуждениях. Пусть игрок P_1 выбирает некоторую свою стратегию i. Тогда в наихудшем случае (а в теории игр игроки предполагаются весьма осторожными и рассчитывают на наименее благоприятный для себя поворот событий, такое наименее благоприятное для игрока P_1 положение дел может наступить, например, в том случае, когда стратегия i станет известной игроку P_2) он получит выигрыш $\min\limits_j a_{ij}$, поскольку второй игрок старается проиграть как можно меньше. Предвидя такую ситуацию, игрок P_1 должен выбрать свою стратегию i_0 так, чтобы максимизировать этот свой минимальный выигрыш:

$$\min_j a_{i_0 j} = \max_i \min_j a_{ij}. \tag{2.1}$$

Следовательно, стоящий в правой части написанного равенства «максимин» является гарантированным выигрышем игрока P_1. Стратегия i_0 игрока P_1 называется *максиминной*.

Симметричные рассуждения, проводимые за игрока P_2, показывают, что игрок P_2 должен выбирать такую свою стратегию j_0, что

$$\max_i a_{ij_0} = \min_j \max_i a_{ij}. \tag{2.2}$$

Здесь стоящий справа «минимакс» является тем выигрышем игрока P_1, больше которого он при правильных действиях противника получить не может. Стратегия j_0 игрока P_2 называется *минимаксной*.

Поэтому фактический выигрыш v_1 игрока P_1 должен при разумных действиях партнеров лежать между правыми частями (2.1) и (2.2):

$$\max_i \min_j a_{ij} \le v_1 \le \min_j \max_i a_{ij}. \tag{2.3}$$

Принцип осторожности, заставляющий игроков придерживаться максиминной и минимаксной стратегий соответственно, называют «Принципом минимакса», а минимаксную и максиминную стратегии объединяют общим термином «Минимаксные стратегии».

2.2.2. Игра с седловой точкой

Определение 2.2. Игра Γ_A называется *игрой с седловой точкой*, если выполняется равенство

$$\max_i \min_j a_{ij} = \min_j \max_i a_{ij}. \tag{2.4}$$

В игре с седловой точкой существуют стратегии i_0 и j_0 такие, что

$$a_{i_0 j_0} = \max_i \min_j a_{ij} = \min_j \max_i a_{ij}.$$

Элемент платежной матрицы $a_{i_0 j_0}$ называется *седловой точкой*. Выигрыш игрока P_1 в игре с седловой точкой равен элементу матрицы $a_{i_0 j_0}$ и называется *значением игры*.

В случае игры с седловой точкой игрокам выгодно придерживаться максиминной и минимаксной стратегий и невыгодно отклоняться от них, они являются *оптимальными стратегиями*. Таким образом, в этом случае говорят, что имеется *ситуация равновесия*.

2.2.3. Игра без седловой точкой

Если игра Γ_A не имеет седловой точки, то игроки в игре вынуждены выбирать стратегии независимо друг от друга, случайным образом, отдавая тем самым предпочтение смешанным стратегиям

$$s = (p_1, ..., p_n) \quad \text{и} \quad \sigma = (q_1, ..., q_m)$$

игроков P_1 и P_2 соответственно.

В качестве платежа берется

$$sA\sigma^T = \sum_{i,j} a_{ij} p_i q_j. \tag{2.5}$$

Теорема 3.2 (фон Нейман). *Имеет место равенство*

$$\max_s \min_\sigma sA\sigma^T = \min_\sigma \max_s sA\sigma^T. \tag{2.6}$$

■

Смешанные стратегии s_* и σ_*, для которых выполняется равенство (2.6), называются *оптимальными*. Пара оптимальных стратегий (s_*, σ_*) называется также *ситуацией равновесия в смешанных стратегиях*.

Число $v = s_* A \sigma_*^T$ называется *значением игры* Γ_A. Это тот максимальный выигрыш, который обеспечивает себе игрок P_1 (и это максимальные потери, понесенные игроком P_2). Значение игры Γ_A может быть представлено формулами

$$v = \max_s \min_{1 \le j \le m} sAe_j^T = \min_\sigma \max_{1 \le i \le n} f_i A \sigma^T, \tag{2.7}$$

где

$$e_j = (0, ..., 0, \underset{j}{1}, 0, ..., 0) \in \mathbb{R}^m, \quad f_i = (0, ..., 0, \underset{i}{1}, 0, ..., 0) \in \mathbb{R}^n.$$

2.2.4. Решение матричной игры

Определение 3.2. *Решением игры* Γ_A называется пара оптимальных стратегий.

Таким образом, для разрешения конфликта, описанного как игра, необходимо найти (смешанные) оптимальные стратегии игроков.

Чаще встречаются матричные игры без седловой точки. Тогда для нахождения их решений используются смешанные стратегии.

Поиск решения игр в смешанных стратегиях производится путем сведения к задаче линейного программирования и решения ее симплекс-методом.

2.2.5. Критерии оптимальности стратегии администратора

Предположим, что администратор – игрок P_1 – не имеет никакой информации, какую стратегию избирает злоумышленник – игрок P_2. Как в этих условиях выбрать стратегию i_0 для игрока P_1?

Определение 2 4. Поставим в соответствие каждой стратегии i число $W_i(A)$, вычисляемое с помощью платежной матрицы A. Пусть $W = \max\limits_i W_i(A)$. *Критерий выбора оптимальной стратегии* состоит в том, чтобы взять для игрока P_1 стратегию i_0 такую, что $W_{i_0} = W$.

Критерий крайнего пессимизма Вальда. Игрок P_1 не имеет никакой информации, какую стратегию избирает игрок P_2. Для критерия Вальда $W_i = \min\limits_j a_{ij}$. Следовательно, по критерию Вальда игрок P_1 выбирает такую стратегию i_0, для которой

$$\min_j a_{i_0 j} = \max_i \min_j a_{ij}.$$

Критерий Байеса (критерий максимального математического ожидания). При использовании этого критерия игроку P_1 должны быть известны вероятности, с которыми злоумышленник – игрок P_2 – применяет свои стратегии j. Обозначим эти вероятности соответственно $q_1, q_2, ..., q_m$.

Полагаем

$$W_i = \sum_{j=1}^{m} a_{ij} q_j.$$

Оптимальным можно считать стратегию i_0 игрока P_1, при котором $W_{i_0} = W$.

Критерий недостаточного основания Лапласа. Если вероятности всех стратегий злоумышленника (примерно) равны, то можно пользоваться критерием Лапласа, для которого

$$W_i = \frac{1}{m} \sum_{j=1}^{m} a_{ij}.$$

Критерий пессимизма–оптимизма Гурвица. Берется

$$W_i = c \cdot \min_j a_{ij} + (1 - c) \cdot \max_j a_{ij},$$

$$c \in [0, 1]$$

– коэффициент пессимизма. Крайнему пессимизму ($c = 1$) можно противопоставить крайний оптимизм ($c = 0$, критерий азартного игрока), когда ставка делается на самый большой возможный выигрыш, т. е. на самый большой элемент платежной матрицы.

2.3. Методы решения матричных игр

Существуют различные способы решения матричных игр. Приведем только два из них – *метод доминирования* и *сведение к линейному программированию*.

2.3.1. Доминирование

Пусть дана матрица $A = (a_{ij})$. Строка i доминирует k-ю строку, если

$$a_{ij} \geq a_{kj} \quad \text{для всех } j$$

и

$$a_{ij} > a_{kj} \quad \text{хотя бы для одного } j.$$

Аналогично, столбец j доминирует l-й столбец, если

$$a_{ij} \leq a_{il} \quad \text{для всех } i$$

и

$$a_{ij} < a_{il} \quad \text{хотя бы для одного } i.$$

Теорема 2.2. *Пусть Γ_A – игра с матрицей A, строки $i_1, ..., i_k$ которой доминируются. Тогда игрок P_1 имеет оптимальную смешанную $s = (p_1, ..., p_n)$, для которой $p_{i_1} = ... = p_{i_k} = 0$. При этом оптимальная смешанная стратегия для игры Γ'_A с матрицей A', получаемой вычеркиванием из матрицы A доминируемых строк $i_1, ..., i_k$, является оптимальной и для первоначальной игры Γ_A.* ■

Аналогичная теорема справедлива и для доминируемых столбцов. Таким образом, учет доминирования позволяет сводить игру к игре с с меньшей матрицей.

В ряде игр удобно использовать в качестве решения игры следующий принцип.

Принцип доминирования. Стратегия x_i является доминирующей, если строка i доминирует остальные строки. Оптимальной в таком случае считаем доминирующую стратегию.

2.3.2. Использование линейного программирования

Нахождение оптимальных смешанных стратегий s, σ игры Γ_A сводится к задаче линейного программирования. Покажем, как это делается.

Можно к каждому элементу матрицы A добавить число a так, что получается матрица A^a и игры Γ_A и Γ_{A^a} имеют одинаковые оптимальные стратегии, и для значений двух игр справедливо равенство $v(A^a) = v(A) + a$ [5, с. 30]. Ясно, что число a можно взять так, что для игры Γ_{A^a}

$$v = v(A^a) > 0.$$

Следовательно, без ограничения общности можем считать, что для игры Γ_A ее значение $v(A) > 0$.

Оптимальная смешанная стратегия $s = (p_1, ..., p_n)$ игрока P_1 может быть определена как решение максиминной задачи

$$v = \max_s \min_{1 \le j \le m} sAe_j^T,$$

где

$$s = (p_1, ..., p_n), \quad e_j = (0, ..., 0, \underset{j}{1}, 0, ..., 0) \in \mathbb{R}^m.$$

Полагаем

$$v(s) = \min_j sAe_j^T.$$

Тогда

$$sAe_j^T = \sum_{i=1}^n a_{ij} p_i \ge v(s), \ j = 1, ..., m.$$

Поэтому задачу игрока P_1 можно представить с помощью вспомогательной величины u как задачу линейного программирования, максимизирующую величину

$$v = \max u$$

при ограничениях

$$\sum_{i=1}^n a_{ij} p_i - u \ge 0, \ j = 1, ..., m,$$

$$\sum_{i=1}^n p_i = 1, \quad \forall i (p_i \ge 0).$$

Одновременно, нахождение оптимальной смешанной стратегии σ игроком P_2 – это минимаксная задача:

$$v = \min_{\sigma} \max_{1 \le i \le n} f_i A \sigma^T,$$

где

$$\sigma = (q_1, ..., q_m), \quad f_i = (0, ..., 0, \underset{i}{1}, 0, ..., 0).$$

Следовательно, если взять

$$v(\sigma) = \max_{j} f_j A \sigma^T,$$

то

$$f_i A \sigma^T = \sum_{j=1}^{m} a_{ij} q_j \le v(\sigma), \ j = 1, ..., m,$$

и тогда задачу для игрока P_2 можно представить с помощью вспомогательной величины u как задачу линейного программирования, минимизирующую величину

$$v = \min w,$$

при ограничениях

$$\sum_{j=1}^{m} a_{ij} q_j - w \le 0, \ i = 1, ..., n,$$

$$\sum_{j=1}^{m} q_j = 1, \quad \forall j (q_j \ge 0).$$

Сделаем следующую замену переменных:

$$u_i = \frac{p_i}{u}, \quad v_j = \frac{q_j}{w}.$$

Так как

$$\frac{1}{u} = \sum_i u_i, \quad \frac{1}{w} = \sum_j v_j,$$

то задачи (1.35) и (1.36) эквивалентны соответственно задачам

$$\sum_{i=1}^{n} u_i \to \min, \tag{2.8}$$

$$\sum_{i=1}^{n} a_{ij} u_i \geq 1,\ j = 1, ..., m,$$

$$u_i \geq 0$$

и

$$\sum_{j=1}^{n} v_j \to \max, \tag{2.9}$$

$$\sum_{j=1}^{m} a_{ij} v_j \leq 1,\ i = 1, ..., n,$$

$$v_j \geq 0.$$

Задачи (2.8) и (2.9) являются двойственными друг другу. Как видим, имеет место

Теорема 2.4. *Решение матричной игры Γ_A эквивалентно решению пары двойственных задач линейного программирования.* ■

Если $u^* = (u_1^*, ..., u_n^*)$ – решение задачи линейного программирования (2.8), то

$$v = \max u = \frac{1}{\sum\limits_{i=1}^{n} u_i^*},$$

и, следовательно,

$$s^* = (p_1^*, ..., p_n^*) = v \cdot (u_1^*, ..., u_n^*) = vu^*$$

– оптимальная стратегия игрока P_1.

Аналогично, $v^* = (v_1^*, ..., v_m^*)$ – решение задачи линейного программирования (2.9), то

$$v = \min w = \frac{1}{\sum_{j=1}^{m} v_j^*},$$

и, следовательно,

$$\sigma^* = (q_1^*, ..., q_m^*) = v \cdot (v_1^*, ..., v_m^*) = vv^*$$

– оптимальная стратегия игрока P_2.

Тогда (s^*, σ^*) – решение игры Γ_A.

2.4. Биматричные игры

Игра, в которой выигрыш одного игрока – это проигрыш другого, т. е. игра с нулевой суммой, не всегда адекватно отражает ситуацию противостояния администратора и злоумышленника. У них разные представления об успехе, разные меры ценности информации. Поэтому полезно иметь теорию такой игры, когда для создания защитной системы используется матрица способов средств защит (или матрица вероятностей наступления угроз), а действия злоумышленника описываются своей собственной матрицей нападений (ценностей). Причем в общем случае эта матрица может не совпадать с матрицей защищающейся стороны. Другими словами, желательно иметь теорию с двумя выигрышными матрицами.

Биматричная игра Γ_{AB} – это конечная игра двух лиц, выигрыш каждого из которых задается отдельной матрицей.

Пусть первый игрок имеет n стратегий i, а второй – m стратегий j. Выигрыши первого (P_1) и второго (P_2) игроков задаем двумя отдельными матрицами:

$$A = (a_{ij}) \quad \text{и} \quad B = (b_{ij}).$$

Если игрок P_1 выбирает стратегию i, а игрок P_2 – стратегию j, то выигрыш первого игрока равен a_{ij}, а второго – b_{ij}.

Выигрыши в смешанных стратегиях s и σ игроков P_1 и P_2 соответственно равны

$$sA\sigma^T \quad \text{и} \quad sB\sigma^T.$$

Биматричная игра – это одноходовая игра.

Биматричную игру часто представляют в форме одной матрицы (табл. 2.2).

Таблица 2.2

Платежная матрица

$P_1 \backslash P_2$	y_1	y_2	...	y_m
x_1	$[a_{11}, b_{11}]$	$[a_{12}, b_{12}]$	...	$[a_{1m}, b_{1m}]$
x_2	$[a_{21}, b_{21}]$	$[a_{22}, b_{22}]$	...	$[a_{2m}, b_{2m}]$
...	...	...	...	...
x_n	$[a_{n1}, b_{n1}]$	$[a_{n2}, b_{n2}]$	...	$[a_{nm}, b_{nm}]$

Определение 2.2. Назовем пару стратегий (i_0, j_0) ситуацией *равновесия* (в чистых стратегиях) в *биматричной игре*, если выполняются неравенства

$$a_{ij_0} \le a_{i_0 j_0} \quad \text{для любого } i,$$

$$b_{i_0 j} \le b_{i_0 j_0} \quad \text{для любого } j.$$

Равновесия в чистых стратегиях ищут следующим образом [43]. В каждом столбце матрицы A помечаем звездочкой максимальные элементы. Затем помечаем звездочкой максимальные элементы в каждой строке матрицы B. И наконец, запишите все пары стратегий (i, j), такие, что оба элемента a_{ij} и b_{ij} отмечены звездочкой. Все такие пары и являются равновесиями.

Определение 2.3. Назовем пару смешанных стратегий (s_0, σ_0) ситуацией *равновесия* в *биматричной игре*, если выполняются неравенства

$$\forall s(sA\sigma_0^T \le s_0 A\sigma_0^T) \quad \text{и} \quad \forall \sigma(s_0 B\sigma^T \le s_0 B\sigma_0^T).$$

Теорема 2.4 ([5, с. 103]). *Биматричная игра имеет ситуацию равновесия в смешанных стратегиях.* ■

2.5. Равновесия Нэша в конечной игре N лиц

Рассмотрим бескоалиционную конечную игру N лиц, которую ведут N игроков $P_1, ..., P_N$. С каждым игроком P_i связано конечное множество *чистых стратегий*

$$x^i_{\alpha_i} \ (\alpha_i = 1, ..., n_i),$$

и его собственная функция выигрыша

$$\pi_i(x^1_{\alpha_1}, ..., x^N_{\alpha_N}) \in \mathbb{R}. \tag{2.10}$$

Кортеж длины N

$$x = (x^1_{\alpha_1}, ..., x^N_{\alpha_N}), \tag{2.11}$$

состоящий из набора чистых стратегий, будем называть чистой *ситуацией*. Чистые ситуации обозначаем x, y,...

Игра состоит в том, что делается **один ход**, к котором каждый игрок P_i выбрал свою стратегию $x^i_{\alpha_i}$. Иначе говоря, имеем ситуацию (2.11), в которой выигрыш каждого игрока P_i равен (2.10).

Для удобства будем через $(x||y^i_{\alpha_i})$ обозначать ситуацию $(x^1_{\alpha_1}, ..., x^{i-1}_{\alpha_{i-1}}, y^i_{\alpha_i}, x^{i+1}_{\alpha_{i+1}}, ..., x^N_{\alpha_N})$.

Определение 2.1. Чистая ситуация x называется *ситуацией чистого равновесия* тогда и только тогда, когда для каждого игрока P_i

$$\pi_i(x) = \max_{y^i_{\alpha_i}} \{\pi_i(x||y^i_{\alpha_i})\}. \tag{2.12}$$

Однако не каждая бескоалиционная конечная игра имеет ситуацию чистого равновесия. Поэтому приходится рассмотреть, как и в случае матричных игр, смешанные стратегии.

Под *смешанной стратегией* игрока P_i будет пониматься набор неотрицательных чисел, сумма которых равна единице и которые поставлены в однозначное соответствие чистым стратегиям этого игрока. Для представления такой смешанной стратегии игрока P_i мы будем писать

$$s_i = \sum_{\alpha=1}^{n_i} p_\alpha^i x_\alpha^i,$$

где

$$\sum_{\alpha=1}^{n_i} p_\alpha^i = 1, \quad \text{и} \quad \forall i\alpha(p_\alpha^i \geq 0).$$

Мы можем рассматривать смешанные стратегии s_i как точки симплекса, вершинами которого являются чистые стратегии x_α^i. Этот симплекс можно рассматривать как выпуклое подмножество векторов вещественного векторного пространства. Смешанные стратегии оказываются, таким образом, линейными комбинациями чистых.

Выражения s_i, t_i, r_i и т. д. будут означать смешанные стратегии.

Функция выигрыша π_i, введенная в вышеприведенном определении конечной игры, имеет единственное распространение на кортежи длины N, состоящие из смешанных стратегий[1]

$$(s_1, ..., s_N),$$

которое линейно относительно смешанной стратегии каждого из игроков (N-линейна). Это распространение также будем обозначать через π_i, записывая его значения как $\pi_i(s_1, ..., s_N)$.

Для обозначения ситуации в смешанных стратегиях будем писать $\mathfrak{f}$ или $\mathfrak{t}$. Если $\mathfrak{f} = (s_1, ..., s_N)$, то положим $\pi_i(\mathfrak{f}) = \pi_i(s_1, ..., s_N)$. Такая ситуация может рассматриваться поэтому как точка векторного пространства, являющегося произведением векторных пространств, содержащих множества

[1] Называемые *ситуациями* в смешанных стратегиях.

смешанных стратегий игроков. Множество всех таких ситуаций образует, очевидно, выпуклый многогранник, являющийся произведением симплексов, представляющих множества смешанных стратегий.

Для удобства будем через $(\mathfrak{f}||t_i)$ обозначать ситуацию $(s_1, ..., s_{i-1}, t_i, s_{i+1}, ..., s_N)$, где $\mathfrak{f} = (s_1, ..., s_N)$.

Определение 2.1. Ситуация $\mathfrak{f}$ называется *ситуацией равновесия Нэша* в смешанных стратегиях тогда и только тогда, когда для каждого игрока P_i

$$\pi_i(\mathfrak{f}) = \max_{r_i}\{\pi_i(\mathfrak{f}||r_i)\}. \tag{2.13}$$

Таким образом, ситуация равновесия является такой ситуацией $\mathfrak{f}$, что каждая входящая в нее смешанная стратегия игрока максимизирует выигрыш этого игрока, если стратегии остальных игроков остаются неизменными. Тем самым в такой ситуации стратегия каждого из игроков оказывается оптимальной против стратегий остальных игроков.

Теорема 2.1 (Нэш, [39]). *Каждая бескоалиционная конечная игра имеет хотя бы одну ситуацию равновесия Нэша в смешанных стратегиях.* ■

Принцип равновесия Нэша. Стратегии, образующие ситуацию равновесия Нэша, называем оптимальными стратегиями в смысле *критерия оптимальности Нэша*.

При пребывании в равновесии Нэша игроки добиваются устойчивого равновесия. Игрокам выгодно сохранять это равновесие, так как любое изменение, отклонение от оптимальной стратегии ухудшит их положение. Поведение игроков должно соответствовать *нэшевской рациональности*, согласно которой необходимо подчас пожертвовать личными интересами, учитывать интерес противника. Но столь рациональными (разумными) должны быть оба игрока.

Нэшевская рациональность свойственна людям со *стратегическим мышлением*. Однако люди не настолько совершенны, чтобы все время мыслить стратегически. Они начинают

действовать иррационально, не веря в рациональность противника, и, как результат, их ситуация, их выигрыш, отклонясь от равновесия, ухудшаются. Для преодоления ограничения, накладываемого нэшевской рациональностью, разрабатываются эволюционные формулировки равновесия, для которых свойственны более слабые допущения по уровню рациональности.

2.6. Дилемма заключённого

Иллюстрацией того, как людям трудно мыслить стратегически и вести себя рационально в случае конфликта интересов, является рассуждение, принадлежащее математику и специалисту по теории игр Такеру и известное под названием «Дилемма заключённого».

Такер следующим образом проиллюстрировал теорему Нэша о равновесии для игры с ненулевой суммой.

> Однажды два взломщика Боб и Джон были схвачены полицией и допрошены по отдельности. Каждый из них имел выбор: признаться или молчать.
>
> Если бы никто из них не сознался, то обоим грозил бы год тюремного заключения, так как при них были обнаружены инструменты взлома. Если бы каждый из них признался и указал на напарника, то оба отправились бы в тюрьму на 10 лет, поскольку пошел на сотрудничество со следствием. Однако, если бы кто-то из них сознался и указал на напарника, а тот в свою очередь не признался бы, то взломщик, оказавший содействие полиции был бы выпущен на свободу, а непризнавшийся был бы посажен на 20 лет.
>
> Платежные матрицы возможных исходов для Боба и Эла таковы:

Таблица 2.3

Платежная матрица игры Боба

Дилемма узника	Джон, молчание	Джон, признание
Боб, молчание	1 год	20 лет
Боб, признание	свобода	10 лет

Таблица 2.4

Платежная матрица игры Джона

Дилемма узника	Джон, молчание	Джон, признание
Боб, молчание	1 год	свобода
Боб, признание	20 лет	10 лет

В форме биматричной игры имеем матрицу:

Таблица 2.5

Игра «Дилемма заключенного»

Дилемма узника	Джон, молчание	Джон, признание
Боб, молчание	[1 год, 1 год]	[свобода, 20 лет]
Боб, признание	[20 лет, свобода]	[10 лет, 10 лет]

Стратегии в этой игре – признание или молчание.

Какой выбор сделает каждый их них? Боб рассуждает так: «Если Джон признается и укажет на меня, а я – нет, то я получу 20 лет. Если он признается и предаст меня и я тоже признаюсь и предам его, то оба получим по 10 лет. Если Джон не признается, а я признаюсь и укажу на него, то буду свободен, а он сядет на 20 лет. В любом случае, мне будет лучше всего, если я признаюсь». Но Джон, находясь в той же ситуации, приходит к тому же выводу. Они оба пришли, как им кажется, к единственному разумному рациональную решению, а в результате оба отправились за решетку на 10 лет. В то же время, если бы они поступили «иррационально» и молчали, то отбыли бы в тюрьме всего

по году. Ведь с точки зрения «равновесия Нэша» Боб и Джон должны оба молчать, и в таком случае каждый из них гарантированно получил бы минимальный срок.

Но узники поступают «рационально». На деле же рациональное поведение оказывается нерациональным. Думая, что ему выгодно отклониться от равновесной стратегии, и считая, что так же поступит другой узник, другой игрок, он и его напарник оказываются в сильном проигрыше.

Иначе говоря, дилемма показывает, что следование личным интересам приводит к тому, что оба игрока оказываются в худшей ситуации в сравнении с той, в которой они пожертвовали бы личными интересами.

2.7. Программное обеспечение для нахождения решения игр

Для нахождения решений теоретико-игровых задач, получения чистых и смешанных равновесий Нэша для различного типа игр созданы компьютерные программы. К сожалению, их не так много, как хотелось бы.

Программа Gambit.[2] Это открытая библиотека игрового программного обеспечения. Содержит теорию и инструменты для построения и анализа конечных игр в нормальной и экстенсивной формах. Gambit разработан для разных платформ; работает под Linux, Mac OS X, и Windows.

Основными разработчиками Gambit являются: Теодор Туроси (Англия), Ричард Д. МакКелви (США, основатель проекта), Эндрю МакЛеннан (университет Квинсленда).

[2]Находится на сайте http://www.gambit-project.org/doc/index.html. Версии Gambit доступны для загрузки по следующему адресу: http://sourceforge.net/projects/gambit/files

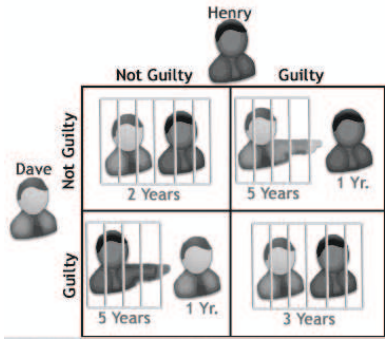


Рис. 2.2. Иллюстрация дилеммы заключенного распространителя продукта Game Theory SoftWare (Super)

Программа GarlicSim.[3] Открытый продукт. Написан на языке Python. Теория игр лишь одна из возможностей пакета.

Разработчик: Ram Rachum (Израиль).

Программа Game Theory SoftWare (Super) 1.0. После установки программа требует покупки ключа. На рис. 2.2 дана рекламная картинка распространителя продукта.

Библиотеки для вычислений по теории игр имеются в известных пакетах Maple, MatLab и MathCAD.

2.8. Бесконечные игры

Когда число стратегий x, y игроков P_1 и P_2 соотвественно бесконечно и $x \in X, y \in Y$, где X, Y – множества произвольной природы, то функция выигрыша – это отображение $f : X \times \times Y \to \mathbb{R}$. *Седловая точка* в этом случае для стратегической игры – пара (x_0, y_0), удовлетворяющая условиям:

$$\forall x \in X \forall y \in Y (f(x, y_0) \leq f(x_0, y_0) \leq f(x_0, y)).$$

[3]Находится на сайте https://pypi.python.org/pypi/ (см. список пакетов).

Седловая точка существует, если $X \subset \mathbb{R}^n, Y \subset \mathbb{R}^m$ – выпуклые компакты, f непрерывна, $\forall y f(x,y)$ вогнута по x, а $\forall x f(x,y)$ выпукла по y [5, с.16]. Под *смешанными стратегиями* игроков понимаются в таком случае вероятностные распределения на множествах стратегий X, Y. Иначе говоря, смешанная стратегия игрока – это случайный выбор стратегии, определяемый вероятностным распределением. Если $X = [a,b], Y = [c,d]$ и f непрерывна, то всякая игра имеет решение в смешанных стратегиях [5, с. 24].

2.9. Джон фон Нейман

Джон фон Нейман

Джон фон Нейман – американский математик и физик[4], создатель математической теории игр, которая сыграла важную роль в экономике. Родился в Венгрии.

Работал в различных областях науки: функциональный анализ, квантовая механика, логика, метеорология. Внёс большой вклад в создание первых ЭВМ и разработку методов их применения.

В 1925 году фон Нейман получает диплом инженера-химика в Цюрихе и успешно защищает диссертацию «Аксиоматическое построение теории множеств» на звание доктора философии в Будапештском университете. Совершенствует свои знания в знаменитом Геттингенском университете, где в то время читали лекции люди, чьи имена стали гордостью науки: К. Рунге, Ф. Клейн, Э. Ландау, Д. Гильберт, Э. Цермело, Г. Вейль, Г. Минковский, Ф. Франк, М. Борн и др. Приглашенными лекторами были Г. Лоренц, Н. Бор, М. Планк, П. Эренфест, А. Пуанкаре, А. Зоммерфельд и др.

Совместно с Д. Гильбертом и Л. Нордгеймом фон Нейман написал статью «Об основаниях квантовой механики».

[4] См.: http://encyklopedia.narod.ru/bios/nauka/neumann/neumann.html

В 1927 году фон Нейман становится приват-доцентом Берлинского, а с 1929 года – Гамбургского университета.

В 1929 году фон Нейман получает приглашение прочитать в течение одного семестра цикл лекций в Принстонском университете. В США он впервые оказался в 1930 году и вскоре после приезда для многих коллег становится просто Джонни. В 1931 г. фон Нейман окончательно расстаётся с Гамбургским университетом, чтобы принять профессуру в Принстоне.

В 1944 году он и Оскар Моргенштерн создали теорию игр, которая изучала игры с нулевой суммойы . Теория впервые была изложена в их книге «Теория игр и экономическое поведение» (Theory of Games and Economic Behavior).

В 1943–1946 годах была построена первая ЭВМ в школе инженеров-электриков Мура Пенсильванского университета, получившая название ЭНИАК (по первым буквам английского названия – электронный цифровой интегратор и вычислитель). Фон Нейман подсказал её разработчикам, как можно модифицировать ЭНИАК, чтобы упростить его программирование.

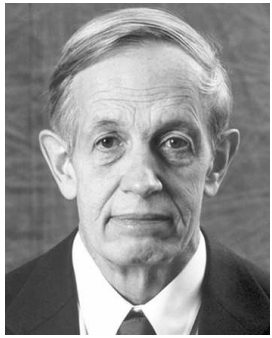
2.10. Джон Нэш

Джон Форбс Нэш-младший (англ. John Forbes Nash, Jr.; род. 13 июня 1928, Блюфилд, Западная Виргиния) – американский математик, работающий в области теории игр и дифференциальной геометрии, лауреат нобелевской премии по экономике 1994 года «За анализ равновесия в теории некооперативных игр» (вместе с Райнхардом Зельтеном и Джоном Харсани)[5].

После окончания школы учился в Политехническом институте Карнеги (ныне частный Университет Карнеги-Меллона), где Нэш пробовал изучать химию, прослушал курс международной экономики, а потом окончательно утвердился в решении заняться математикой. В 1947 году, окончив институт с двумя дипломами – бакалавра и магистра, – он поступил в Принстонский университет.

[5]Использована статья о Нэше из «Википедии».

В Принстоне Джон Нэш узнал о теории игр, в ту пору только созданной Джоном фон Нейманом и Оскаром Моргенштерном. Она поразила его воображение, да так, что в 20 лет Джон Нэш сумел создать основы научного метода, сыгравшего огромную роль в развитии мировой экономики. В возрасте двадцати одного года (в 1949 г.) им была написана диссертация по теории игр. Сорок пять лет спустя он получил за эту работу нобелевскую премию по экономике. Научным руководителем был Альберт Такер.



Джон Форбс Нэш

В 1950–1953 годах Нэш опубликовал четыре революционные работы по теории игр с ненулевой суммой – класса игр, в которых сумма выигрыша выигравших участников не равна сумме проигрыша проигравших участников. Примером такой игры могут стать переговоры об увеличении зарплаты между профсоюзом и руководством компании. Эта ситуация может завершиться либо длительной забастовкой, в которой пострадают обе стороны, либо достижением взаимовыгодного соглашения. Нэш сумел разглядеть новое лицо конкуренции, смоделировав ситуацию, впоследствии получившую название «равновесие по Нэшу» или «некооперативное равновесие», при которой обе стороны используют идеальную стратегию, что и приводит к созданию устойчивого равновесия. Игрокам выгодно сохранять это равновесие, так как любое изменение только ухудшит их положение.

Джон Нэш получил серьезные результаты в геометрии. Его преследовало тяжелое заболевание – шизофрения. Жизнь Нэша представлена в художественном американском фильме «Игры разума».

Глава 3

Пример матричной игры «злоумышленник – администратор»

В этой главе рассматривается пример игры между злоумышленником и администратором компьютерной системы [62].

Показано, как находится оптимальная стратегия в случаях, когда либо доступна, либо недоступна априорная информация о частоте появления конкретных типов угроз.

3.1. Игра с матрицей вероятностей

Предположим, что записи в игровой матрице представляют **вероятности эффективности использования** администратором компьютерной системы трех стратегий (строки x_1, x_2, x_3) против пяти угроз (колонки y_A, y_B, y_C, y_D, y_E).

Будем в качестве примера рассматривать игру со следующей платежной матрицей:

Таблица 3.1

Платежная матрица

	y_A	y_B	y_C	y_D	y_E
x_1	0,3	0,6	0,4	0,5	0
x_2	1	0	0	0	0
x_3	1	0,5	0	0	1

3.1.1. Случай отсутствия априорной частотной информации о типах угроз

Равновесная пара (строка, колонка) чистых стратегий – это седловая точка в табл. 3.1, которая является минимумом в строке и максимумом в колонке [63]. Однако в табл. 3.1 нет никакой седловой точки. Поэтому в данной игре отсутствуют как равновесная пара, так и пара оптимальных чистых стратегий.

Но, по теореме фон Неймана, любая игра обладает хотя бы одной равновесной парой смешанных стратегий.

Поскольку каждое число столбца y_C не больше, чем число в той же самой строке столбцов y_B или y_D, то столбец y_C доминирует над столбцами y_B и y_D. Следовательно оба столбца y_B и y_D могут быть устранены из матрицы игры, не изменяя равновесной пары для матрицы игры, данной в табл. 3.1. Аналогично столбец y_E доминирует над столбцом y_A, и столбец y_A также может быть устранен из матрицы игры.

В результате имеем следующую матрицу игры:

0,4	0
0	0
0	1

С другой стороны, каждое число в строке 1 в матрице выше не меньше числа в той же самой колонке строки 2, то есть строка 1 доминирует над строкой 2. Следовательно, строка 2 может быть устранена из вышеупомянутой матрица игры, не изменяя равновесной пары:

0,4	0
0	1

Финальная матрица игры приведена в таблице 3.2.

Таблица 3.2

Матрица игры

	y_C	y_E
x_1	0,4	0
x_3	0	1

Согласно [63], если s – смешанная стратегия для строки (администратора) и σ – смешанная стратегия для колонки (злоумышленника), то имеется равновесная пара (s^*, σ^*) для матрицы игры

a	b
c	d

где $s^* = (p, 1-p)$ и $\sigma^* = (q, 1-q)$ и

$$p = (d-c)/R,\ q = (d-b)/R,\ R = a-b-c+d. \qquad (3.1)$$

Взяв $a = 0,4, b = c = 0$ и $d = 1$, получаем $R = 1,4, p = 1/1,4 = 5/7, q = 1/1,4 = 5/7$.

Таким образом, $s^* = \sigma^* = (5/7, 2/7)$.

Поэтому в игре с табл. 3.1 оптимальная стратегия защиты противостояния пяти угрозам состоит в использовании стратегии x_1 в течение $5/7$ времени работы ресурса и стратегии x_3 в течение $2/7$ времени.

3.1.2. Известна априорная информации о частоте появления угроз

Предположим, что пять угроз появляются с частотами $(0,1; 0,3; 0,3; 0,1; 0,2)$.

Эффективность для чистой стратегии x_1 равна

$$0,3 \cdot 0,1 + 0,6 \cdot 0,3 + 0,4 \cdot 0,3 + 0,5 \cdot 0,1 + 0 \cdot 0,2 = 0,38.$$

Точно так же эффективность для чистой стратегии x_2 равна 0,1, а для чистой стратегии x_3 – $1 \cdot 0,1 + 0,5 \cdot 0,3 + 1 \cdot 0,2 = = 0,45$.

Следовательно, оптимальной чистой стратегией для администратора является стратегия x_3.

3.2. Игра с матрицей затрат

Рассмотрим игру с нулевой суммой администратора со стратегиями x_1, x_2 и x_3 и злоумышленника, имеющего стратегии y_1, y_2 и y_3. Матрица игры – это матрица **затрат**, которые надо понести для закупки и установки защитного оборудования и программ против каждой из трех угроз (в тыс. руб, см. табл. 3.3). Одновременно это урон, который наносит злоумышленник в случае успешной атаки, и, следовательно, это выигрыш злоумышленника.

Таблица 3.3

Матрица игры

	y_1	y_2	y_3
x_1	3	4	2
x_2	1	5	3
x_3	2	1	2

Ожидаемая выгода владельца ресурса складывается из его капитала за вычетом затрат на систему защиты и ущерб от удачной атаки злоумышленника.

Например, если администратор использует стратегию x_1, то противостояние трем угрозам y_1, y_2, y_3 выражается в затратах по установке защитного оборудования и программ и сводится к тому, что необходимо инвестировать сумму в размере 9 тыс. руб. (9=3+4+2).

Положим владелец располагал суммой в 40 тыс. руб. Злоумышленник, используя стратегию y_1, может нанести ущерб в 3 тыс. руб. Тем не менее ожидаемая остаточная сумма средств для владельца (администратора), использующего стратегию

x_1, составляет 40–9–3=28 тыс. руб. Без системы защиты потери могли бы быть больше, поскольку все ресурсы оказались бы во власти злоумышленника.

В табл. 3.3 нет седловых точек, поэтому в игре нет равновесной пары или оптимальных чистых стратегий. Однако, согласно теореме фон Неймана, у матрицы игры есть по крайней мере одна равновесная пара смешанных стратегий.

Пусть администратор использует смешанную стратегию $s = (p_1, p_2, p_3)$, где $p_i \geq 0, i = 1, 2, 3$,

$$p_1 + p_2 + p_3 = 1, \tag{3.2}$$

а злоумышленник использует смешанную стратегию $\sigma = (q_1, q_2, q_3)$, где $q_i \geq 0, i = 1, 2, 3$,

$$q_1 + q_2 + q_3 = 1. \tag{3.3}$$

Для оптимальной смешанной стратегии администратора и для любой чистой стратегии $y_j = e_j$ злоумышленника платеж $sAy_j^T = v$ $(j = 1, 2, 3)$, где v – значение игры. Поэтому

$$\begin{aligned} 3p_1 + p_2 + 2p_3 &= v, \\ 4p_1 + 5p_2 + p_3 &= v, \\ 2p_1 + 3p_2 + 2p_3 &= v. \end{aligned} \tag{3.4}$$

Используя (3.2), отсюда получаем $p_1 = 2/9, p_2 = 1/9, p_3 = 2/3$ и $v = 19/9$.

Аналогично, для оптимальной смешанной стратегии злоумышленника σ и для любой чистой стратегии $x_i = f_i$ администратора выплата $x_i^T A\sigma = v$ $(i = 1, 2, 3)$. Поэтому

$$\begin{aligned} 3q_1 + 4q_2 + 2q_3 &= v, \\ q_1 + 5q_2 + 3q_3 &= v, \\ 2q_1 + q_2 + 2q_3 &= v. \end{aligned} \tag{3.5}$$

Они дают $q_1 + 3q_2 = 0$, следовательно $q_j = 0, j = 1, 2, 3$. Это противоречит (3.3). Следовательно, одна из чистых стратегий администратора отсутствует, т. е. одна из $p_i = 0, i = 1, 2, 3$.

1) $p_1 = 0$. Матрица игры в этом случае имеет вид:

	y_1	y_2	y_3
x_2	1	5	3
x_3	2	1	2

Первая колонка доминирует над третьей. Поэтому предыдущая матрица сводится к матрице

	y_1	y_2
x_2	1	5
x_3	2	1

Из (3.1) получаем: $s = (0, 1/5, 4/5)$ и $\sigma = (4/5, 1/5, 0)$. Имеем $\min\max = \min\{1/5 + 8/5, 5/5 + 4/5, 3/5 + 8/5\} = 9/5$, $\max\min = \max\{12/5 + 4/5, 4/5 + 5/5, 8/5 + 1/5\} = 16/5$. Так как $\min\max \neq \max\min$, то точка равновесия отсутствует. Это говорит о том, что предположение $p_1 = 0$ не верно.

2) $p_2 = 0$.

Матрица игры в этом случае имеет вид:

	y_1	y_2	y_3
x_1	3	4	2
x_3	2	1	2

Третья колонка доминирует первую. Поэтому предыдущая матрица сводится к матрице

	y_2	y_3
x_1	4	2
x_3	1	2

Из (3.1) получаем: $s = (1/3, 0, 2/3)$ и $\sigma = (0, 0, 1)$. Имеем $\min\max = \min\{3/3 + 4/3, 4/3 + 2/3, 2/3 + 4/3\} = 2$, $\max\min = = \max\{2, 3, 2\} = 3$. Так как $\min\max \neq \max\min$, то точка равновесия отсутствует и, следовательно, в действительности $p_2 \neq 0$.

3) $p_3 = 0$.

Матрица игры в этом случае имеет вид:

	y_1	y_2	y_3
x_1	3	4	2
x_2	1	5	3

Первая колонка доминирует вторую. Поэтому предыдущая матрица сводится к матрице

	y_1	y_3
x_1	3	2
x_2	1	3

Из (3.1) получаем: $s = (2/3, 1/3, 0)$ и $\sigma = (1/3, 0, 2/3)$. Имеем $\min\max = \min\{6/3 + 1/3, 8/3 + 5/3, 4/3 + 3/3\} = 7/3$, $\max\min = \max\{3/3 + 4/3, 1/3 + 6/3, 2/3 + 4/3\} = 7/3$. Так как $\min\max = \max\min$, то (s, σ) – точка равновесия.

Следовательно, оптимальная стратегия администратора против трех стратегий злоумышленника состоит в использовании стратегии x_1 в течение $2/3$ времени работы ресурса и стратегии x_2 в течение $1/3$ времени.

Глава 4

Программное приложение для выбора оптимального набора средств защиты

В настоящее время, по мере развития и расширения сферы применения средств вычислительной техники, острота проблемы обеспечения безопасности компьютерных систем и защиты хранящейся и обрабатываемой в них информации от различных угроз всё более возрастает. В первую очередь эта проблема связана с широким распространением локальных, особенно глобальных, компьютерных сетей. Защита информации необходима для уменьшения вероятности утечки (разглашения), модификации (умышленного искажения) или утраты (уничтожения) информации, представляющей определенную ценность для ее владельца.

Сегодня на рынке представлено огромное разнообразие средств защиты компьютерных систем, и администратору безопасности приходится принимать субъективные решения о

выборе в пользу тех или иных программных продуктов. Использование теории матричных игр позволяет обеспечить оптимизацию выбора программных продуктов для защиты компьютерной информации.

В этой главе представлено программное приложение [8], основанное на применении теории матричных игр и автоматизирующее выбор оптимального набора средств защиты.

4.1. Постановка задачи и игровой подход

Для поиска наиболее оптимальных стратегий защиты информационных ресурсов можно провести математическую игру двух сторон, одной из которых является система защиты компьютерной информации, а с другой – возможные атаки злоумышленников. Нанесение хакером ущерба обычно является скорее следствием его действий, а не самой целью. В действительности при атаке он может преследовать какие-то свои цели, порой известные лишь ему. Поскольку целью данной работы являлось определение администратором безопасности оптимальной стратегии защиты, а цели атакующих злоумышленников были неважны, то можно считать, что злоумышленник увлечен желанием нанести как можно больший ущерб атакуемой компьютерной системе. При таком предположении выигрыш злоумышленника будет равен проигрышу администратора безопасности и можно получить матрицу для игры двух лиц с нулевой суммой.

В качестве стратегий хакера будем понимать строки x_i $(i = 1, ..., n)$ некоторой матрицы, а в качестве стратегий администратора безопасности – ее столбцы y_j $(j = 1, ..., m)$. К стратегиям злоумышленника можно отнести различные виды компьютерных атак, а к стратегиям администратора – различные средства защиты компьютерной информации.

В настоящее время рынок может предложить огромное количество программных продуктов, обеспечивающих защиту

Таблица 4.1

Таблица матричной игры

		y_1	y_2	...	y_m
x_1	$p(x_1)$	a_{11}	a_{12}	...	a_{1m}
x_2	$p(x_2)$	a_{21}	a_{22}	...	a_{2m}
...	...	...	...	...	...
x_n	$p(x_n)$	a_{n1}	a_{n2}	...	a_{nm}

компьютерной информации. Чтобы ограничиться выбором конечного списка программных продуктов, был исследован ресурс Anti-Malware.ru, который проводит независимую экспертизу персональных и корпоративных продуктов и сервисов по информационной безопасности. Анализ выбранных средств защиты позволяет каждому программному продукту сопоставить возможность устранить определенные угрозы.

Для проведения на компьютере игры надо также знать результаты игры при каждой паре стратегий x_i и y_j (например, a_{ij} – причинённый хакером материальный ущерб) и вероятности реализации атак злоумышленников $p(x_i)$ при выбранной стратегии x_i. Вероятности реализации атак $p(x_i)$ могут быть определены по результатам статистических исследований. Полезно установить систему обнаружения атак злоумышленников (IDS), например, свободно распространяемую систему Honeynet. Она позволяет набрать статистику, с помощью которой можно выявить наиболее распространённые типы атак x_i и вычислить вероятности $p(x_i)$ хакерских стратегий. Если вероятности атак неизвестны, то можно предположить, что все они равновероятны, т. е. $p(x_i) = 1/n$.

В качестве коэффициентов a_{ij} матрицы игры можно рассматривать, например, годовые потери для всех вариантов комбинаций x_i $(i = 1, ..., n)$ и y_j $(j = 1, ..., m)$. Для этого нужно сопоставить каждую атаку с каждым методом защиты и определить ущерб, который может быть при этом нанесён. Покупка, установка и использование средств защиты могут требовать дополнительных затрат, что также нужно вносить в

ущерб при расчётах.

Построив игровую матрицу (табл. 4.1) и проанализировав её, можно заранее оценить затраты каждого решения по защите компьютерной информации и выбрать наиболее эффективные варианты для всего диапазона атак. Если построена игровая матрица , в которой результатами игры являются материальные потери от атак, то наилучшей в условиях имеющейся информации об атаках будет стратегия системы защиты компьютерной информации y_i, при которой будут минимальны средние потери, т. е. будет минимальна сумма:

$$\sum_{i-1}^{n} a_{ij} p(x_i).$$

Для выбора наиболее оптимального набора средств защиты компьютерных информационных ресурсов в математической игре следует использовать в качестве стратегий различные сочетания из атак и методов защиты. Прекращение использования или добавление нового средства (атаки или защиты) можно рассматривать как переход от одной стратегии к другой.

Целью игрока I (злоумышленника) в матричной игре является, естественно, получение по возможности большего выигрыша. Цель же игрока II (администратора) состоит в том, чтобы дать злоумышленнику I возможно меньший выигрыш. Поэтому разумное поведение игроков в матричной игре должно основываться на следующих рассуждениях.

Пусть игрок I выбирает некоторую свою стратегию x_i. Тогда в наихудшем случае (а в теории игр игроки предполагаются весьма осторожными и рассчитывают на наименее благоприятный для себя поворот событий; такое наименее благоприятное для игрока I положение дел может наступить, например, в том случае, когда стратегия x_i станет известной игроку II) он получит выигрыш

$$\min_j a_{ij}.$$

Предвидя такую возможность, игрок I должен выбрать свою

стратегию x_{i_0} так, чтобы максимизировать этот свой минимальный выигрыш:

$$\min_j a_{i_0 j} = \max_i \min_j a_{ij}. \tag{4.1}$$

Значит, стоящий в правой части написанного равенства «максимин» является гарантированным выигрышем игрока I. Симметричные рассуждения, проводимые за игрока II, показывают, что игрок II должен выбирать такую свою стратегию y_{j_0}, что

$$\min_i a_{ij_0} = \min_j \max_i a_{ij}. \tag{4.2}$$

Здесь стоящий справа «минимакс» является тем выигрышем игрока I, больше которого он при правильных действиях противника получить не может. Поэтому фактический выигрыш игрока I должен при разумных действиях партнеров лежать между правыми частями формул (4.1) и (4.2).

4.2. Расчет ущерба от применения злоумышленником тех или иных стратегий

Для расчета ущерба, наносимого той или иной стратегией злоумышленника, требуются следующие данные:

- Общее число угроз.
- Набор угроз из текущей стратегии злоумышленника.
- Величины ущерба для каждой угрозы.
- Общее число средств защиты.
- Набор средств защиты из текущей стратегии администратора безопасности.
- Стоимость каждого из средств защиты.
- Соотношение каждого средства защиты с угрозами, от которых оно защищает.

Ущерб складывается из величин ущерба D_1, который может быть нанесен при реализации текущей стратегии зло-

умышленника, если система не была защищена от нее средствами защиты из текущей стратегии администратора безопасности, и из общей стоимости D_2 всех средств защиты из текущей стратегии администратора безопасности.

Для подсчета общей стоимости нужных средств защиты D_2 необходимо сопоставить текущий набор средств защиты и все имеющиеся средства защиты и суммировать величины стоимости тех средств, которые присутствуют в первом наборе.

Подсчет ущерба от реализации угроз D_1 вычисляется в два этапа. Сначала текущая стратегия злоумышленника сопоставляется с каждым из средств защиты из текущей стратегии администратора безопасности, и если средство защищает от каких-то угроз из текущего набора, то данные угрозы удаляются из набора. Сопоставив текущий набор угроз со всеми средствами из текущей стратегии администратора безопасности, получаем некоторое количество угроз, от которых система в данном случае не защищена. Полученные угрозы нужно сопоставить со всеми имеющимися угрозами и суммировать величины ущерба тех угроз, что присутствуют в полученном наборе. Далее эти две суммы D_1 и D_2 складываются и получается ущерб при применении текущей пары стратегий злоумышленника и администратора безопасности.

Пример 4.1. Если хакер применяет стратегии x_2, x_3, x_6, x_8, а администратор – y_5, y_8, y_9. Тогда $D_2 = s_5 + s_8 + s_9$, а из табл. 4.2 видно, что от угрозы x_6 текущая защита не работает и, следовательно, $D_1 = d_6$. Здесь s_i – стоимость средства защиты x_i, p_j – ущерб от атаки x_j. Общая сумма ущерба равна $D_1 + D_2 = d_6 + s_5 + s_8 + s_9$.

4.3. Вычисление оптимальной стратегии для администратора безопасности

Для вычисления оптимальной стратегии необходимы следующие данные:

• Все возможные комбинации из угроз, которые может реализовать злоумышленник.

• Все возможные комбинации из продуктов, обеспечивающих защиту.

• Величины ущерба от применения тех или иных пар стратегий (см. § 4.2).

В качестве основы для расчетов берется формула (4.2) из § 4.1. Для начала составляется таблица (матрица), строками которой являются стратегии злоумышленника, а столбцами – стратегии администратора безопасности. На пересечении стратегий ставятся величины ущерба, рассчитанные по алгоритму из § 4.2.

Так как предполагается, что злоумышленник стремится нанести как можно больший вред компьютерной системе, то необходимо для каждой стратегии администратора безопасности выбрать максимальную величину ущерба среди значений, соответствующих текущей стратегии и всем стратегиям злоумышленника. Таким образом, для каждой стратегии администратора безопасности вычисляется максимально возможный ущерб. Логично теперь из всех полученных максимальных величин ущерба выбрать минимальное значение, т.е.

$$\min \max\{D_1 + D_2\}.$$

Стратегия, соответствующая данному значению, и будет искомой *оптимальной стратегией.*

4.4. Описание программного продукта

В данной работе был реализован программный продукт, который по введённым значениям стоимости средств защиты и ущерба от применения всех возможных пар атака–защита вычисляет оптимальный набор из имеющихся в его базе программных продуктов. Этот программный продукт представляет собой веб-приложение, полностью выполняющееся на стороне клиента. Интерфейс его создается с помощью HTML и

CSS, взаимодействие с пользователем и простые операции осуществляются языком JavaScript, а сложные ресурсоемкие вычисления доверяются Java-апплету. На рис. 4.1 представлена главная страница реализованного приложения.

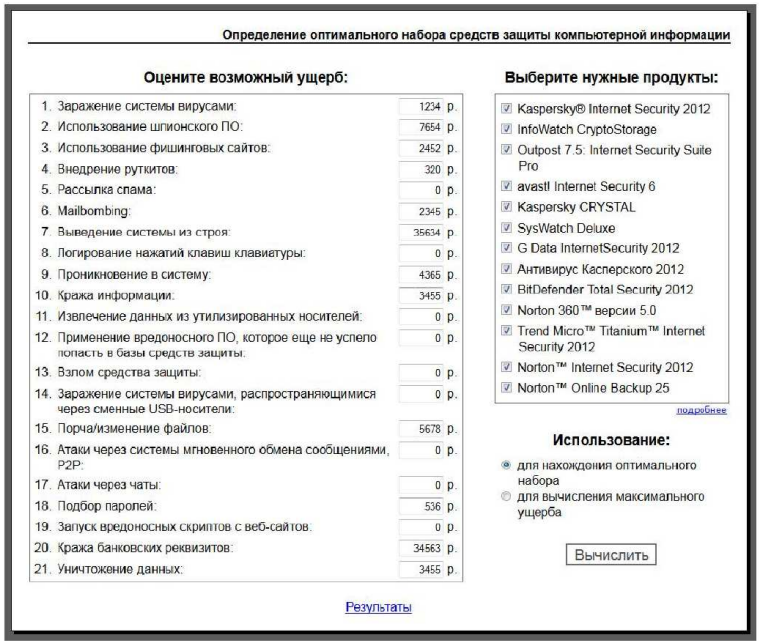


Рис. 4.1. Главная страница приложения

Справа на главной странице реализованного приложения расположен список используемых продуктов, обеспечивающих безопасность компьютерной информации. При каждом элементе он имеет поле типа «checkbox», что позволяет использовать в расчетах не все предложенные продукты, а только часть из них.

Под списком продуктов есть ссылка «подробнее», нажав на которую можно более детально ознакомиться с имеющимися продуктами.

Данная ссылка открывает новую страницу, где перечислены продукты, указана их цена и дано краткое описание (рис. 4.2). Название каждого продукта так же является ссылкой, ведущей на страницы соответствующих продуктов на официальных сайтах производителей.

Таблица 4.2

Соответствие средств защиты (стратегий y_j, столбцов) и возможных угроз (стратегий x_i, строк)

	1	2	3	4	5	6	7	8	9	10	11	12	13
1	+		+	+	+	+		+	+	+	+	+	
2	+		+	+	+		+	+	+	+	+	+	
3	+				+		+	+		+	+	+	
4	+			+	+			+					
5	+		+	+	+		+		+	+	+	+	
6				+			+			+		+	
7	+				+			+		+		+	
8								+					
9	+		+	+	+		+		+	+	+	+	
10		+			+						+		+
11		+			+						+		
12			+										
13			+			+							
14			+										
15			+			+					+		
16				+						+		+	
17				+			+			+		+	
18					+								
19	+		+	+	+	+	+			+	+	+	
20							+						
21					+				+	+	+		+

Определение оптимального набора средств защиты компьютерной информации методами теории игр

Продукты, обеспечивающие безопасность компьютерной информации

1. **Kaspersky® Internet Security 2012** (1600р.)
   Kaspersky Internet Security 2012 — решение для обеспечения оптимального уровня безопасности. Инновационная гибридная защита мгновенно устраняет вредоносные программы, спам и другие интернет-угрозы, экономя ресурсы компьютера за счет комбинации облачных и антивирусных технологий.
2. **InfoWatch CryptoStorage** (675р.)
   Надежный и простой способ защитить ваши данные от несанкционированного использования с помощью шифрования. Предназначен для небольших компаний и персонального использования.
3. **Outpost 7.5: Internet Security Suite Pro** (1299р.)
   Outpost Security Suite Pro — мощное, проактивное средство антивирусной защиты от широкого спектра существующих и будущих компьютерных угроз, таких как вирусы, шпионское ПО, руткиты и компьютеры-зомби.
4. **avast! Internet Security 6** (1000р.)
   Решение avast! Internet Security обеспечивает комплексную защиту от вирусов, шпионского ПО, спама и защиту при помощи брандмауэра и теперь дополнено новой технологией avast! SafeZone™. Эта технология создает изолированный виртуальный рабочий стол, невидимый для потенциального взломщика, на котором можно безопасно совершать покупки и банковские операции в Интернете.

Рис. 4.2. Подробное описание программных продуктов, обеспечивающих защиту информации (стратегии y_j, $j = 1, 2, ...$)

В связи с тем, что каждое из описанных выше средств обеспечения безопасности защищает сразу от нескольких угроз,

необходимо каждому средству защиты поставить в соответствие угрозы, от которых оно способно защитить. Анализ возможностей перечисленных в предыдущем разделе средств защиты позволяет каждому продукту поставить в соответствие определенные угрозы. Результат приведен в табл. 4.2. Столбцы представляют средства защиты, с номерами, соответствующими рис. 4.2, а строки представляют возможные угрозы с номерами из табл. слева на рис. 4.1.

Под списком средств защиты находится блок под названием «Использование» (см. рис. 4.1). Он определяет, как будут использованы выбранные выше средства защиты. Первый режим означает, что из выбранных продуктов будут составлены все возможные комбинации и из этих комбинаций будет выбрана оптимальная.

Второй режим, – что будет произведено вычисление максимального ущерба при использовании выбранных средств. Так же следует отметить, что при переключении на первый режим все поля типа «checkbox» заполняются, а на втором режиме очищаются. Это сделано для того, чтобы, при выборе своего набора средств на проверку не приходилось постоянно вручную очищать все ненужные поля. Особенно это критично, если для выбора доступно большое число средств защиты.

С левой стороны главной страницы расположен список возможных угроз (рис. 4.1). Здесь необходимо оценить в рублях, какой ущерб может быть нанесен при реализации той или иной угрозы. По умолчанию в данном программном продукте все величины ущерба равны нулю и администратору безопасности потребуется их заполнение на основе установленной системы обнаружения атак злоумышленников и их статистического анализа.

В самом низу главной страницы приложения находится кнопка «Вычислить» (см. рис. 4.1). По ее нажатии производятся необходимые вычисления и выводится результат (рис. 4.3). Если в текущей сессии уже проводились какие-то вычисления, то ниже блока «Использование» будет отображена ссылка «Результаты», пройдя по ней, можно ознакомиться с результатами

вычислений, полученными ранее. После нажатия на кнопку «Вычислить» или ссылку «Результаты» открывается блок с результатами вычислений (рис. 4.3).

Результат вычислений содержит таблицу со следующими полями:

1) номер подпункта;

2) набор из средств защиты (в зависимости от режима является либо оптимальным набором, либо набором средств, что был выбран на главной странице);

3) общая сумма стоимости средств защиты;

4) максимальный ущерб, который можно получить, используя данный набор средств защиты;

5) сумма стоимости средств защиты и максимального ущерба.


Определение оптимального набора средств защиты компьютерной информации

Результаты

№	Набор средств защиты	Стоимость	Максимальный ущерб	Сумма
1	2, 4, 8, 9	4674	3502	8176
2	2, 4, 8	3689	5734	9423
3	2, 3, 4, 8, 9	5973	3016	8989

Назад

Рис. 4.3. Результат вычислений приложения

Результат последнего вычисления добавляется в конец таблицы. При выборе из базы данных оптимального набора программных продуктов для защиты компьютерной информации предпочтение отдаётся более дешёвым аналогам. Таким образом, администратор безопасности может сначала получить оптимальный набор методов защиты, а потом изменять его, сверяясь с получающейся величиной максимального ущерба. При этом, если вычисление проводилось в режиме поиска оптимального набора средств защиты, то новая запись имеет белый фон. Если же производился подсчет максимального ущерба для выбранных средств защиты, то запись выделяется се-

рым фоном. Как видно из рис. 4.3, оптимальная стратегия всегда имеет общую сумму ущерба меньшую, чем у стратегий, составленных пользователем, даже если они выигрывают по стоимости средств защиты или величине ущерба.

Приложение позволяет распечатать как страницу с описанием используемых продуктов, обеспечивающих безопасность компьютерной информации, так и страницу с результатами вычислений. Для того чтобы данные имели подходящий для печати вид, используются альтернативные каскадные таблицы стилей. При отправке страницы на печать они автоматически преобразуют ее в нужную форму. Чтобы распечатать список средств защиты, нужно всего лишь, находясь на странице с их описанием, воспользоваться функцией печати, предлагаемой браузером. Результаты вычислений можно печатать как с главной страницы, так и со страницы с самими результатами.

4.5. Средства разработки и среда выполнения приложения

Основой для создания приложения стали идеи программирования веб-приложений. Веб-приложение может запускаться на любой системе, не требуя перекомпиляции, и для его запуска нужен лишь веб-браузер. На наш взгляд, хорошим выбором является создание веб-приложения, полностью выполняющегося на стороне клиента. Его интерфейс строится с помощью HTML и CSS, взаимодействие с пользователем и простые операции осуществляются языком JavaScript, а сложные ресурсоемкие вычисления доверяются Java-апплету.

Приложение должно запускаться в браузере. Но одного лишь браузера недостаточно для его корректной работы. Для работы Java-апплета на машине, с которой запускается приложение, должна стоять JVM (Виртуальная машина Java). HTML, CSS и JavaScript в настоящее время поддерживаются всеми популярными браузерами, но не всегда одинаково. Из-за проблем совместимости приложение, которое успешно

запускается в одном браузере, может некорректно работать в другом.

При разработке данного приложения использовались Java Runtime Environment версии 6 update 30 от компании Oracle и браузер Mozilla Firefox 9.0.1. Поэтому корректную работу приложения можно гарантировать только при использовании этого программного обеспечения.

Применение реализованного в данной работе программного продукта даст администратору безопасности возможность оценить эффективность используемого программного обеспечения и выбрать наиболее оптимальный набор средств защиты компьютерной информации.

Глава 5

Отражение атак в киберпространстве

Появление и развитие Интернета привело к тому, что многие государственные и частные организации, финансовые учреждения стали полагаться на него в повседневной деятельности.

Как результат, появилась возможность осуществлять враждебные действия против этих организаций, учреждений и государств через Интернет, которые могут быть взаимными, поэтому правомерным стало появление для компьютерного противостояния в пространстве Интернета, называемого *киберпространством*, такого термина, как «кибервойна».

Кибервойна – это использование одним государством киберпространства и связанных с ним технологических и информационных средств с целью причинения вреда военной, технологической, экономической, политической и информационной безопасности и суверенитету другого государства.

Поскольку исторически теория игр практически с самого момента своего появления стала применяться в военном деле, то естественно ее использовать для обороны информационных ресурсов, размещенных в киберпространстве.

5.1. Оборона в киберпространстве как матричная игра

Предположим, что в распоряжении группы B, отвечающей за компьютерную безопасность, находятся n серверов. Группа, в силу ограниченности ресурсов, способна обеспечивать эффективную безопасность только одного сервера. Пусть ценность сервера i равна c_i $(i = 1, ..., n)$ и все числа c_i различны. В таком случае математическое ожидание ущерба, наносимого нападающей стороной A в том случае, если сторона B защищает сервер j, а противник нападает на сервер i, равно [24, с. 83]

$$u_{ij} = \begin{cases} c_i(1-p), & i = j \\ c_i, & i \neq j \end{cases}, \tag{5.1}$$

где p – вероятность того, что сервер остается работоспособным и, следовательно, $(1-p)$ – это вероятность того, что защита сервера сломана.

Нападающая сторона стремится нанести максимальной ущерб u_{ij} обороняющейся стороне A, которая в свою очередь делает всё, чтобы минимизировать свои потери u_{ij}. Интересы сторон антагонистичны, и конфликт можно описывать как матричную игру с матрицей выигрышей

$$H = \begin{pmatrix} c_1(1-p) & c_1 & ... & c_1 \\ c_2 & c_2(1-p) & ... & c_2 \\ ... & ... & ... & ... \\ c_n & c_n & ... & c_n(1-p) \end{pmatrix}. \tag{5.2}$$

5.1.1. Случай $p = 1$: квалифицированная защита

Пусть $p = 1$. Это говорит о том, что группа B состоит из очень грамотных опытных специалистов по компьютерной защите.

Матрица игры имеет вид

$$H = \begin{pmatrix} 0 & c_1 & ... & c_1 & c_1 \\ c_2 & 0 & ... & c_2 & c_2 \\ ... & ... & ... & ... & ... \\ c_n & c_n & ... & c_n & 0 \end{pmatrix}. \tag{5.3}$$

Примем для удобства $c_1 > c_2 > ... > c_n$.

Вариант 1. Рассмотрим случай, когда для всех $3 \leq k \leq n$ верны неравенства

$$c_k > \frac{k-2}{\sum\limits_{i=1}^{k-1} \frac{1}{c_i}}. \tag{5.4}$$

Тогда смешанные стратегии $s^* = (p_1, ..., p_n), \sigma^* = (q_1, ..., q_n)$, где

$$p_i = \left(c_i \sum_{k=1}^{n} \frac{1}{c_k} \right)^{-1},$$

$$q_j = \frac{c_j \sum\limits_{k=1}^{n} \frac{1}{c_k} - (n-1)}{c_j \sum\limits_{k=1}^{n} \frac{1}{c_k}},$$

будут оптимальными для игроков A и B соответственно [24, с. 70] и

$$sH\sigma^T = \frac{n-1}{\sum\limits_{i=1}^{n} \frac{1}{c_i}}$$

является значением игры.

Вариант 2. Неравенства (5.4) для некоторых k не выполняются. Тогда существует минимальное число k_0, для которого неравенства (5.4) еще выполняются, но уже для $(k_0 + 1)$ не

выполняются, т. е.

$$c_k > \frac{k-2}{\sum\limits_{i=1}^{k-1} \frac{1}{c_i}}, \quad 3 \le k \le k_0, \tag{5.5}$$

$$c_{k_0+1} \le \frac{k_0 - 1}{\sum\limits_{i=1}^{k_0} \frac{1}{c_i}}. \tag{5.6}$$

Тогда смешанные стратегии $s^* = (p_1, ..., p_{k_0}, 0, ..., 0)$, $\sigma^* = (q_1, ..., q_{k_0}, 0, ..., 0)$, где

$$p_i = \left(c_i \sum_{k=1}^{k_0} \frac{1}{c_k} \right)^{-1} \text{ для } 1 \le i \le k_0,$$

$$q_j = \frac{c_j \sum\limits_{k=1}^{k_0} \frac{1}{c_k} - (k_0 - 1)}{c_j \sum\limits_{k=1}^{k_0} \frac{1}{c_k}} \text{ для } 1 \le j \le k_0$$

будут оптимальными для игроков A и B соответственно [24, с. 72] и

$$sH\sigma^T = \frac{k_0 - 1}{\sum\limits_{i=1}^{k_0} \frac{1}{c_i}}$$

является значением игры.

5.1.2. Случай $p < 1$: сильный противник

Пусть $p < 1$. Иначе говоря, обороняющаяся сторона отдает себе отчет в том, что противник очень сильный и среди его специалистов имеются изощренные в средствах взлома серверов энтузиасты.

Ищем смешанные стратегии, которые пренебрегают нападением и соответственно защитой менее ценных серверов. Иначе говоря, ищем смешанные стратегии вида

$$s^* = (p_1, ..., p_{k_0}, 0, ..., 0), \quad \sigma^* = (q_1, ..., q_{k_0}, 0, ..., 0),$$

Примем, что

$$\frac{c_1 - c_2}{c_1} < p.$$

В этом случае седловая точка матрицы H – элемент $h_{11} = c_1(1-p)$ и соответствующие чистые стратегии игроков A и B состоят в том, что при нападении на самый ценный объект 1 игрок B его защищает.

Пусть k_0 – наименьший индекс, для которого неравенство

$$c_{k_0} > \frac{k_0 - p - 1}{\sum\limits_{i=1}^{k_0-1} \frac{1}{c_i}} \tag{5.7}$$

выполняется, а при $(k_0 + 1)$ нарушается:

$$c_{k_0+1} \leq \frac{k_0 - p}{\sum\limits_{i=1}^{k_0} \frac{1}{c_i}}. \tag{5.8}$$

В этом случае оптимальные смешанные стратегии имеют вид

$$s^* = (p_1, ..., p_{k_0}, 0, ..., 0), \quad \sigma^* = (q_1, ..., q_{k_0}, 0, ..., 0),$$

где

$$p_i = \left(\sum_{k=1}^{k_0} \frac{1}{c_k} \right)^{-1} \text{ для } 1 \leq i \leq k_0,$$

$$q_j = \frac{c_j - v}{p c_j}, \text{ для } 1 \leq i \leq k_0,$$

$$v = \frac{k_0 - p}{\sum\limits_{i=1}^{k_0} \frac{1}{c_i}}.$$

Число v является значением игры [24, с.85].

5.2. Защита машин компьютерной сети как игра с ненулевой суммой

Рассмотрим предложенную в [64] теоретико-игровую модель защиты серверов компьютерной сети в случае, когда игра не является матричной и, следовательно, функции выигрышей группы защитников и атакующей группы задаются отдельно.

Примем, что противник применяет только один тип атаки. Например, эксплуатирует известную уязвимость в устаревшей версии веб-сервера. Эта специфическая атака не всегда будет применима к каждой машине в сети (например, если атака успешна против Windows-машины, а атака производится на Linux-машину).

Обороняющаяся сторона – игрок $З$, конечно, обладает знанием, какие машины в сети являются уязвимыми, а атакующий – игрок $А$ – нет.

Игра происходит в следующем порядке:

1. Волей случая какие-то машины в сети являются уязвимыми для атак.

2. Игрок $З$ выделяет ресурсы на защиту каждой машины на основе свой стратегии.

3. Нападающий противник $А$ атакует машины в сети, направляя свои усилия на взлом машин в соответствии с его стратегией.

4. Если нападающему $А$ удастся взломать и нанести ущерб, по крайней мере одной машине, он получает еще один шанс перераспределить свои ресурсы и напасть на остальные машины.

5. Вычисляются функции выигрыша для обеих сторон (игроков).

Стратегии защиты – это то, как распределяются имеющиеся в распоряжении обороняющейся стороны средства для того, чтобы поставить защиту на тот или иной сервер. Стратегии атакующего сводятся к тому, в каком порядке он будет атаковать машины сети.

5.2.1. Стратегии

Будем считать, что в сети n серверов, каждый из них имеет свою ценность, которая известна нападающей стороне A. Но сторона A не знает степень защищенности каждого сервера.

Ценность сервера может относиться к чему угодно. Она включает и денежную стоимости машины, и время, необходимое для ее ремонта, и стоимость замены машины, и возможный ущерб, нанесенный успешным нападением, включая потерю военного (конкурентного) преимущества и разглашение персональных данных.

Ценность сервера будем характеризовать целым числом от 1 до 100.

Стратегии атаки. Предполагаем, что атакующий – игрок A – использует только две стратегии, называемые HIGH и PROPORTIONAL.

Игрок A атакует сразу $m \in \{1, ..., n\}$ $(m < n)$ серверов.

Выбор атакующим A стратегии атаки HIGH означает, что все свои усилия он направляет на машину с наивысшей ценностью (из m машин). Если атакующий взломал $k > 0$ машин, то он будет использовать эту же стратегию для взлома оставшихся $n - k$ машин на следующем шаге.

При выборе стратегии PROPORTIONAL атакующий распределяет свои усилия, уделяя часть их на взлом конкретной машины пропорционально ее значимости (ценности) в сети. Например, если данная машина представляет собой 30% стоимости всех компьютеров, входящих сеть, то эта стратегия предполагает, что атакующий будет атаковать эту машину с 30%-й долей всех его усилий.

Стратегии защиты. Защищающаяся сторона *З* использует для защиты три стратегии: HIGH, PROPORTIONAL и NOTHING.

Считаем, что сторона З не имеет возможности перераспределить свои ресурсы по защите машин сети до того, как атакующий начинает (возможно) повторную атаку. Это – естественное допущение. Действительно, во-первых, сторона *З* не знает, как нападающий будет атаковать, а во-вторых, его атаки будут идти по времени так близко друг к другу, что перераспределение средств защиты просто невозможно.

Стратегия HIGH заставляет сторону *З* сосредоточиться на защите машины с высокой стоимостью. Эта стратегия часто используется в реальных ситуациях, поскольку администраторы верят, что злоумышленник предпочитает нападать на самую ценную машину, т. е., что он использует при атаке стратегию HIGH. Однако будем считать, что защищающаяся сторона отдает себе отчет, что не в полной мере может обеспечить защиту самой ценной машины. Она оценивается с некоторым критическим уровнем $e' \in [0, 1]$. Говорим, что стратегия «HIGH e'» – это стратегия HIGH, которая гарантирует защиту самой ценной машины с уровнем e'. Например, стратегия HIGH 0,5 означает стратегию HIGH с уровнем $0,5$.

При выборе стратегии PROPORTIONAL администратор распределяет средства защиты пропорционально ценности машин. Причем стратегия «PROPORTIONAL e'» – это стратегия PROPORTIONAL, в которой в случае, когда ценность машины i в сети равна $s_i\%$, то она защищена с уровнем гарантии $e' \cdot s_i\%$.

Стратегия NOTHING такова, что игрок *З* вообще не озабочен защитой сети. С этой стратегией он экономит средства на защиту, правда, терпит убытки.

Можно ввести в рассмотрение стратегию LOWHIGH, согласно которой игрок *З* защищает самую неценную машину, предполагая, что именно ее будет атаковать хитрый злоумышленник, следуя своей подобной же стратегии LOWHIGH.

5.2.2. Функции выигрыша

Функция выигрыша для стороны $З$ – это потери от атаки минус стоимость защиты каждой машины. Нападающая сторона удовлетворяется нанесенным ущербом. Следовательно, функции выигрыша имеют вид:

$$\pi_З = -l - \sum_{i=1}^{n} c(e_i), \tag{5.9}$$

$$\pi_A = l, \tag{5.10}$$

где l – процент потерь, понесенных владельцем компьютерной сети (или, что эквивалентно, нанесенных атакующим), n – количество машин в сети; $c : [0,1) \to [0,+\infty)$ – стоимостная функция; e_i – минимальное усилие, необходимое, чтобы проникнуть на i-ю машину в сети.

Таким образом, например величина $c(0,5)$ – это стоимость средств на защиту машины, которую игрок $З$ должен должен заплатить для обеспечения успешной обороны против атак на нее при условии, что не более 50% усилий злоумышленника направлено для проникновения на рассматриваемую машину.

Функция выигрыша атакующего строится на предположении, что он не несет существенных технических затрат[1], связанных с атакой. Атакующий игрок A производит либо одну атаку, либо только несколько отдельных атак. Он ограничен в объеме предпринимаемых усилий во время атаки. Более того, времени на атаки у злоумышленника крайне мало, поскольку он не желает быть обнаруженным и идентифицированным. Поэтому мы и считаем, что его выигрыш – это проигрыш обороняющейся стороны, иначе говоря, определяем его функцию выигрыша пропорциональной сумме ущерба, который он может причинить.

[1]Если это хакер-энтузиаст. Если же атаку организует государство, то оно, конечно, несет расходы, оплачивая труд и технику своих бойцов киберфронта.

А вот потери владельца сети могут быть существенными. При этом, как это часто бывает, владелец всегда ограничен в средствах, которые он может выделить на защиту своей сети.

Поскольку нарастание усилий $e \in [0,1]$ на взлом машины со стороны нападающего требует для успешной защиты вложения новых денежных средств (т. е. функция $c(e)$ – возрастающая), которые у владельца ограничены, то при некотором значении e они должны рассматриваться владельцем, как умопомрачительные. Поэтому при задании функции $c(e)$ следует данную ситуацию смоделировать как наличие значения $e = a$, при котором $c(e)$ имеет вертикальную асимптоту (рис. 5.1).

В силу сказанного берем

$$c(e) = \begin{cases} f(e), & f(e) \geq 0, \\ \infty, & f(e) < 0, \end{cases}, \tag{5.11}$$

где

$$f(e) = -\frac{se^2}{ae-1}, \quad s > 0,\ a \geq 1.$$

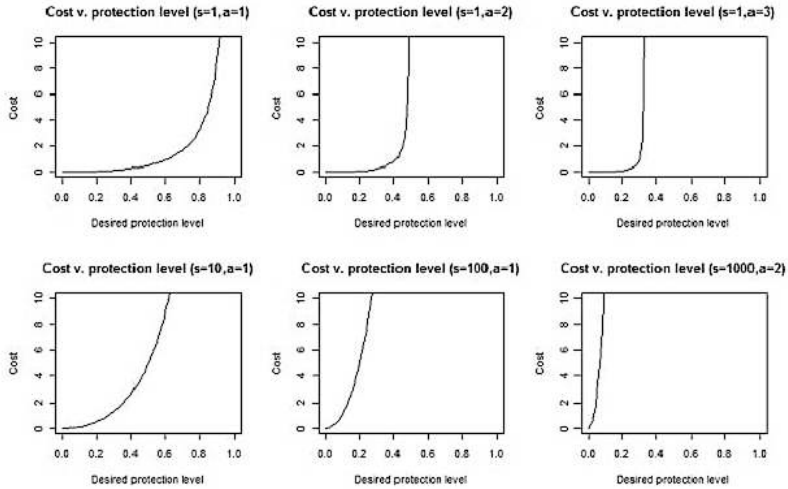


Рис. 5.1. Примеры стоимостной функции

5.2.3. Компьютерное моделирование

Проведем компьютерное моделирование по взлому сети, состоящей из $n = 100$ машин.

Шаг 1. Фиксируем $a \in [1, 10]$ и $s \in \{0, 1, ..., 0.9\}$.

Шаг 2. Сторона A выбирает стратегию атаки σ_A.

Шаг 3. Сторона $З$ выбирает стратегию защиты $s_З$.

Шаг 4. Сторона $З$ устанавливает уровни защиты e'_i, $i = 1, ..., 100$ для машин.

Шаг 5. Сторона A атакует случайно взятые три машины с усилиями, соответствующими выбранной стратегии.

Машина i успешно взломана, если усилия $e_i \in [0, 1]$ атакующего машину i больше, чем уровень e'_i гарантированной защиты этой машины, установленный стороной $З$ в выбранной стратегии $s_З$. Если взлом был успешным, атакуем остальные 97 машин.

Вычисляем функции выигрыша по формулам (5.9), (5.10). При моделировании примем, что

$$l = \text{сумма ценности взломанных машин.}$$

Поскольку надо смоделировать ситуацию, когда могут быть атакованы любые три машины, повторяем атаку на три машины, но уже другие и делаем такие повторные атаки для набора статистики 50 000 раз. Итак,

Шаг 6. Идем на шаг 5 и повторяем это 50 000 раз.

Шаг 7. Вычисляем средние значения

$$avg(\pi_З), \quad avg(\pi_A)$$

функций выигрыша игроков, помещаем их в клетки матрицы рис. 5.2, задавая тем самым биматричную игру $\Gamma(a, s)$.

Шаг 8. Находим для полученной биматричной игры $\Gamma(a, s)$ равновесие Нэша, которое отмечаем как точку на прямоугольнике $[1, 10] \times [0, 1; 0, 9]$ в осях $a - s$, помня, что это за оптимальная стратегия.

Шаг 9. На прямоугольнике $[1, 10] \times [0, 1; 0, 9]$ выделяем множества, состоящие из одинаковых пар стратегий защитника.

Шаг 10. Идем на шаг 1 и делаем такие возвраты на шаг 1, пока не переберем все пары (a, s).

5.2.4. Алгоритм программы моделирования

Программа компьютерных экспериментов пишется на основе следующего алгоритма:

for $s \in \{0.1, 0.2, ..., 1\}$ **do**
  for $a \in \{1, 2, ..., 10\}$ **do**
    for $\sigma_A \in \{\text{PROPORTIONAL}, \text{HIGH}\}$ **do**
      for $s_{\mathcal{З}} \in \{\text{NOTHING}, \text{PROPORTIONAL}, \text{HIGH}\}$ **do**
        for $e'_i \in \{0.1, 0.2, ..., 0.9\}$ **do**
          $\pi_A := 0$
          $\pi_{\mathcal{З}} := 0$
          loop 50,000 times
            machineValues := случайные 3 числа от 1 до 100
            игрок $\mathcal{З}$ защищает машины согласно $s_{\mathcal{З}}$
            игрок A нападает согласно σ_A
            if игрок A взломал сервер **then**
            игрок A атакуют остальные сервер(ы)
            end if
            $\pi_A := \pi_A +$ нанесенные потери на этом шагу
            $\pi_{\mathcal{З}} := \pi_{\mathcal{З}} -$ нанесенные потери на этом шагу
          end loop
% создание биматричной (нормальной формы) игры:
          Record $avg(\pi_{\mathcal{З}}), avg(\pi_A)$ в клетку $(s_{\mathcal{З}}\ e'_i, \sigma_A)$
        end for
      end for
    end for
    Находим равновесия Нэша для созданной биматричной игры
    с матрицей (19×2) и отметим его как точку на $a - s$ графике.
  end for
end for

Следуя этому алгоритмы, нетрудно написать программу для компьютерных экспериментов. Она должна обращаться

к известным программам вычисления равновесий Нэша для биматричных игр.

Стратегии обороняющейся стороны	Стратегии нападающей стороны: HIGH	Стратегии нападающей стороны: PROPORTIONAL
NOTHING	-70 , 70	-70 , 70
HIGH 0.1	-63 , 63	-70 , 70
HIGH 0.2	-63 , 63	-70 , 69
HIGH 0.3	-65 , 62	-71 , 69
HIGH 0.4	-70 , 63	-70 , 63
HIGH 0.5	OUT OF RANGE	OUT OF RANGE
HIGH 0.6	OUT OF RANGE	OUT OF RANGE
HIGH 0.7	OUT OF RANGE	OUT OF RANGE
HIGH 0.8	OUT OF RANGE	OUT OF RANGE
HIGH 0.9	OUT OF RANGE	OUT OF RANGE
PROPORTIONAL 0.1	-51 , 50	-70 , 70
PROPORTIONAL 0.2	-51 , 51	-69 , 69
PROPORTIONAL 0.3	-52 , 51	-70 , 69
PROPORTIONAL 0.4	-53 , 51	-70 , 67
PROPORTIONAL 0.5	OUT OF RANGE	OUT OF RANGE
PROPORTIONAL 0.6	OUT OF RANGE	OUT OF RANGE
PROPORTIONAL 0.7	OUT OF RANGE	OUT OF RANGE
PROPORTIONAL 0.8	OUT OF RANGE	OUT OF RANGE
PROPORTIONAL 0.9	OUT OF RANGE	OUT OF RANGE

Рис. 5.2. Матрица игры для $a = 2$; $s = 0, 1$. Равновесия Нэша – ячейка $[-69, -69]$, и лучшие ответы подчеркнуты. Комбинации, которые невозможны из-за неопределенности функции стоимости защищающейся стороны, обозначены словами OUT OF RANGE (вне области задания)

5.2.5. Результаты экспериментов

Биматричная игра $\Gamma(2, 0.1)$ приведена на рис. 5.2. Оптимальные чистые стратегии – чистое равновесия Нэша – таковы:

$$s_A = PROPORTIONAL, \qquad \pi_A = 69,$$

$$\sigma_З = PROPORTIONAL\ 0.2, \quad \pi_З = -69.$$

Как видим, оптимальна стратегия защиты, при которой в случае ценности $s_i\%$ машины i, обеспечивается ее защита с уровнем гарантии $0,2 \cdot s_i\%$. Для атакующего оптимальна стратегия, при которой он распределяет свои усилия, уделяя часть их взлому конкретной машины пропорционально ее значимости (ценности) в сети.

Равновесия Нэша по всем играм $\Gamma(s, s)$ даны на рис. 5.3. Атакующий использует чистые оптимальные стратегии PROPORTIONAL, а чистые оптимальные PROPORTIONAL для обороняющей стороны занимают 10% площади прямоугольника $[1, 10] \times [0, 1; 0, 9]$.

Данные рис. 5.3 показывают, как влияет конфигурация (a, s), т. е. задание функции выигрыша $s(e)$, на выбор оптимальных стратегий.

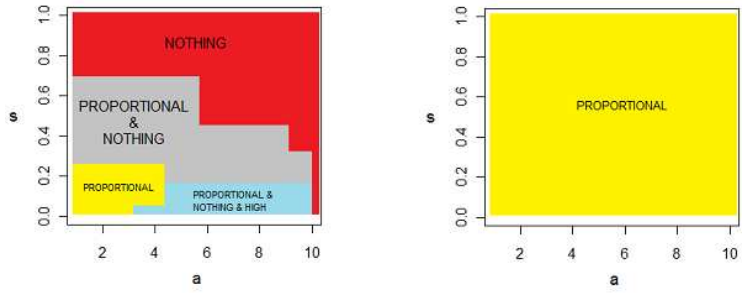


Рис. 5.3. Распределение равновесий игр $\Gamma(a, s)$ для игрока $З$ (слева) и атакующего $А$ (справа)

5.3. Н.Н. Воробьев

Николай Николаевич Воробьев (1925–1995) – советский выдающийся математик – основатель и глава советской (российской) теоретико-игровой школы.

Благодаря ему советские математики сумели за короткое время включиться в развитие математических основ теории

игр буквально с самого начала ее зарождения.

Н.Н. Воробьев

Н.Н. Воробьев учился в Ленинградском кораблестроительном институте, затем закончил математико-механический факультет Ленинградского университета (1948) по специальности «Алгебра». Обучался в аспирантуре Ленинградского отделения математического института (ЛОМИ) АН СССР им. В.А. Стеклова, где изучал конструктивную математическую логику под руководством А.А. Маркова. В 1952 г. защитил кандидатскую диссертацию по логическим правилам дедукции в системах с сильным отрицанием. С 1952 г. – научный сотрудник ЛОМИ АН СССР.

В 1950-е годы стала развиваться теория игр – наука, описывающая и исследующая математические модели конфликтных ситуаций. Н.Н. Воробьев увлекся этой наукой, и его первой работой по теории игр была статья «Управляемые процессы и теория игр», опубликованная в 1955 г.

С этого времени теория игр становится главным научным интересом Н.Н. Воробьева. В 1960-е гг. им построен алгоритм решения биматричных игр (в дальнейшем получивший название алгоритма Воробьева–Куна), получено несколько результатов об эквивалентности рандомизированных стратегий в позиционных играх, о существовании ситуаций равновесия в играх с запрещенными ситуациями и др.

Большое значение он придавал пропаганде теории игр в СССР. Под его редакцией были осуществлены перевод на русский язык и издание ряда важнейших монографий по теории игр, включая основополагающую «Теория игр и экономическое поведение» Дж. фон Неймана и О. Моргенштерна.

Глава 6

Выбор средства эффективной защиты от DoS/DDoS-атак

DDoS-атаки являются распространенным способом нанесения ущерба компьютерным системам. Они бывают двух типов: истощение ресурсов сети и истощение ресурсов хоста. Атаки осуществляются с помощью непосредственной посылки жертве большого количества пакетов (как, например, UDP и ICMP flood) или посредством использования для этой цели промежуточных узлов (примеры: Smurf и Fraggle), а также посредством передачи слишком длинных пакетов (Ping Of Death), некорректных пакетов (Land) или большого количества трудоемких запросов (TCP SYN) и т. д.

Статистика распределения типов DDoS-атак такова: 31 % – Smurf-ping; 10% – ICMP flood; 14% – UDP flood; 11% – TCP flood; 31% – TCP SYN flood; 3 % – прочие [53, 66].

Успешная DDoS-атака – это катастрофа для информационного ресурса, означающая для его владельца как финансовые потери, так и моральные. DDoS-атака является катастрофой и

в математическом смысле, поскольку допускает описание как катастрофа «сборки» в рамках математической теории катастроф [4, 22].

Актуальной является задача построения эффективной системы защиты от DDoS-атак. Решение этой задачи должно включать механизмы предупреждения атаки, обнаружения факта атаки, определения источника атаки и противодействия ей.

6.1. *DDoS*-атаки на компьютерные системы

DDoS-атаки – наиболее распространенная атака злоумышленников на компьютерный информационный ресурс. Существуют два способа добиться от сервера отказа в обслуживании (*Denial of Service*).

Первый способ позволяет остановить работу *всей* атакуемой компьютерной системы. Для этого злоумышленник посылает серверу-жертве данные или пакеты, которые она не ожидает, и это приводит либо к остановке системы, либо к ее перезагрузке. В результате никто не сможет получить доступ к ресурсам. Атака хороша тем, что с помощью нескольких пакетов можно сделать систему неработоспособной.

Второй способ (*Flood*-атаки) состоит в том, чтобы добиться переполнения системы с помощью такого большого количества пакетов, которое невозможно обработать. Например, если система может обрабатывать только 10 пакетов в секунду, а злоумышленник отправляет к ней 20 пакетов в секунду, то остальные пользователи при попытке подключиться к системе получают отказ в обслуживании, поскольку все ресурсы заняты. При таких атаках значительно снижается производительность компьютерной системы или приложений. Очевидно, что при этом способе атаки наблюдается резкое возрастание входящего трафика.

Есть и третий способ атаки, при которой стараются до-

биться переполнения канала, т.е. резко снизить пропускную способность канала.

6.2. Выбор средства эффективной защиты от $DoS/DDoS$-атак

Продемонстрируем, как работают методы теории игр при выборе эффективного средства защиты от DoS/DDoS-атак (Абденов, Заркумова [1, 2, 27]).

Допустим, имеются информационные ресурсы (ИР), которые подвергаются DoS/DDoS-атаке (Denial of Service – отказ в обслуживании и Distributed Denial of Service – распределенный отказ в обслуживании).

Будем рассматривать лишь часто используемые средства защиты (в силу их доступности и распространенности) и их комбинации: (1) – фаейрвол (со стандартными настройками); (2) – средства обнаружения вторжения; (3) – резервирование канала связи;(1+2) – фаейрвол и средства обнаружения вторжения; (1+3) – фаейрвол и резервирование канала связи; (1+2+3) – фаейрвол, средства обнаружения вторжения и резервирование канала связи.

Рассматриваем игру

.	S_1	S_2	...	S_j
A_1	w_{11}	w_{12}	...	w_{1j}
A_2	w_{21}	w_{22}	...	w_{2j}
...	...	...	...	...
A_i	w_{i1}	w_{i2}	...	w_{ij}

Опираясь на критерий Вальда

$$W = \max_i \min_j w_{ij}$$

и критерий Гурвица

$$W = \max_i [\alpha \min_j w_{ij} + (1 - \alpha) \max_j w_{ij}],$$

требуется определить средство эффективной защиты от упомянутых типов атак.

6.3. Методика решения

Пусть известны следующие показатели: $A_1, A_2, ..., A_i, ..., A_m$ – виды атак; m – количество видов атак; $S_1, S_2, ..., S_j, ..., S_n$ – средства защиты; n – количество средств защиты; X_j $j = 1, ..., n$ – стоимость применяемого средства защиты; Y – величина предполагаемого ущерба; w_{ij} или $p_{ij}^{(\text{З})}$ $i = 1, ..., m$, $j = 1, ..., n$ – вероятность отражения атаки A_i при использовании средства защиты S_j , т. е. вероятность защиты; $p_{ij}^{(a)}$ – вероятность проведения i-й атаки; $p_{ij}^{(\text{У})}$ – вероятность нанесения ущерба при i-й атаке и j-м средстве защиты с учетом частоты использования i-й атаки.

Условием эффективной защиты является следующее правило: стоимость средств защиты должна быть меньше стоимости потерь, понесенных при успешной реализации атаки.

Поскольку вероятность применения одного из средств защиты равна 1 (100 %), составим следующее неравенство:

$$1 \cdot X_j \le p_{ij}^{(\text{У})} \cdot Y, \quad i = \overline{1, m}, \quad j = \overline{1, n}. \tag{6.1}$$

Представим вероятность нанесения ущерба $p_{ij}^{(\text{У})}$ через вероятность защиты от атаки и вероятность проведения атаки

$$p_{ij}^{(\text{У})} = (1 - p_{ij}^{(\text{З})}) \cdot p_{ij}^{(\text{а})}, \quad i = \overline{1, m}, \quad j = \overline{1, n}. \tag{6.2}$$

Подставив (6.2) в (6.1), получим

$$X_j \le (1 - p_{ij}^{(\text{З})}) \cdot p_{ij}^{(\text{а})} \cdot Y, \quad i = \overline{1, m}, \quad j = \overline{1, n}. \tag{6.3}$$

Разделив обе части соотношения (6.3) на выражение, стоящее в правой части этого неравенства, получим

$$\frac{X_j}{(1 - p_{ij}^{(\text{З})}) \cdot p_{ij}^{(\text{а})} \cdot Y} \le 1, \quad i = \overline{1, m}, \quad j = \overline{1, n}. \tag{6.4}$$

Обозначим в (6.4) левую часть неравенства через λ_{ij} и назовем коэффициентом эффективной защиты

$$\lambda_{ij} = \frac{X_j}{(1 - p_{ij}^{(\text{з})}) \cdot p_{ij}^{(\text{а})} \cdot Y}, \quad i = \overline{1,m}, \quad j = \overline{1,n}. \tag{6.5}$$

Согласно (6.4) условием эффективной защиты будет соотношение

$$\lambda_{ij} \leq 1, \quad i = \overline{1,m}, \quad j = \overline{1,n}. \tag{6.6}$$

Это условие будем использовать с критерием Вальда. Правило выбора решения в соответствии с максиминным критерием Вальда: платёжная матрица дополняется строкой, каждый элемент которой есть минимальное значение выигрыша в соответствующей стратегии лица принимающего решение, т. е. $W_{ij} = \min\ w_{ij}$, $i = \overline{1,m}$, $j = \overline{1,n}$. Оптимальной по данному критерию считается та стратегия лица, принимающего решение, при выборе которой минимальное значение выигрыша максимально: $W = \max\ W_{ij}$, $i = \overline{1,m}$. Выбранная таким образом стратегия полностью исключает риск. Это означает, что принимающий решение не может столкнуться с худшим результатом, чем тот, на который он ориентируется.

6.4. Численные эксперименты

Допустим, что известны вероятности отражения атак различными средствами защиты (в дальнейших расчетах будем использовать только первые пять типов атак) (табл. 6.1) и данные о стоимости средств защиты(табл. 6.2).

Далее рассчитаем коэффициенты эффективной защиты при величине предполагаемого ущерба от реализации атак, равный 50 у.е. Применим для матрицы коэффициентов эффективной защиты (табл. 6.3) стратегию Вальда, также будем учитывать условие (6.6).

Таким образом, при стоимости ресурсов, равной 50 у. е., оптимальным средством защиты является фаейрвол.

Таблица 6.1

Матрица вероятностей отражения атаки [27]

Атаки \ Защиты	Вероятность использования злоумышленниками различных видов атак	Средства защиты						
		Отсутствие средств защиты (0)	Файрвол (1)	Средства обнаружения вторжения (2)	Резервирование канала связи (3)	(1)+(2)	(1)+(3)	(1)+(2)+(3)
		Вероятности отражения атаки						
Smurf-атака	0,310	0,000	0,700	0,900	0,800	0,920	0,900	0,980
ICMP flood	0,100	0,000	0,800	0,950	0,830	0,970	0,870	0,995
UDP flood	0,140	0,000	0,800	0,950	0,800	0,960	0,850	0,980
TCP flood	0,110	0,000	0,800	0,990	0,800	0,995	0,900	0,999
TCP SYN flood	0,310	0,000	0,600	0,930	0,750	0,950	0,800	0,980

Таблица 6.2

Стоимость средств защиты в у. е. (при соблюдении пропорций) [27]

Атаки\ Защиты	Отсутствие средств защиты (0)	Файрвол (1)	Средства обнаружения вторжения (2)	Резервирование канала связи (3)	(1)+(2)	(1)+(3)	(1)+(2)+(3)
Стоимость средства защиты	0	1	50	20	51	21	71

Таблица 6.3

Матрица коэффициента эффективной защиты при величине предполагаемого ущерба Y = 5 у. е. с выбором средства эффективной защиты [27]

Атаки \ Защиты	Вероятность использования злоумышленниками различных видов атак	Средства защиты						
		Отсутствие средств защиты (0)	Файрвол (1)	Средства обнаружения вторжения (2)	Резервирование канала связи (3)	(1)+(2)	(1)+(3)	(1)+(2)+(3)
		Коэффициенты эффективной защиты						
Smurf-атака	0,310	0,0000	0,2151	32,2581	6,4516	41,1290	13,5484	229,0323
ICMP flood	0,100	0,0000	1,0000	200,0000	23,5294	340,0000	32,3077	2840,0000
UDP flood	0,140	0,0000	0,7143	142,8571	14,2857	182,1429	20,0000	507,1429
TCP flood	0,110	0,0000	0,9091	909,0909	18,1818	1854,5454	38,1818	12909,0900
TCP SYN flood	0,310	0,0000	0,1613	46,0829	5,1613	65,8065	6,7742	229,0323
Минимум по столбцу			0,1613	32,2581	5,1613	41,1290	6,7742	229,0323
Максимум по строке <1			0,1613					

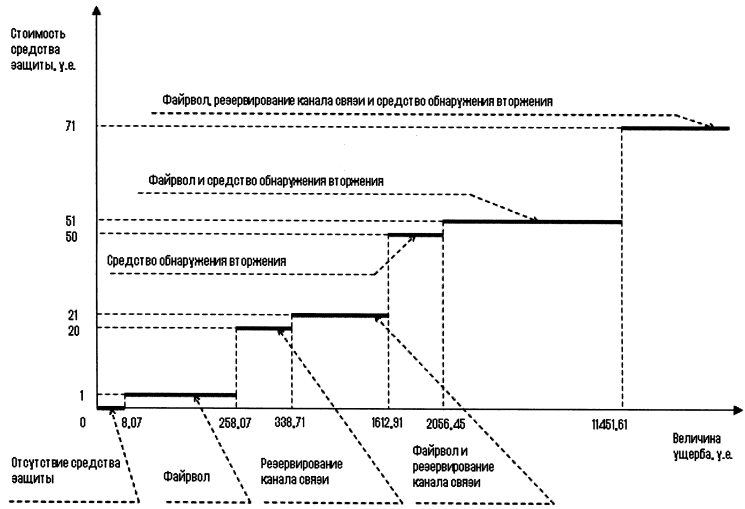


Рис. 6.1. График зависимости стоимости средств защиты от величины предполагаемого ущерба [27]

Таблица 6.4

Стоимостные коридоры в у. е. [27]

	Отсутствие средств защиты (0)	Файрвол (1)	Резервирование канала связи (3)	(1)+(3)	Средства обнаружения вторжения (2)	(1)+(2)	(1)+(2)+(3)
0 (стратегия крайнего оптимизма)	0	50,00	1176,47	1909,09	45454,55	92727,27	645454,50
0,2	0	41,61	992,79	1695,01	36686,22	74593,11	518653,90
0,4	0	33,23	809,11	1280,94	27917,89	56458,94	391853,34
0,5 (пессимистично-оптимистическая стратегия)	0	29,03	717,27	1123,90	23533,73	47391,86	328453,05
0,6	0	24,84	625,43	966,86	19149,56	38324,78	265052,76
0,8	0	16,45	441,75	652,79	10381,23	20190,61	138252,19
1 (стратегия Вальда, стратегия крайнего пессимизма)	0	8,07	258,07	338,71	1612,91	2056,45	11451,62

В соответствии с критерием Гурвица рассчитаем все пороговые значения Y при $\alpha = 0,5$ и $0 \leq \alpha \leq 1$, при шаге $\alpha = 0,2$. В результате получим интервалы значений Y, в которых применение конкретного средства защиты будет эффективным. Полученные значения представлены в табл. 6.4.

Таким образом, придерживаясь пессимистично-оптимистической стратегии и предполагая ущерб, равный 5000 у.е., можем использовать фаейрвол и средства обнаружения вторжения. График зависимости при использовании критерия Вальда приведен на рис. 6.1.

6.5. *DDoS*-атака как катастрофа «сборки»

Покажем, что второй способ *DDoS*-атаки, описанный в § 6.1, допускает математическое описание в рамках математической теории катастроф.

Мы видим, что для рассматриваемого способа *DDoS*-атаки, во-первых, важную роль играет *входящий трафик*. Трафик – это параметр τ, характеризующий типичную ситуацию для функционирующей компьютерной системы, которая говорит, что, как правило, ежедневный трафик именно таков и система способна с ним справляться с определенным запасом её надежности.

Увеличение трафика требует для его обработки увеличения свободных ресурсов системы.

Во-вторых, мы видим, что важным параметром стойкости, надежности компьютерной системы является ее *производительность* p, выражающаяся как в скорости обработки входящих пакетов, так и количестве устанавливаемых соединений.

При получении сервером пакета данных происходит его обработка. Это требует времени и определенных ресурсов компьютерной системы. Если приходит новый пакет, а сервер занят приемом или обработкой предыдущего или другого паке-

та, то вновь приходящий запрос-пакет *ставится в очередь*, занимая при этом часть ресурсов системы.

При *Flood*-атаках происходит исчерпание ресурсов, а точнее ресурсов процессора, памяти или каналов связи, сводящиеся к следующим моментам:

• Ограниченное количество соединений, находящихся в состоянии установки (соединения), которыми располагает система (при TCP SYN *Flood*- и TCP *Flood*-атаках направляется большое количество запросов на инициализацию TCP-соединения с потенциальной системой-жертвой). Добиваются того, что система не может устанавливать новые соединения.

• Способность системы отвечать на посылаемые ping-запросы (*ICMP Flood*-атаки, *Smurf*-атаки), на которые система должна отвечать автоматически. Если запрос использует большие (64 кБ), сильно фрагментированные ICMP-пакеты, то при получении таких пакетов атакуемая система зависает.

• Снижение пропускной способности канала связи за счет потока большого количества UDP-пакетов разного размера (UDP *Flood*-атаки). Происходит перегрузка канала связи, и сервер, работающий по протоколу TCP, перестаёт отвечать.

Таким образом, способность к нормальному функционированию определяется числом откликов на запросы.

Обозначим через $x(t)$ число откликов на запросы в момент времени t.

Тогда

$$x(t+1) = x(t) + f[x(t)] + \tau, \tag{6.7}$$

где $f[x(t)]$ – результат работы системы по обработке запросов на момент t. В уравнении отражено требование, что больший трафик требует нарастания числа откликов на запросы.

Примем для простоты, что $f[x(t)] = kx(t)$, где k – величина, определяющая производительность системы

$$k = \{p - g[x(t)]\}, \tag{6.8}$$

сводящаяся к средней скорости обработки входящих пакетов p с учетом ее падения или увеличения в зависимости от объ-

ема занятых ресурсов: чем больше загружены ресурсы, тем меньше скорость обработки входящих пакетов.

Пакет x, стоящий в очереди, должен пройти через соединение (либо просто пройти по забитому каналу, как UDP-пакет) и после обработки, возможно, породить отклик для пославшего его компьютеру. Иначе говоря, пакет участвует в процессе его обработки как минимум дважды . Поэтому мы это отразим путем принятия предположения, что $g[x] = x^2$.

Таким образом, $g[x(t)] = [x(t)]^2$, и тогда

$$x(t+1) = x(t) + [(p - p_0) - x^2(t)]x(t) + (\tau - \tau_0), \qquad (6.9)$$

где введены некоторые «типичные», характерные для данного сервера величины производительности p_0 и трафика τ_0. При переходе к непрерывному времени уравнение (6.9) сводится к уравнению

$$\frac{dx}{dt} = [(p - p_0) - x^2(t)]x(t) + (\tau - \tau_0),$$

или

$$\frac{dx}{dt} = -\frac{\partial}{\partial x}V(x, p, \tau), \qquad (6.10)$$

где

$$V(x, p, \tau) = \frac{1}{4}x^4 - \frac{1}{2}(p - p_0)x^2 - (\tau - \tau_0)x. \qquad (6.11)$$

Из вида выражения (6.11) видим, что сервер – это потенциальная динамическая система, потенциал которой описываются катастрофой «сборка» [4].

Естественно предположить, что в повседневных рутинных условиях сервер имеет в среднем одни и те же производительность p и трафик τ. При этом число откликов в среднем является более или менее постоянным, т. е. $x(t) = x_0 = const$. В таком случае

$$\frac{dx}{dt} = 0$$

и, следовательно, $x_0 = x_0(p, \tau)$ – это решение уравнения

$$\frac{\partial}{\partial x}V(x_0, p, \tau) = 0.$$

Такие решения называются состояниями *стационарного равновесия* системы. Сервер, таким образом, пребывает, как правило, в состоянии стационарного равновесия. Точки равновесия (x_0, p, τ) находятся в пространстве с осями x, p, τ и началом $(0, p_0, \tau_0)$ на поверхности M_V (рис. 6.2).

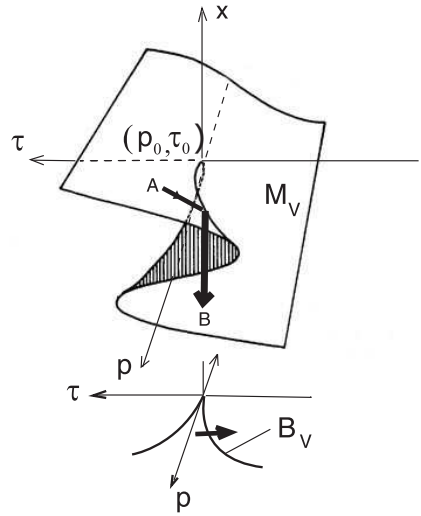


Рис. 6.2. Катастрофа сборки

Из рис. 6.2 видно, что если компьютерная система имела производительность $p < p_0$, т. е. не очень высокую, и трафик $\tau > \tau_0$ и находилась в равновесии A, то при нарастании трафика (жирная стрелка на рис. 6.2 от A к B) происходит скачкообразное обрушение такой характеристики, как количество откликов на запросы. Другими словами, происходит переход к равновесию «упавшего» сервера B.

Глава 7

Защита компьютерной системы от НСД

Защита компьютерной системы начинается с ее описания, описания доступных средств защиты и предъявляемых к системе требований.

После этого строится множество допустимых (доступных) проектов защиты, составляющих перечень стратегий администратора.

Составляется перечень возможных способов атаки на компьютерную систему со стороны злоумышленников. Это набор его стратегий нападения.

Поиск оптимального по соотношению затрат и ожидаемого эффекта проекта защиты можно теперь осуществить в рамках конечной одношаговой бескоалиционной игры двух игроков.

Рассмотрим матричную игру, предложенную С.А. Нестеровым [39].

7.1. Матрица игры

Стратегии администратора – игрока А – заключаются в установке в компьютерной системе одного из проектов защи-

ты или отказе от каких-либо действий. Обозначим множество проектов подсистемы защиты компьютерной системы через Z, а текущее состояние системы как C . Тогда администратор будет выбирать стратегии, соответствующие элементам множества $A = Z \cup \{C\}$.

Обозначим через U конечное множество обобщенных угроз информационной безопасности защищаемой компьютерной системы.

Под обобщенной угрозой понимается совокупность угроз, сходных по оказываемому на компьютерную систему воздействию и причиняемому ущербу. Разбиение всего множества угроз на обобщенные угрозы формируется на базе экспертных оценок.

Игровая стратегия злоумышленника – игрока X – заключается в выборе элемента из множества $X = U \cup \{U_0\}$, где U_0 – отказ от реализации угроз информационной безопасности.

Будем рассматривать злоумышленника X источником всех угроз безопасности: как преднамеренных, так и случайных.

Конечная одношаговая антагонистическая игра Γ_H (матричная игра) задается платежной матрицей H в виде:

$$H = \begin{array}{c} \\ Z_1 \\ \dots \\ Z_{m-1} \\ C \end{array} \begin{array}{c} \begin{array}{cccc} U_1 & \dots & U_{n-1} & U_0 \end{array} \\ \begin{bmatrix} -h_1 - \overline{h}_{11} & \dots & -h_1 - \overline{h}_{1(n-1)} & -h_1 \\ \dots & \dots & \dots & \dots \\ -h_{m-1} - \overline{h}_{(m-1)1} & \dots & -h_{m-1} - \overline{h}_{(m-1)(n-1)} & -h_{m-1} \\ -\overline{h}_{m1} & \dots & -\overline{h}_{m(n-1)} & 0 \end{bmatrix} \end{array},$$

где $\overline{h}_{ij}$ – оценки потерь от реализации злоумышленником j-й обобщенной угрозы в отношении компьютерной системы, где реализован i-й проект защиты; h_i – затраты на реализацию i-го проекта. Обе составляющие берутся со знаком минус, т. к. для владельца компьютерной системы (администратора) это потери (отрицательный выигрыш).

Построенная антагонистическая игра отражает ситуацию наиболее пессимистичного прогноза: предполагается, что зло-

умышленник очень способный, опытный и имеет задачу нанести максимальный вред.

Достоинством предложенной игровой модели «администратор-хакер» является то, что она позволяет учесть не только стоимость, но и особенности внедряемого проекта (через изменение оценок ожидаемых потерь) [39].

7.2. Решение игры

Для нахождения решения игры, т. е. выбора наиболее предпочтительной стратегии защиты, выбора проекта защиты, необходимо определиться, какой (классический) критерий принятия решения будет использоваться.

Например, можно использовать критерий Вальда (максиминный критерий) или Лапласа (критерий недостаточного основания). Первый из них отражает позицию владельца компьютерной системы, готовящегося к самому худшему исходу, а второй – позицию «снижения среднего значения ожидаемых потерь». В реальной ситуации представляется целесообразным применение нескольких критериев и сравнение результатов.

Если для построенной игры существует решение в чистых стратегиях, то это указывает наиболее предпочтительный проект (или проекты) системы защиты компьютерной системы. В этом случае оптимальные игровые стратегии будут совпадать с оптимальными стратегиями, выбранными в соответствии с критерием Вальда. Значение игры покажет максимальные ожидаемые потери при реализации наилучшего проекта.

Если решение существует только в смешанных стратегиях, оно нуждается в дополнительной интерпретации: в реальной компьютерной системе невозможно поочередно использовать различные проекты защиты, как это предполагается определением смешанной стратегии. Иначе говоря, нельзя надеяться, что, используя некоторый проект, скажем, 70% времени, надеяться, что в это время злоумышленник применяет угрозы, успешно отражаемые именно данным способом защиты.

Более правильно будет отобрать проекты защиты, попавшие в спектр оптимальной стратегии, и попытаться сформировать компромиссный вариант, объединяющий их сильные стороны.

7.3. Биматричная игра. Учет информации о злоумышленнике

Предположим, что имеется возможность достоверно оценить как возможности злоумышленника, так и ценность для него результатов атаки на компьютерную систему. В таком случае для нахождения стратегии защиты компьютерной системы предлагается [39] использовать биматричную игру Γ_{HH_1}.

Такая игра, основанная на дополнительных знаниях о злоумышленнике, дает менее пессимистичный прогноз, касающийся построения эффективной защиты информационных ресурсов.

Для этого, кроме приведенной выше платежной матрицы H, рассматриваемой как матрица выигрышей администратора, вводится дополнительная матрица выигрышей злоумышленника:

$$H_1 =$$

$$\begin{array}{c} \\ Z_1 \\ \dots \\ Z_{m-1} \\ C \end{array} \begin{array}{c} \begin{array}{cccc} U_1 & \dots & U_{n-1} & U_0 \end{array} \\ \begin{bmatrix} \widetilde{h}_{11} - \widehat{h}_{11} & \dots & \widetilde{h}_{1(n-1)} - \widehat{h}_{1(n-1)} & 0 \\ \dots & \dots & \dots & \dots \\ \widetilde{h}_{(m-1)1} - \widehat{h}_{(m-1)1} & \dots & \widetilde{h}_{(m-1)(n-1)} - \widehat{h}_{(m-1)(n-1)} & 0 \\ \widetilde{h}_{m1} - \widehat{h}_{m1} & \dots & \widetilde{h}_{m(n-1)} - \widehat{h}_{m(n-1)} & 0 \end{bmatrix} \end{array}$$

где $\widetilde{h}_{ij}$ – оценка выигрыша злоумышленника от реализации j-й угрозы в отношении компьютерной системы, где реализован i-й проект; $\widehat{h}_{ij}$ – оценка затрат хакера на реализацию этой угрозы. Для угроз, источниками которых являются случайные события (например, сбой оборудования), принимаем $\widehat{h}_{ij} = 0$, а

$\widetilde{h}_{ij}$– равным по модулю соответствующему элементу матрицы выигрышей игрока A. Нули в последнем столбце матрицы выигрышей злоумышленника соответствуют отказу от атаки на компьютерную систему.

Пример 7.1. Продемонстрируем биматричную задачу с уже заданными матрицами платежей H, H_1. Игрок P1 имеет три стратегии, а игрок P2 – четыре.

Имеем

$$H = \begin{pmatrix} 1 & 1 & 2 & 9 \\ 2 & 3 & 0 & -1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$$

$$H_1 = \begin{pmatrix} 1 & 2 & 0 & 5 \\ 7 & 0 & 0 & -2 \\ 4 & 1 & 1 & -3 \end{pmatrix}$$

Обратимся к on-line программе Solve a Bimatrix Game (VIRTUALIZED)[1], которая позволяет находить решения биматричных игр. В результате находим три ситуации равновесия с соответствующими платежами: одна из них смешанная $s = (9/13, 14/13, 0)$, $\sigma = (10/11, 0, 0, 1/11)$ и две чистые $s = (0, 1, 0), \sigma = (1, 0, 0, 0)$ и $s = (1, 0, 0), \sigma = (0, 0, 0, 1)$.

EE = Extreme Equilibrium, EP = Expected Payoff

Decimal Output

EE1 P1: 0.692 0.307 0.0 EP= 1.727 P2: 0.909 0.0 0.0 0.090 EP= 2.846
EE2 P1: 0.0 1.0 0.0 EP=2.0 P2: 1.0 0.0 0.0 0.0 EP=7.0
EE3 P1: 1.0 0.0 0.0 EP=9.0 P2: 0.0 0.0 0.0 1.0 EP=5.0

Rational Output

EE1 P1: 9/13 4/13 0 EP=19/11 P2: 10/11 0 0 1/11 EP=37/13
EE2 P1: 0 1 0 EP=2 P2: 1 0 0 0 EP=7
EE3 P1: 1 0 0 EP=9 P2: 0 0 0 1 EP=5

[1]Сайт: http://banach.lse.ac.uk/

Глава 8

Размещения конфиденциальной информации на серверах

Одной из важных задач является задача безопасного размещения информации на серверах корпоративной информационной системы.

8.1. Теоретико-игровой подход

В работах В.В. Дятчина, П.И. Тутубалина, К.В. Бормотова [25] и С.А. Нестерова [40] предложено использовать теоретико-игровые модели, где одним из игроков выступает администратор системы защиты информации корпоративной информационной системы (игрок А), а другим – потенциальный нарушитель (игрок В).

Рассмотрим ситуацию с точки зрения администратора системы защиты информации корпоративной информационной

системы. В распоряжении игрока А находится n стратегий $x_1, x_2, ..., x_n$, где x_i – стратегия игрока А, состоящая в том, что конфиденциальные данные нужно расположить на i-м сервере. В распоряжении нарушителя также находится n стратегий $y_1, ..., y_n$, где y_j – стратегия игрока В, состоящая в том, что конфиденциальные данные нужно искать на j-м сервере.

Построим платежную матрицу (a_{ij}) для игрока А:

$$(a_{ij}) = \begin{pmatrix} P + c_1 & c_1 & ... & c_1 \\ c_2 & P + c_2 & ... & c_2 \\ ... & ... & ... & ... \\ c_n & c_n & ... & P + c_n \end{pmatrix} \tag{8.1}$$

a_{ij} – потери игрока А, если атаке подвергается j-й сервер, а конфиденциальные данные находятся на i-м сервере (т. е. игрок А выбрал стратегию x_i). Здесь $P > 0$ – это материальный (в финансовом смысле) ущерб, наносимый системе при нарушении конфиденциальности ее данных, а c_i – стоимость хранения конфиденциальных данных на i-м сервере.

Обозначим через

$$s = (p_1, ..., p_n)$$

смешанную стратегию игрока А, в которой чистые стратегии x_i принимаются с вероятностями p_i. Обозначим множество смешанных стратегий игрока А через S_A. Как известно, любая чистая стратегия x_i принадлежит множеству смешанных стратегий S_A.

Смешанные стратегии, которыми руководствуется администратор системы защиты информации, определяют случайный механизм размещения реальных конфиденциальных данных на серверах баз данных корпоративной информационной системы. При использовании этих стратегий конфиденциальные данные будут случайно размещаться на одном из серверов корпоративной информационной системы, а на других серверах в это время будут присутствовать «ложные» файлы. В таком случае пользователь, запрашивающий данные, будет знать адрес только соответствующего сервера приложений, а точное

их местоположение в текущий момент будет определяться системными программными средствами данного сервера с одновременным обеспечением свойства прозрачности доступа.

8.2. Постановка игровой задачи

Следуя принципу гарантированного результата, определим в качестве наилучшего поведения для игрока А применение гарантирующей смешанной стратегии [31]. Гарантирующая смешанная стратегия игрока А находится как решение следующей задачи линейного программирования [24, 41]:

$$\nu \to \min, \tag{8.2}$$

при выполнении условий

$$\sum_{i=1}^{n} p_i a_{ij} - \nu \le 0, \quad j = 1, ..., n, \tag{8.3}$$

$$\sum_{i=1}^{n} p_i = 1, \quad p_i \ge 0, \quad i = 1, ..., n. \tag{8.4}$$

В качестве дополнительного критерия будем использовать среднюю стоимость C размещения конфиденциальных данных на серверах системы:

$$C = \sum_{i=1}^{n} c_i a_{ij} \to \min. \tag{8.5}$$

Построим множество оптимальных, по Парето, решений [37] с использованием линейной свертки критериев (8.2), (8.5) вида:

$$L(\alpha, p_1, ..., p_i, ..., p_n, \nu) = \alpha\nu + (1-\alpha)\sum_{i=1}^{n} c_i a_{ij} \to \min. \tag{8.6}$$

Варьируя параметр свертки $\alpha \in [0, 1]$ в указанном интервале, получим множество решений $\{(p_1, ..., p_n)\}$, оптимальных по

Парето. Это множество предоставляется игроку А, который, исходя из сопоставления значений ν и C, выбирает конкретный вариант значений $p_1^0, ..., p_n^0$.

Рассмотрим вопрос реализации случайного механизма размещения конфиденциальных данных. Пусть p_i^0, $i = 1, ..., n$ полученные из выбранного администратором варианта вероятности размещения конфиденциальных данных на серверах сети. Построим с их использованием интервалы

$$[0, p_1^0), [p_1^0, p_1^0 + p_2^0), ..., [\sum_{i=1}^{k-1} p_i^0, \sum_{i=1}^{k} p_i^0), ..., [\sum_{i=1}^{n-1} p_i^0, 1).$$

Далее в начале каждого рабочего дня или в течение каждого часа будем генерировать равномерно распределённое число $\xi \in [0, 1]$. Если это число попадает в некоторый k-й интервал, где $k = 1, ..., n$, то в этот день или час конфиденциальные данные располагаются на k-м сервере.

8.3. Размещение информации при использовании баз данных класса MS SQL Server

Рассмотрим один из вариантов реализации механизма случайного размещения информации при использовании в корпоративной информационной системе серверов баз данных класса MS SQL Server. Если случайному размещению подлежат отдельные информационные объекты, для их перемещения могут быть использованы следующие подходы [25]:

1. Использование хранимых процедур с распределенными запросами. Механизм распределенных запросов MS SQL Server позволяет обращаться в пределах одного запроса к другим серверам MS SQL Server. Таким образом, для реализации рассматриваемого подхода требуется создать SQL-процедуру, которая производит копирование данных с внешнего сервера на текущий и удаление данных на внешнем сервере.

2. Использование средств встроенного в MS SQL Server механизма преобразования данных DTS (Data Transformation Services – служба преобразования данных). Данные средства позволяют осуществлять копирование/перемещение данных как между серверами MS SQL Server, так и с внешними механизмами хранения с помощью технологии универсального доступа OLE DB. Выполнение процедуры перемещения в таком случае может быть инициировано с помощью графических средств самого сервера MS SQL, из командной строк или из внешних программ.

Кроме этого, среда MS SQL Server предоставляет возможности автоматического запуска как хранимых процедур, так и пакетов в DTS по расписанию в строго определенные моменты времени с помощью службы SQL Agent.

Оба представленных варианта предполагают хранение информации о текущем местоположении конфиденциальных данных на одном из серверов группы. Данная информация модифицируется соответствующим образом в ходе выполнения каждой операции перемещения.

Таким образом, можно обеспечить ежедневный (ежечасный) запуск процедуры случайного выбора сервера и перемещения конфиденциальных данных по серверам корпоративной информационной системы в соответствии с методикой, описанной выше.

Глава 9

Борьба с вирусами

Теорию игр можно применять при организации противодействия заражению компьютерной сети интеллектуальными вирусами. В качестве системы защиты от таких вирусов может использоваться самая простая IDS (Intrusion Detection System).

Assane Gueye [54] предложил описать противодействие вирусов и IDS как стратегическую (статичную) игру с нулевой суммой с неполной информацией, где IDS настроен на обнаружение превышения порога x входящего трафика[1], а вирус неизбежно увеличивает трафик на величину β, которая задана его создателем и определяет степень заражения компьютерной сети.

9.1. Построение игры

IDS обнаруживает вторжения, анализируя объем входящего трафика. Будем считать, что повседневный трафик характеризуется величиной α.

[1] *Трафик* – это объем данных, которые проходят через сервер за какое-то определенное время. Измеряться он может в килобайтах, мегабайтах и гигабайтах, в зависимости от масштабов.

IDS настраивается на некоторое пороговое значение трафика x. Превышение объема трафика пороговой величины x на некотором временном интервале IDS расценивает как проникновение вируса. Принимаем, что IDS такова, что можно регулировать величину порога x.

Интеллектуальный вирус всегда выбирает уровень заражаемости, посылая на атакуемый сервер один или несколько вирусов, которые, если подсчитать получаемый при этом объем посылаемых пакетов, приводят к дополнительному увеличению входящего трафика на величину β. Интеллектуальный вирус балансирует между агрессивным нападением с большим значением величины β, которое может быть легко обнаружено по наносимому им значительному ущербу, и «легким» нападением с небольшим значением величины β, которое наносит меньше ущерба.

IDS берет в буфер и измеряет объем X_n трафика в интервале $[(n-1)T, nT]$ для $n = 1, 2, 3, ...$ и решает, что вирус присутствует в первый раз, когда $X_n > x$, где T является параметром системы защиты и предполагаемый быть известным и x является порогом, который будет выбран IDS. Если IDS решает, что есть инфекция, то он стирает содержимое буфера, чтобы препятствовать тому, чтобы вирус заразил машины сети. Буферизация вводит некоторую задержку (T) в трафик.

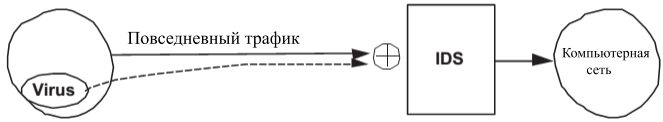


Рис. 9.1. Вирусная атака [54]

Противостояние вируса и IDS рассматриваем как матричную игру с нулевой суммой.

9.1.1. Стратегии

Стратегии IDS – это выбор величины порога x, а стратегии вируса – выбор уровня заражаемости β. Наша задача – найти оптимальную пару стратегий (x_*, β_*) в смысле равновесия Нэша.

9.1.2. Платежная матрица

Считаем, что компьютерная сеть имеет два состояния: неинфицированное состояние 0 и инфицированное – 1.

Пусть q – это вероятностью того, что компьютер заражен на некотором временном шаге, т. е. q – вероятность перехода $0 \to 1$.

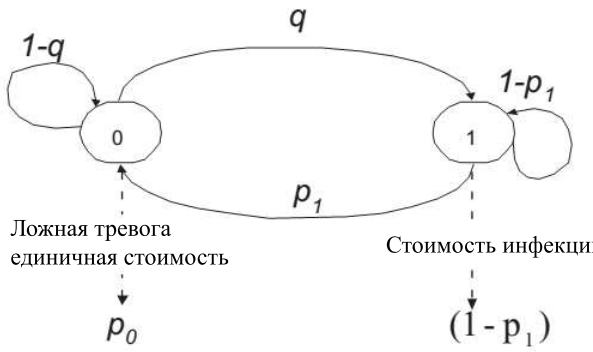


Рис. 9.2. Схема состояний компьютерной сети [54]

На рис. 9.2 показана диаграмма состояний компьютерной сети с переходными вероятностями $P(i,j) : i \to j$ $(i,j = 0$ или $1)$ при смене состояний в зависимости от работы IDS (цепь Маркова).

При этом $p_0 = P[X_n > x|\text{Нет вируса}]$ вероятность того, что IDS объявляет, что компьютер заражен, когда это не верно (ложная тревога) и $p_1 = P[X_n > x|\text{Вирус есть}]$ вероятность того, что IDS объявляет, что компьютер заражен, когда это

верно. Пусть q – вероятность заражения компьютера на временном шаге. Ясно, что $P(0,1) = q$.

В случае ложной тревоги производится проверка сети и стоимость проведенных работ принимаем как единичную, т. е. равную 1. Если IDS правильно объявил о заражении, т. е. сеть находится в состоянии 1, то он стирает данные в буфере, сеть очищается от вируса и переходит в состояние 0. Значит, $P(1,0) = p_1$.

Когда сеть заражена, т. е. находится в состоянии 1, но IDS не заметил проникновения вируса, а это возможно с вероятностью $(1 - p_1)$, то проникло среднее число вирусов, равное $\beta(1-p_1)$, предполагая, что каждый пропущенный вирус наносит ущерб, равный γ.

Следовательно, средняя стоимость нанесенного ущерба в единицу времени равна

$$C(x, \beta) = p_0\pi_0 + \gamma\beta(1 - p_1)\pi_1,$$

где π_0 (соотв. π_1) – вероятность, что система находится в состоянии 0 (соотв. 1).

Рассматриваем *стационарный* случай игры. Тогда

$$q\pi_0 = p_1\pi_1 \quad \text{и} \quad \pi_0 + \pi_1 = 1.$$

Следовательно,

$$C(x, \beta) = \frac{p_0p_1 + \gamma\beta(1 - p_1)q}{p_1 + q}. \tag{9.1}$$

Злоумышленник, создавший данный интеллектуальный вирус, выбирает зараженность β так, чтобы максимизировать стоимость $C(x, \beta)$, в то время как IDS вычисляют лучший порог x, чтобы минимизировать ее.

Если $\beta > \alpha$, то со 100%-й вероятностью вирус будет обнаружен. Поэтому создатель вируса настраивает его так, чтобы всегда имели, что $\beta < \alpha$.

Для того, чтобы задать матрицу платежей, надо вычислить вероятности p_0, p_1. Вероятность q считаем параметром игры.

Разумно предположить, что величина трафика X_n является случайной и распределена равномерно на $[0, \alpha]$, когда компьютер не заражен, и равномерно на $[\beta, \alpha + \beta]$, когда компьютер заражен (вирус увеличивает трафик на β). При этих предположениях находим вероятности p_0 и p_1 как функции от x и α (табл. 9.1).

Таблица 9.1

Значения вероятностей перехода

x	p_0	p_1
$0 \le x \le \beta$	$\frac{\alpha - x}{\alpha}$	1
$\beta < x \le \alpha$	$\frac{\alpha - x}{\alpha}$	$\frac{\alpha + \beta - x}{\alpha}$
$\alpha < x \le \alpha + \beta$	0	$\frac{\alpha + \beta - x}{\alpha}$

Матрица платежей представлена в табл. 9.2, где

$$C_3(x, \beta) = \frac{\alpha + \beta(1 - \gamma\beta q + (x^2 + (\alpha\gamma\beta q - 2\alpha - \beta)))}{\alpha(q+1) + \beta - x}.$$

Таблица 9.2

Матрица платежей игры

		Стратегии β
Стратегии $\boldsymbol{\chi}$	$0 \le x \le \beta$	$\frac{1 - \frac{\alpha}{x}}{q+1}$
	$\beta < x \le \alpha$	$C_3(x, \beta)$
	$\alpha < x \le \alpha + \beta$	$\gamma\beta q \frac{x - \beta}{\alpha(q+1) + \beta - x}$

9.2. Результаты моделирования

Так как игра является игрой с нулевой суммой, то стратегии равновесия Нэша соответствуют стратегиям максимина.

Следовательно, чтобы найти равновесия Нэша, надо минимизировать $C(x, \beta)$ по x в каждом из интервалов для x, данных выше в табл. 9.2. Для каждого интервала это дает функцию, которая только зависит от β. Мы тогда максимизируем ее по β в каждом интервале и находим в нем максимальное значение. Вычисления можно проделать, например в Matlab. Равновесие Нэша зависит от параметров q и α. В табл. 9.3 приведены равновесия Нэша для $\alpha = 1000$.

Таблица 9.3

Равновесия Нэша для $\alpha = 1000$

q	γ	β	x
0.01	0.02	150	980
0.01	0.05	220	900
0.01	0.10	250	750
0.10	0.02	250	580
0.10	0.05	150	310
0.10	0.10	90	180

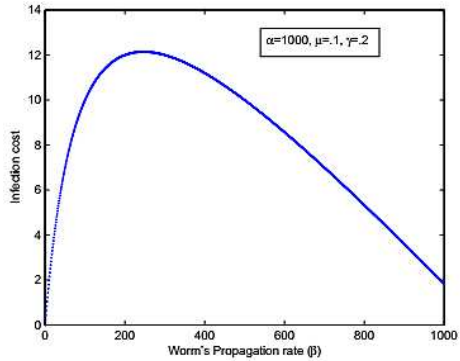


Рис. 9.3. Стоимость потерь при заражении [54]

На рис. 9.3 представлен график стоимости $C(x(\beta), \beta)$ в зависимости от стратегии, принятой интеллектуальным виру-

сом. Для каждой стратегии заражения β вычислен лучший ответ IDS, который соответствует оптимальному порогу $x = x(\beta)$ для этого уровня заражаемости. Затем мы вычисляем стоимость по формуле (9.1).

Как видно из графика, цена потерь от нападения вируса маленькая, когда зараженность является небольшой. Это говорит, что менее агрессивный вирус нанесет меньше ущерба компьютерной системе. График также показывает, что маленькая стоимость потерь наблюдается для высоких уровней заражаемости. Следовательно, очень агрессивный вирус наносит, как ни удивительно, маленький ущерб системе. Однако, если подумать, то удивление было чисто эмоциональным. Ведь очевидно, что агрессивное нападение с большим β будет с большой вероятностью обнаружено на уровне IDS, который примет необходимые меры по обезвреживанию опасности, и вирус не проникнет в компьютерную систему.

Таким образом, используя построенную теоретико-игровую модель, можно гарантировать определенный уровень безопасности независимо от того, насколько агрессивен нападающий вирус. Устанавливая порог $x = x_*$, т. е. отвечающий равновесию Нэша, мы добьемся того, что стоимость $C(\beta, x_*)$ потерь от инфицирования будет всегда меньше, чем $C(\beta_*, x_*)$.

Глава 10

Позиционные игры

Позиционная игра – это последовательная многоходовая бескоалиционная игра N лиц.

В позиционной игре ходы делаются последовательно. Каждый ход делается либо одним из игроков P_i (личный ход) $(i = 1, ...N)$, либо выбирается случайным образом (случайный ход) в соответствии с заданным распределением вероятностей. В каждой конечной позиции игры задан вектор выигрышей игроков.

Позиционные игры описываются с помощью *дерева игры*.

10.1. Графы и деревья

Граф – это пара (V, E), где V – конечное множество вершин (узлов), а и E – множество ребер. Ребро состоит из двух вершин.

Ориентированным графом, или *орграфом*, называется граф (V, E), ребра которого упорядочены, т. е. для каждого ребра $\{v, w\}$ сказано, которая из двух вершин ребра первая, а которая вторая. Если v – первая, а w – вторая, то упорядоченное ребро записываем как (v, w). Упорядоченное ребро (v, w) называем *дугой*. Также говорят, что дуга $e = (v, w)$ выходит из

вершины v и входит в вершину w.

Последовательность вершин $v_0, v_1, ..., v_k$ называется *путем* из вершины v_0 в вершину v_k длины k в графе (орграфе) (V, E), если $(v_{i-1}, v_i) \in E$ для $i = 1, ..., k$. Путь называется простым, если в нем нет повторяющихся вершин. Замкнутый (когда $v_0 = v_k$) путь называют *циклом*. Простой цикл не имеет повторяющихся вершин.

Граф называется *связным*, если между любыми его двумя вершинами имеется путь.

Дерево – это связный граф без циклов. Дерево $D = (V; E)$, состоит из множества вершин (узлов) V и множества дуг E.

Ориентированным деревом называется ориентированный граф, если[1] $|E| = |V| - 1$ и в каждую вершину входит не более одной дуги. Существует единственная вершина, в которую не входят дуги, она называется *корнем*. Вершины, из которых не выходят дуги, называются *листьями*.

К любому листу в ориентированном дереве из корня ведет только один путь.

10.2. Дерево игры

Позиционная игра игроков $P_1, ..., P_N$ представляется *ориентированным деревом игры* $D = (V, E)$.

Позиционная игра является многоходовой игрой.

Каждая вершина – это *позиция игры*. Она метится либо обозначением игрока P_i $(i = 1, ..., N)$, либо буквой O, говорящей о том, что ход в этой позиции делается случайным образом.

Буква O говорит о том, что ход делается не игроком, а каким-либо случайным механизмом, часто называемым *природой*.

Ход в позиции игры, помеченной игроком, – это ход данного игрока и задается он любой дугой, исходящей из вершины.

[1] Через $|A|$ обозначена мощность множества A.

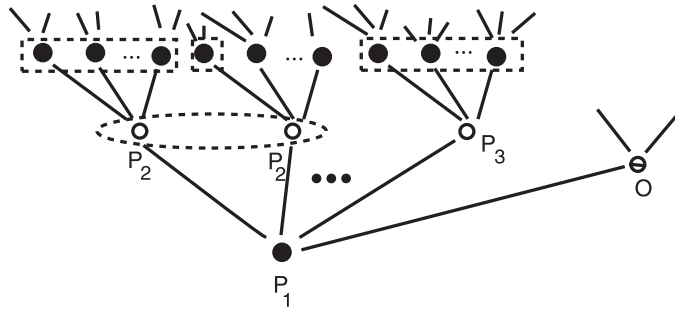


Рис. 10.1. Дерево игры. Пунктирными овалом и прямоугольниками обозначены информационные множества

Корень дерева – это выделенная вершина, она соответствует *начальной позиции* игры. С нее она начинается. Листья являются *конечными позициями*.

Если вершина помечена игроком P_i, то это говорит о том, что в этой позиции ход делает игрок P_i. Дуги, выходящие из позиции и соединяющие ее со следующими за ней, называются *альтернативами*.

Дугам e_v, выходящим из узла v с меткой O, приписаны вероятности $p_v(e_v)$ с которыми делается соответствующий ход e_v

$$\sum_{e_v} p_v(e_v) = 1.$$

Листья – конечные вершины игры; это позиции, их которых ходы не делаются. Каждому листу l приписан вектор $(\pi_1(l), ..., \pi_N(l))$ выигрышей игроков в случае, когда игра заканчивается в данной вершине l.

Из корня к листу ведет единственный путь. Такой путь называется *партией*. Процесс игры состоит в том, что происходит переход вдоль пути, ведущего из корня к листу, через промежуточные позиции, по которым проходит путь.

10.3. Информационные множества

Информация в игре задается с помощью информационных множеств.

Множество вершин разбито на непересекающиеся подмножества, называемые информационными множествами.

Вершины, входящие в одно информационное множества – это позиции одного и того же игрока.

Две вершины (позиции) принадлежат одному *информационному множеству*, если игрок, который должен делать ход в каждой из этих позиций, не может отличить одну позицию от другой.

Таким образом, информационное множество – это некоторое множество вершин, помеченных одним и тем же игроком P_i. Игрок P_i при этом находится в позиции, из которой он делает ход, но он не знает, как оказался в этой позиции (вершине). Таким образом, если информационное множество не сводится к одной вершине, то игрок не обладает полной информацией об игре.

Из данного определения следует, что из всех вершин одного информационного множества выходит одинаковое число дуг.

Информационное множество – это совокупность позиций, которые игрок не различает между собой. Поэтому, находясь в любой позиции информационного множества, игрок имеет один и тот же набор альтернатив.

Игрок P_i может иметь несколько различных информационных множеств $I_1^{(i)}, ..., I_{n_i}^{(i)}$. Естественно, одинаковые в смысле выбираемой игроком стратегии ходы, которые делаются игроком в позициях разных информационных множеств, – это различные ходы этого игрока.

Информационное множество вершины, помеченной O, состоит только из этой вершины.

Если L – партия игры, т. е. путь, идущий от корня дерева к одному из его листьев, и если I – любое информационное множество, то существует не больше одной вершины, принадлежащей обоим множествам L и I. Другими словами, ни одна

партия не должна пересекать информационное множество более одного раза.

Если все информационные множества игры состоят из одной вершины, то имеем *игру с полной информацией*. В противном случае имеем *игру с неполной информацией*, или *байесовскую игру*.

10.4. Стратегии игроков в позиционной игре

Пусть $A_j^{(i)}$ – множество ходов игрока P_i в позициях информационного множества $I_j^{(i)}$ $(1 \le j \le n_i)$.

Стратегия игрока P_i – кортеж

$$s^{(i)} = (a_1^{(i)}, ..., a_{n_i}^{(i)}),$$

где $a_j^{(i)} \in A_j^{(i)}$. Соответственно, множество всех стратегий игрока P_i – это множество

$$S_i = \{s^{(i)} = (a_1^{(i)}, ..., a_{n_i}^{(i)}) : a_j^{(i)} \in A_j^{(i)}\}.$$

Как видим, через $a_j^{(i)}$ обозначен конкретный ход из множества возможных $A_j^{(i)}$, который он будет делать, если игра достигнет какой-либо позиции информационного множества $I_j^{(i)}$.

Кортеж длины N

$$s = (s^{(1)}, ..., s^{(N)}) \in S_1 \times ... \times S_N, \tag{10.1}$$

состоящий из набора (чистых) стратегий всех N игроков будем называть *ситуацией*.

Нетрудно понять, что число стратегий у каждого игрока максимально в случае, если ведется игра с полной информации, а в случае неполной информации, доступной игроку, т. е. в случае, когда его информационное множество содержит более одной вершины, число его стратегий уменьшается.

Каждая ситуация s приводит игру к окончанию в одной из конечных позиций l некоторого подмножества $K(s)$ конечных позиций, являющихся листьями дерева игры.

Для листа l вычисляется выигрыш каждого игрока $\pi_i(l)$. Можно определить средний выигрыш игрока P_i в ситуации s посредством формулы

$$\overline{\pi}^{(i)}(s) = \sum_{l \in K(s)} p_l(s)\pi_i(l),$$

где $p_l(s)$ – вероятность единственного пути, ведущего корня (из начальной позиции) к листу l, получаемой перемножением вероятностей дуг, составляющих данный путь.

Точнее, если игрок осмысленно выбирает конкретную альтернативу, то вероятность этой дуги равна 1, а невыбранных – 0. Но вполне допустимо, что игроку предложен выбор альтернативы с некоторой вероятностью. В этом случае каждой альтернативе в рассматриваемой позиции игрока приписана вероятность, с которой она выбирается (естественно, сумма вероятностей, приписанных альтернативам, равна 1).

10.5. Равновесия в позиционных играх

Определение 2.1. Ситуация s называется *ситуацией равновесия* в чистых стратегиях тогда и только тогда, когда для каждого игрока P_i

$$\overline{\pi}^{(i)}(s) = \max_{r_i}\{\overline{\pi}^{(i)}(s||r_i)\}. \tag{10.2}$$

Далеко не все позиционные игры имеют ситуацию равновесия в чистых стратегиях. Но игры с с полной информацией всегда имеют ситуацию равновесия в чистых стратегиях.

Теорема 2.2 (Кун, [58]). *Каждая позиционная игра с полной информацией имеет ситуацию равновесия в чистых стратегиях.* ■

10.6. Нормализация позиционной игры

Позиционную игру можно свести к стратегической в один ход. Это называется *нормализацией* позиционной игры.

10.6.1. Сведение к стратегической форме

Для позиционной игры с деревом игры (V, E) и выигрышными функциями $(\pi_1(l), ..., \pi_N(l))$ *её нормальная* (*стратегическая*) *форма* – это бескоалиционная игра N лиц $P_1, ..., P_N$ со стратегиями $S_1, ..., S_N$ и функциями выигрыша $(\overline{\pi}^{(1)}(s), ..., \overline{\pi}^{(N)}(s))$, $s = (s^{(1)}, ..., s^{(N)}) \in S_1 \times ... \times S_N$.

Однако, если найдена ситуация равновесия для игры в нормальной форме, надо иметь в виду, что не все эти ситуации равновесия являются ситуациями равновесия для исходной игры в позиционной форме. Причина этого в том, что при переходе к нормальной (стратегической) форме теряется информация о последовательности ходов [43, с. 64].

10.6.2. Сведение к матричной игре

Если игроков только два и отсутствует случайный выбор, т.е. игрок O, то возможно сведение позиционной игры к матричной игре:

	$s_1^{(2)}$	...	$s_n^{(2)}$
$s_1^{(1)}$	$[\pi_{11}^{(1)}, \pi_{11}^{(2)}]$	...	$[\pi_{1n}^{(1)}, \pi_{1n}^{(2)}]$
...	...	...	...
$s_m^{(1)}$	$[\pi_{m1}^{(1)}, \pi_{m1}^{(2)}]$	...	$[\pi_{mn}^{(1)}, \pi_{mn}^{(2)}]$

где

$$\pi_{jk}^{(i)} = \overline{\pi}^{(i)}(s_j^{(1)}, s_k^{(2)}), \quad i = 1, 2, \quad j = 1, ...m, \quad k = 1, ..., n,$$

– выигрыш i-го игрока в ситуации

$$(s_j^{(1)}, s_k^{(2)}) \in S_1 \times S_2,$$

$s_1^{(1)}, ..., s_m^{(1)}$ – все стратегии 1-го игрока, а $s_1^{(2)}, ..., s_n^{(2)}$ – все стратегии 2-го игрока.

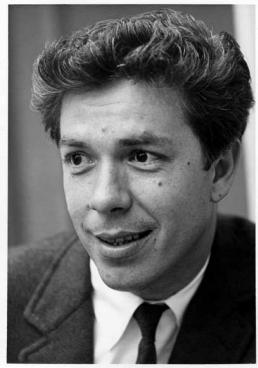
10.7. Процесс игры. Построения дерева игры

Игра начинается либо игроком P_i, либо случайным образом. Другими словами дерево строить начинаем вводя вершину, которая будет корнем, и помечая ее либо буквой P_i, либо буквой O.

На втором шаге из корня проводим дуги, являющиеся альтернативами, который делает игрок в корне. Это первый ход. Альтернативы соединяют корневого игрока с вершинами (позициями) игроков, которые делают второй ход и т. д. до тех пор, пока не будут обозначены окончательные вершины (листья).

По достижения любого листа игрокам назначается выигрыш.

10.8. Харольд Уильям Кун



Х. Кун (1961)

Создателем теории позиционных игр является Х. Кун, изложивший ее в статье [58].

Харольд Уильям Кун (род. в 1925 г.) – американский математик.

Служил в армии США с 1944 по 1946 год. Изучал японский язык по армейской программе в Ельском университете.

Закончил Калифорнийский технологический институт (степень бакалавра, 1947) и аспирантуру Принстонского университета (PhD, 1950). Научным руководителем по диссертации был Ральф Фокс. Один из авторов известной теоремы Куна-Такера.

Награжден премией Джона фон Неймана (1980).

Друг и коллега нобелевского лауреата Джона Нэша. Сыграл ключевую роль в присуждении Нэшу нобелевской премии. Представлял работы Нэша по теории игр на Церемонии награждения нобелевских лауреатов в 1994 году.

Почетный профессор Принстонского университета.

Кун и Нэш сотрудничали с Альбертом Такером – научным руководителем Нэша.

Глава 11

Защита компьютерной системы как позиционная игра

Для решения задачи защиты информации в компьютерных системах каждая компьютерная система должна включать подсистему защиты информации, обеспечивающую комплексную защиту информации.

Рассмотрим, каким образом данная проблема решается Г.Г. Грездовым с привлечением методов теории позиционных игр с неполной информацией [20].

11.1. Описание игры

Из теории игр известен способ, как обеспечить гарантированную границу своего проигрыша, хуже которого быть не должно [32]. Естественно строить защиту компьютерной системы, рассматривая все без исключения варианты использования существующих механизмов защиты информации, имеющиеся в распоряжении администратора. При этом каждый

вариант использования механизмов защиты информации будем описывать бинарным вектором γ.

Пусть злоумышленник – игрок I – обладает n стратегиями $x_1, ..., x_n$ атаки на компьютерную систему. Каждая такая стратегия x_i – это i-я угроза, способная нарушить работу компьютерной системы.

Администратор располагает m способами защиты ресурса. Рассмотрим $M = 2^m - 1$ вариантов защиты γ_j, где

$$y_j = (\gamma_{j1}, \gamma_{j2}, ..., \gamma_{jm}),$$

$$\gamma_{jk} = 0 \text{ или } 1,$$

где 1 на k-м место в кортеже y_j означает, что задействован k-й способ защиты. Ясно, что кортеж $(0, ..., 0)$ мы не принимаем в расчет, поскольку комплексная система защиты информации должна функционировать в составе всех компьютерных систем.

Стратегии администратора – игрока II – это кортежи

$$y_1, ..., y_M.$$

Ходами в игре выступают варианты использования существующих механизмов защиты информации в компьютерной системе, т.е. варианты вектора y_j. Дерево игры дано на рис. 11.1.

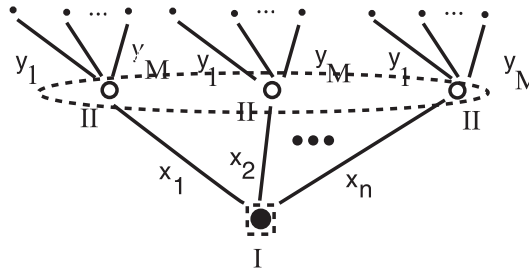


Рис. 11.1. Дерево игры

Поскольку игрок II не знает, какой способ атаки избрал хакер, то будем вести позиционную игру с неполной информацией (см. § 2.5).

Информационное множество для игрока II помечено на рис. 11.1 пунктирным эллипсом, а информационные множества хакера – пунктирным прямоугольником.

Множеством поиска решений будут все значения бинарных векторов y_j $(j = 1, ..., M)$ размерности m (за исключением вектора, состоящего из одних нулей).

Для каждого из бинарных векторов y_j необходимо вычислить размер остаточного риска

$$R(y_j) = \sum_{i=1}^{n} L_i(P_i - \sum_{k=1}^{m} G_{ik}\gamma_{jk}), \tag{11.1}$$

где

- L_i – оценка стоимости потерь в случае реализации i-й угрозы (руб.);
- P_i – вероятность реализации i-й угрозы;
- G_{ik} – эффективность k-го механизма защиты информации по нейтрализации i-й угрозы;

а также размер средств, выделяемых на обеспечение защиты информации в компьютерной системе

$$C_d(y_j) = \sum_{k=1}^{m} \gamma_{jk}(C_k + X_k), \tag{11.2}$$

где

- C_k – затраты на приобретение (разработку) и использование k-го механизма защиты информации (руб.),
- X_k – размер потерь компьютерной системы, вызванных использованием k-го механизма защиты информации в составе комплексной системы защиты информации компьютерной системы (руб.).

Выигрышные функции в листе, к которому ведет альтернатива y_j, для злоумышленника это $R(y_j)$, для администратора – $C_d(y_j)$.

Таким образом, нормализуя игру, получаем следующую (би)матричную игру:

	x_1	...	x_n
y_1	$[R(y_1), C_d(y_1)]$	...	$[R(y_1), C_d(y_1)]$
...	...	...	...
y_M	$[R(y_M), C_d(y_M)]$	...	$[R(y_M), C_d(y_M)]$

Теория игр позволяет найти решение, оптимальное или рациональное в среднем [32].

11.2. Определение вероятности проявления i-й угрозы

Для построения адекватной системы защиты информации необходимо реально оценить возможности злоумышленника.

При определении вероятностей проявления угроз информации будем полагать, что все множество угроз информации компьютерной системы формируется из множества активных и пассивных угроз. Причиной возникновения пассивных угроз будем считать уязвимости и особенности компонентов компьютерной системы, активных – действия злоумышленника.

Будем полагать, что вероятность i-й угрозы информации определяется через вероятность «активной» P_i^a и «пассивной» P_i^n составляющей угрозы. Значение P_i^n может быть получено статистическими методами или выведено с помощью метода экспертных оценок [30].

При оценке возможных действий злоумышленника сложно определить, какой именно из доступных способов будет им выбран для нанесения ущерба объекту защиты. Таблица 11.1 содержит порядок средств, выделяемых на реализацию атак различными категориями злоумышленников.

Таблица 11.1

Сравнительный анализ возможностей вероятного злоумышленника

Категории вероятного противника	Средства, выделяемые для реализации атак (доллары США)
Одиночки	100
Группы хакеров	1 000
Мелкие преступные группы	100 000
Крупные преступные группы	1 000 000
Транснациональные преступные организации, спецслужбы иностранных государств	100 000 000

Поэтому при получении составляющей P_i^a будем исходить из наихудших предположений о возможностях противника [30]. Значение $P_i^a = 1$, если финансовые возможности злоумышленника превышают стоимость хотя бы одного из средств нападения, способного вызвать дестабилизирующий фактор. В противном случае $P_i^a = 0$.

Вероятность проявления i-й угрозы может быть получена следующим образом:

$$P_i = \max(P_i^a, P_i^n).$$

11.3. Выбор стратегий

При организации защиты информации в компьютерных системах, обрабатывающих информацию, составляющую государственную, военную или коммерческую тайну, могут быть использованы самые различные критерии.

Перечислим стратегии [20], которые могут быть выбраны в случае, когда возможности злоумышленника значительно уступают возможностям владельца компьютерной системы:

1. Увеличить размер остаточного риска (R) при своих малых значениях средств, которые могут быть потрачены на реализацию атаки.

2. Увеличить затраты C_k на приобретение (разработку) и использование k-го механизма защиты информации, а также размер потерь компьютерной системы, вызванных использованием k-го механизма X_k защиты информации в составе комплексной системы защиты информации компьютерной системы.

3. Дезинформировать формирующего КСЗИ, чтобы существующая комплексная система защиты информации была перестроена. Указанная мера приведет к уменьшению C_d.

Рассмотрим стратегии, которые могут быть выбраны в случае, когда возможности злоумышленника равны возможностям владельца компьютерной системы [20]:

1. Найти такие γ_j, при которых у формирующего комплексную систему защиты информации не хватит средств, которые могут быть выделены на защиту информации в компьютерной системе (C_d).

2. Найти такие механизмы реализации атаки, чтобы у атакующей стороны на них C_d средств хватило, а у формирующего комплексную систему защиты информации – нет.

Наконец, в случае, когда возможности злоумышленника стороны значительно превышают возможности владельца компьютерной системы, можно использовать следующие стратегии [20]:

1. Атакующая сторона выбирает много вариантов для одновременной реализации атак, а у формирующего комплексную систему защиты информации не хватит финансовых средств для противодействия.

2. Атакующая сторона выбирает такие варианты реализации атаки на компьютерную систему, чтобы у формирующего комплексную систему защиты информации не хватило финансовых средств для построения эффективной комплексной системы защиты информации («тактика истощения»).

Глава 12

Моделирование поведения азартного злоумышленника

В этой главе покажем, как с помощью теории игр можно смоделировать поведение азартного злоумышленника, представляющее особую угрозу при атаках на информационные компьютерные ресурсы [6, 7].

12.1. Постановка и решение задачи

Будем понимать стратегии злоумышленника как строки x_i $(i = 1, 2, ..., n)$ некоторой матрицы, а стратегии администратора информационных ресурсов – как ее столбцы y_j $(j = 1, 2, ..., m)$. К стратегиям злоумышленника можно отнести различные виды компьютерных атак. Например, это может быть удаленное или локальное проникновение в компьютер, удаленное или локальное блокирование компьютера, применение сетевых сканеров для сбора информации о компьютерах сети и программах, потенциально уязвимых к атакам, исполь-

зование сканеров уязвимых мест программ в поисках компьютеров, уязвимых к тому или иному конкретному виду атаки, применение вскрывателей паролей, применение сетевых анализаторов (снифферов) и др.

К стратегиям администратора можно отнести различные варианты использования методов и средств защиты информации. Например, применение и регулярное обновление антивирусных программ, шифрование, использование межсетевых экранов и средств обнаружения атак, оперативная установка от производителей исправлений для программ (чтобы ликвидировать неблагоприятные последствия ошибок в них), применение вскрывателей паролей и сканеров уязвимых мест и др.

Для проведения на компьютере игры Γ_A надо также знать результаты игры a_{ij} при каждой паре стратегий x_i и y_j (например, a_{ij} – причиненный материальный ущерб) и вероятности реализации атак злоумышленников $p(x_i)$ при выбранной стратегии x_i. Построив игровую матрицу (табл. 12.1) и проанализировав ее, можно заранее оценить затраты каждого решения по защите компьютерной информации и рекомендовать наиболее эффективные варианты для всего диапазона атак.

Таблица 12.1

Платежная матрица и вероятности

		y_1	y_2	...	y_m
x_1	$p(x_1)$	a_{11}	a_{12}	...	a_{1m}
x_2	$p(x_2)$	a_{21}	a_{22}	...	a_{2m}
...	...	...	...	...	...
x_n	$p(x_n)$	a_{n1}	a_{n2}	...	a_{nm}

Если построена игровая матрица $A = (a_{ij})$, в которой результатами игры являются материальные потери от атак, то наилучшей в условиях имеющейся информации об атаках будет стратегия системы защиты компьютерной информации y_j, при которой будут минимальны средние потери, т.е. будет минимальна сумма:

$$\sum_{i=1}^{n} a_{ij} \cdot p(x_i).$$

Вероятности реализации атак $p(x_i)$ могут быть определены по результатам статистических исследований. Если вероятности атак неизвестны, то предполагается, что все они равновероятны, т. е. $p(x_i) = 1/n$.

Азартный злоумышленник увлечен желанием нанести как можно больший ущерб атакуемой компьютерной системе. В силу своей психологии он преувеличивает свои выигрыши и преуменьшает свои неудачи в предыдущих попытках атак на систему, воспринимая игру Γ_A как матричную игру $f(\Gamma_A)$ с матрицей $f(A)$, где f – так называемая функция полезности. В случае азартного нарушителя эта функция задается непрерывной выпуклой (вниз) вещественной функцией $f : \mathbb{R} \to \mathbb{R}$ [39, с.222]. На рис. 12.1 приводится вид функции полезности азартного злоумышленника. Такой функцией может являться, например, функция $f(a) = e^a - 1$.

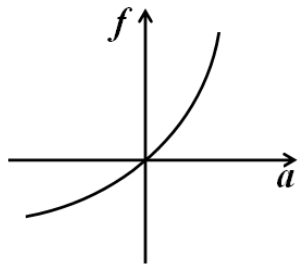


Рис. 12.1. Вид функции полезности азартного игрока [39, с. 222]

Обозначим через $val(A)$ значение матричной игры Γ_A с матрицей $A = (a_{ij})$. В случае азартной функции полезности имеют место утверждения [39]:

1) из $val(A) = 0$ следует $val(f(A)) \geq 0$, т. е. нарушитель может видеть победу там, где ее нет;

2) из $val(A) > 0$ следует $val(f(A)) \geq val(A)$, т. е. азартный нарушитель преувеличивает размер успеха;

3) при любом опыте l предыдущих вторжений существует такая игра Γ_{A_0}, что $val(A_0) < 0$ (реальный проигрыш, неудачная атака) и $val(A_0 + lE) > f(l)$, где E – матрица, состоящая

из единиц, т. е. азартный нарушитель всегда будет повторять некоторые проигрышные атаки (игру Γ_{A_0}).

Учет психологии азартного злоумышленника и моделирование его поведения позволяет строить ловушки либо для его идентификации, либо для направления его активности по ложному пути.

Применение игровых методов дает преимущества администратору безопасности перед субъективными случайными решениями и обеспечивает оптимизацию стратегий защиты компьютерной информации. Организация проигрышных атак и подробное исследование матричной игры Γ_{A_0} сводится к изучению психологии азартного нарушителя.

12.2. Оценка материальных потерь

Построение игровых матриц и выбор наиболее приемлемых решений при использовании игровых моделей требует оценки результатов функционирования систем защиты компьютерной информации в целом при различных возможных вариантах решений. Опишем один из способов определения коэффициентов a_{ij} матрицы игры A.

Единичные потери P_{ij}^1 при взломе j-й рабочей станции в случае однократной реализации угрозы x_i можно оценить следующим образом:

$$P_{ij}^1 = R_j k_i,$$

где R_j – стоимость ресурса «рабочая станция пользователя» при использования j-й комбинации методов и средств защиты; k_i – процент потерь в случае реализации угрозы x_i на данном ресурсе. Стоимость ресурса R_j обычно включает стоимость сопровождения и восстановления, прямые затраты на покупку и обновление соответствующего оборудования и программного обеспечения, расходы на поддержание информационной системы, административные расходы, затраты на обучение пользователей и убытки от вынужденных простоев.

Годовая оценка инцидента N_i, т. е. число, отражающее частоту проявления угрозы x_i в год, может быть рассчитана так:

$$N_i = s\nu_i,$$

где s – число подверженных атаке рабочих станций и ν_i – частота реализации угрозы x_i в год (может быть найдена на основе собственного опыта или усредненной статистической информации).

Годовые потери P_{ij} j-й рабочей станции в результате реализации угрозы x_i можно оценить следующим образом:

$$P_{ij} = P_{ij}^1 N_i.$$

В качестве коэффициентов a_{ij} матрицы игры A можно рассматривать годовые потери P_{ij} для всех вариантов комбинаций x_i $(i = 1, 2, ..., n)$ и y_j $(j = 1, 2, ..., m)$.

12.3. Опасность перемирия со злоумышленником

Если администратор компьютерного ресурса имеет противостояние с несколькими злоумышленниками, действующими разрозненно, поодиночке, то мы имеем бескоалиционную игру.

Как известно, в таком случае возможна ситуация равновесия по Нэшу в смешанных стратегиях. Иначе говоря, и администратор и злоумышленники в таком случае какое-то время сохраняют неизменными свои стратегии поведения, и в силу этого каждая сторона имеет устраивающий ее максимальный выигрыш. Все довольны и никто не пытается разрушить установившуюся ситуацию. Имеет ли это какое-либо практическое значение?

Ясно, что подобная ситуация возникает в том случае, когда злоумышленник «пробил» защиту, расположился в сети, но не наносит ущерб хозяину сети. Администратор обнаружил несанкционированное проникновение в сеть, но не знает, как избавиться от «гостя». Более того, администратор может опасаться, что его действия по выпроваживанию гостя

могут спровоцировать гнев злоумышленника, вследствие которого он перейдет к разрушительным действиям. Каковы рекомендации следует дать администратору в данном случае?

Пожалуй, следует напомнить ему Уголовный кодекс РФ:

> **Статья 274**. Нарушение правил эксплуатации средств хранения, обработки или передачи компьютерной информации и информационно-телекоммуникационных сетей
>
> 1. Нарушение правил эксплуатации средств хранения, обработки или передачи охраняемой компьютерной информации либо информационно-телекоммуникационных сетей и оконечного оборудования, а также правил доступа к информационно-телекоммуникационным сетям, повлекшее уничтожение, блокирование, модификацию либо копирование компьютерной информации, причинившее крупный ущерб, – наказывается штрафом в размере до пятисот тысяч рублей или в размере заработной платы или иного дохода осужденного за период до восемнадцати месяцев, либо исправительными работами на срок от шести месяцев до одного года, либо ограничением свободы на срок до двух лет, либо принудительными работами на срок до двух лет, либо лишением свободы на тот же срок.
>
> 2. Деяние, предусмотренное частью первой настоящей статьи, если оно повлекло тяжкие последствия или создало угрозу их наступления, – наказывается принудительными работами на срок до пяти лет либо лишением свободы на тот же срок.

Как видим, «мирное сосуществование» со злоумышленником, – это как минимум нарушение правил доступа к информационно-телекоммуникационным сетям, создающее угрозу наступления тяжких последствий, что влечет привлечение администратора к уголовной ответственности.

Глава 13

Стохастические игры

Стохастические игры были изобретены Л. Шепли в начале 1950-х годов [65].

13.1. Понятие стохастической игры

Стохастическая игра – это многошаговая игра, в которой имеется несколько игровых состояний, а переход от одного состояния к другому совершается с определенной вероятностью. Игроки совершают действия.

В начале каждого шага игра находится в некотором состоянии. Игроки выбирают свои действия и получают выигрыши, зависящие от текущего состояния и действий. После этого система переходит случайным образом в другое состояние, распределение вероятности переходов зависит от предшествующего состояния и действий игроков. Эта процедура повторяется в течение конечного или бесконечного числа шагов.

При конечном числе игроков, конечных множествах действий и состояний игра с конечным числом повторений всегда имеет равновесие Нэша.

На каждом шаге игры предусматриваются выигрыши. В стохастической игре возможны возвращения к предшествую-

щей позиции.

С целью предотвращения бесконечного продолжения игры и бесконечно большого выигрыша вводится правило, по которому задаются такие переходные вероятности, чтобы бесконечное продолжение игры имело вероятность нуль, а математическое ожидание выигрыша было конечным.

Стохастическая игра с двумя игроками – это кортеж

$$(S, A^1, A^2, Q, R^1, R^2, \beta),$$

где

$S = \{s_1 ..., s_N\}$ – множество состояний игры;

$A^k = \{\alpha_1^k, ..., \alpha_{M^k}^k\}$, $k = 1, 2$ – набор действий игрока P_k. Набор действий A_s^k для игрока P_k в состоянии s – это подмножество множества A^k, то есть $A_s^k \subset A^k$ и $\bigcup_{s \in S} A_s^k = A^k$. $M^k = card(A^k) = |A^k|$;

$Q : S \times A^1 \times A^2 \times S \to [0, 1]$ – переходная функция состояний, и $R^1 : S \times A^1 \times A^2 \to \mathbb{R}$, $R^2 : S \times A^1 \times A^2 \to \mathbb{R}$ – выигрышные функции игроков;

β, $0 < \beta \leq 1$ – коэффициент обесценивания (дисконтирования, discount), обесценивающий будущие вознаграждения, то есть, при каждом переходе в новые состояния вознаграждение уменьшается в β раз от его полной стоимости в текущем состоянии.

В игру играют следующим образом:

• В момент дискретного времени $t \in [0, N]$ игра находится в состоянии $s_t \in S$.

• Игрок P_1 выбирает действие $a_t^1 \in A^1$ и игрок P_2 выбирает действие $a_t^2 \in A^2$. Игрок P_1 тогда получает вознаграждение $r_t^1 = R^1(s_t, a_t^1, a_t^2)$, игрок P_2 – вознаграждение $r_t^2 = R^2(s_t, a_t^1, a_t^2)$.

• Игра затем переходит в новое состояние s_{t+1} с условной вероятностью $P(s_{t+1}|s_t, a_t^1, a_t^2)$, равной $Q(s_t, a_t^1, a_t^2, s_{t+1})$.

13.2. Стационарные стратегии

Пусть

$$\Omega^n = \{(p_1, ..., p_n) \in \mathbb{R}^n : \sum_{i=1}^{n} p_1 = 1,\ p_i \geq 0\}.$$

Стационарная стратегия игрока P_k $(k = 1, 2)$ – это отображение

$$p^k : S \to \Omega^{M_k}.$$

Тогда

$$p^k(s) = (p_1^k(s), ..., p_{M^k}^k(s)).$$

Интерпретируем число $p_j^k(s)$ как вероятность того, что, находясь в состоянии s, игрок P_k совершит действие $\alpha_j^k \in A^k$.

Стационарная стратегия игрока P_k независима от времени t и истории.

Смешанная, или рандомизированная, стационарная стратегия – это та стратегия, для которой $p_j^k(s) \geq 0$ для $\forall s \in S$ и $\forall j \in \{1, ..., M^k\}$, и *чистая стратегия* – та, где $p_{j_0}^k(s) = 1$ для некоторого j_0.

13.3. Ожидаемый доход игроков в стохастической игре

Цель каждого игрока – максимизировать некоторый ожидаемый доход. Пусть s_t – состояние во время t и r_t^k – вознаграждение, полученное игроком P_k $(k = 1, 2)$ во время t.

Определим ожидаемый выигрыш как вектор-колонку

$$v_{p^1,p^2}^k = (v_{p^1,p^2}^k(s_1), ..., v_{p^1,p^2}^k(s_N))^T,$$

где

$$v_{p^1,p^2}^k(s) = \mathbf{E}_{p^1,p^2}\{r_t^k + \beta r_{t+1}^k + \beta^2 r_{t+2}^k + \beta^N r_{t+N}^k | s_t = s\} =$$

$$= \mathbf{E}_{p^1,p^2} \left\{ \sum_{n=0}^{N} \beta^n r^k_{t+n} | s_t = s \right\}.$$

Оператор ожидания $\mathbf{E}_{p^1,p^2}$ используется, чтобы показать, что игрок P_k применяет вероятностную стратегию p^k, точнее игрок P_k выбирает действие, используя распределение вероятности $p^k(s_{t+n})$ в s_{t+n} и получает непосредственное вознаграждение

$$r^k_{t+n} = p^1(s_{t+n})^T R^k(s_{t+n}) p^2(s_{t+n})$$

для $n \geq 0$, где

$$R^k(s) = ||R^k(s, a_1, a_2)||_{a_1 \in A^1, a_2 \in A^2}$$

– премиальная матрица игрока P_k в состоянии s, строки и столбцы которой помечены индексами a_1, a_2.

Для игры, бесконечной по времени $N = \infty$ (с бесконечным повторением) принимается $\beta < 1$. Тогда v^k – ожидаемый *дисконтированный выигрыш*. Для конечной по времени игры ($N < \infty$) – $\beta = 1$. Векторы v^k называют также вектор-значением игрока P_k.

13.4. Равновесие Нэша

Равновесие Нэша – это пара стационарных стратегий (p^1_*, p^2_*), для которых

$$v^1_{p^1_*,p^2_*} \geq v^1_{p^1,p^2_*} \quad \text{для} \quad \forall p^1 \in \Omega^{M^1},$$

$$v^2_{p^1_*,p^2_*} \geq v^2_{p^1_*,p^2} \quad \text{для} \quad \forall p^2 \in \Omega^{M^2}$$

покомпонентно.

В равновесии у игроков нет стимула, чтобы отклониться от их стратегий равновесия. Отклонение будет означать, что один или оба игрока будут иметь более низкие ожидаемые выигрыши, то есть $v^1_{p^1,p^2}$ и/или $v^2_{p^1,p^2}$. Пара стратегий, являющихся равновесием Нэша, известны как лучшие выигрыши, т. е. если игрок P_1 играет π^1_*, то лучший ответ для игрока P_2 есть π^2_*, и наоборот.

Теорема 13.1. *Игра с ожидаемым дисконтированным выигрышем имеет хотя бы одно равновесие Нэша в смешанных стационарных стратегиях.* ■

В игре с $N = \infty$ для вычисления равновесия Нэша используется нелинейная программа[1] из [49], которую назовем NLP-1. В случае $N < \infty$ надо воспользоваться программой из [52].

13.5. Программа NLP-1

Равновесие Нэша при $N = \infty$ ищется сведением к задаче нелинейного программирования:

Найти

$$\min_{u^1,u^2,\sigma^1,\sigma^2} \mathbf{1}^T[u^k - R^k(\sigma^1,\sigma^2) - \beta P(\sigma^1,\sigma^2)u^k], \quad k = 1,2 \quad (13.1)$$

при условиях

$$R^1(s_i)\sigma^2(s_i) + \beta T(s_i,u^1)\sigma^2(s_i) \le u^1(s_i)\mathbf{1}, \quad i = 1,...,N, \quad (13.2)$$

$$\sigma^1(s_i)^T R^2(s_i) + \beta\sigma^1(s_i)^T T(s_i,u^2) \le u^2(s_i)\mathbf{1}^T, \quad i = 1,...,N, \quad (13.3)$$

где $u^k \in \mathbb{R}^N$, $\sigma^k \in \Omega^{M^k}$ – переменные векторы, $\mathbf{1}$ – единичный вектор,

$$R^k(\sigma^1,\sigma^2) = [\sigma^1(s_1)^T R^k(s_1)\sigma^2(s_1)...\sigma^1(s_N)^T R^k(s_N)\sigma^2(s_N)]^T$$

– вектор, представляющий выигрыш при выборе игроками P_1 и P_2 пары стратегий (σ^1,σ^2),

$$P(\sigma^1,\sigma^2) = [\sigma^1(s)^T[p(s'|s,a^1,a^2)]_{a^1\in A^1,a^2\in A^2}\sigma^2(s)]_{s,s'\in S}.$$

– стохастическая матрица для марковской цепи, индуцированной парой стратегий (σ^1,σ^2),

$$T(s,u) = [[p(s_1|s,a^1,a^2)...p(s_N|s,a^1,a^2)]^T u^T]_{a^1\in A^1,a^2\in A^2}$$

[1] Имеется метод нелинейного программирования в задачах оптимизации.

– матрица, представляющая выигрыши в будущем в следующем состоянии игры в матричной форме.

Решение $(u_*^1, u_*^2, \sigma_*^1, \sigma_*^2)$ задачи нелинейного программирования (13.1)-(13.3) есть искомое равновесие Нэша $(v_*^1, v_*^2, p_*^1, p_*^2)$ в игре.

13.6. Ллойд Стауэлл Шепли

Стохастические игры созданы американским экономистом и математиком Л. Шепли в начале 1950-х годов.

Ллойд Стауэлл Шепли (Lloyd Stowell Shapley) родился 2 июня 1923 года в Кембридже, шт. Массачусетс (США).

Образование: Гарвадский (1948) и Принстонский университеты. Защитил диссертацию в Принстоне (Ph.D. in Mathematics, 1953) на тему «Additive and Nonadditive Set Functions» под руководством Альберта Такера.

Л.С. Шепли

Лауреат нобелевской премии по экономике 2012 года совместно с американским экономистом Элвином Ротом – за вклад в теорию устойчивого распределения и практику моделирования рынка.

Служил в Военно-воздушных силах США (1943–1945). Бакалавр (1948) Гарвардского университета; доктор философии (1953) Принстонского университета. Работал в корпорации RAND (1948–1949; 1954–1981), с 1981 года преподает в Калифорнийском университете (Лос-Анджелес).

Награжден боевой медалью Бронзовая звезда (1944). Академик Американской академии искусств и наук (с 1974) и Национальной академии наук США (с 1979). Премия имени Джона фон Неймана (1981). Почетный член Американской экономической ассоциации (с 2007). Почетный доктор Hebrew

University of Jerusalem (1986).

13.7. Альберт Уильям Такер

Жизнь трех создателей современной теории игр – Нэша, Куна, Шепли – пересекалась с замечательным математиком А.У. Такером.

А.У. Такер

Альберт Уильям Такер (28 ноября 1905 – 25 января 1995) канадский математик, внесший важный вклад в топологию, теорию игр и нелинейное программирование.

Альберт Такер родился в Ошаве (Онтарио, Канада). Получил степень бакалавра в Университете Торонто в 1928, а степень магистра – в 1929 году. В 1932 году он защитил диссертацию (PhD) в Принстонском университете под руководством тополога Соломона Лефшеца на тему «Абстрактное описание многообразий» (Abstract Approach to Manifolds).

В 1932–33 годах Такер – научный сотрудник Кембриджа, Гарварда и университета Чикаго. Затем он вернулся в Принстон, где пробыл до 1974 года. Такер возглавлял факультет математики в течение приблизительно двадцати лет. Среди его защитившихся аспирантов два нобелевских лауреата по экономике – Нэш и Шепли, и филдсовский лауреат Дж. Милнор.

Такер – один из авторов хорошо известных условий Каруша–Куна–Такера, являющихся базовым результатом в нелинейном программировании.

Умер 25 января 1995 года (в 89 лет) в Highstown, Нью-Джерси, США.

Глава 14

Анализ безопасности компьютерной сети

Покажем, как теоретико-игровые методы позволяют проводить анализ безопасности компьютерной сети[1].

Будет продемонстрирована игровая модель атаки и защиты компьютерной сети, изображенной на рис. 14.1.

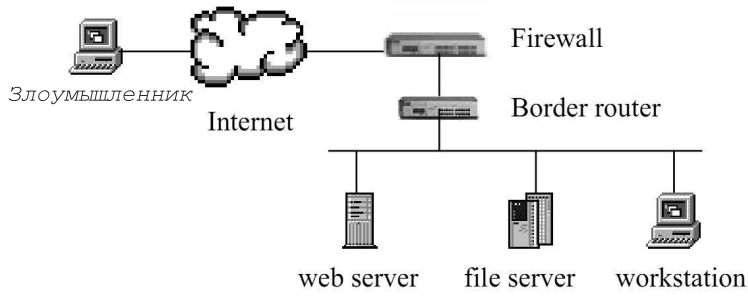


Рис. 14.1. Пример сети

Основой моделирования является теория стохастических

[1]Использована статья Kong-wei Lye и Jeannette Wing [59].

игр.

В нашем примере даны два представления об игре: точка зрения злоумышленника (рис. 14.3) и точка зрениия администратора (рис. 14.4). Эти рисунки будут описаны подробно в § 14.1.3.

14.1. Описание игры

Состояния компьютерной сети могут означать констатацию различных типов хранящейся информации или особенностей, относящихся к аппаратным средствам, к программному обеспечению, к возможностям соединения, к пользовательским привилегиям и т. д. Использование большего количества функций при задании состояния позволяет нам представлять сеть лучше, но чем больше деталей, тем более сложным становится анализ безопасности сети.

Рассматриваем компьютерную сеть в виде графа, изображенного на рис. 14.2. Вершины графа являются физическими объектами, такими как компьютер злоумышленника (вершина E), компьютер (workstation) в сети (вершина N), веб-сервер (вершина W), файл-сервер (вершина F). Ребра графа представляют пути непосредственной связи (физической или беспроподной). Например, внешний компьютер (узел E) имеет прямой доступ только к веб-серверу.

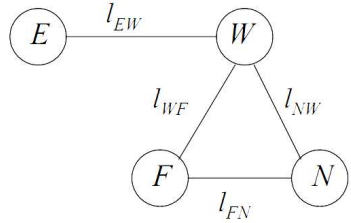


Рис. 14.2. Сеть как граф

14.1.1. Состояния игры

Пусть

$$P \subset \{ftpd, httpd, nsfd, process, sniffer, viruses\_anti\}$$

– перечень установленного программного обеспечения, под process понимаем запущенный пользователем процесс,

$$a \in \{u, c\}$$

– переменная, представляющая состояние учетной записи (account) пользователей, u – нет учетных записей под угрозой, c – хотя бы одна учетная запись под угрозой,

$$d \in \{c, i\}$$

– представляет состояние данных в вершине; c – данные повреждены или украдены, i – данные не повреждены или не украдены,

Вводим состояния сети

$$S = \{n_W, n_F, n_N, t\},$$

где

$$n_X = (P, a, d), \quad X \in \{W, F, N, E\},$$

и t – состояние трафика сети в целом.

Следовательно, например, если

$$n_W = (\{ftpd, httpd, sniffer\}, c, i),$$

то это говорит, что на веб-сервере запущены программы ftpd, httpd, sniffer, некоторые учетные записи находятся под угрозой, но данные еще не повреждены или не украдены.

Движение информации (traffic) для сети в целом представляется состоянием трафика $t =< \{l_{XY}\} >$, где X и Y – вершины графа сети и $l_{XY} \in \{0, 1/3, 2/3, 1\}$ указывает на наличие передачи данных по этому каналу. Цифра 1 говорит о задействовании максимума возможностей.

Например, при соединении, основанном на стандарте 10Base-T, цифры 0, 1/3, 2/3 и 1 означают 0Mbps, 3.3Mbps, 6.7Mbps и 10Mbps соответственно. Для сети, представленной графом на рис. 14.2, состояние трафика описывается как $t =< l_{EW}, l_{WF}, l_{FN}, l_{NW} >$. Нормальному трафику этой сети соответствует состояние $t =< 1/3, 1/3, 1/3, 1/3 >$.

Потенциальная мощность числа состояний для нашей сети огромна:

$$|S| = |n_W| \cdot |n_F| \cdot |n_N| \cdot |t| =$$

$$= (63 \cdot 2 \cdot 2)^3 \cdot 4^4 = 4 \text{ млн состояний},$$

но только 18 (15 на рис. 14.3 и три дополнительных на рис. 14.4) относится к нашему примеру. На этих рисунках каждое состояние представлено с использованием прямоугольника с символическим именем состояния и значениями параметров состояния. Для удобства мы будем обращаться к состояниям, используя их символические имена.

14.1.2. Действия игры

Действия злоумышленника и администратора заставляют сеть переходить из одного состояния в другое с определенной вероятностью.

Отдельно взятое действие злоумышленника может быть любой частью из его стратегии нападения, такой как flooding (падение) сервера посредством посылки пакетов SYN или загрузкой файла пароля. Когда игрок ничего не делает, мы обозначаем это бездействие как $\emptyset$.

Совокупность действий злоумышленника состоит из всех действий, которые он может совершить во всех состояниях:

$$A^{Attacker} =$$

$$
\begin{aligned}
=\{&Attack\_httpd, Attack\_ftpd, Continue\_hacking,\\
&Deface\_website\_leave, Install\_sniffer,\\
&Run\_DoS\_virus,\\
&Crack\_file\_server\_root\_password,\\
&Crack\_workstation\_root\_password,\\
&Capture\_data, Shutdown\_network, \emptyset\}.
\end{aligned}
\tag{14.1}
$$

Действия злоумышленника в каждом состоянии – это подмножество множества $A^{Attacker}$. Например, в состоянии **Normal operation** (см. рис. 14.3, самое верхнее (topmost) состояние), злоумышленник совершает множество действий

$$A^{Attacker}_{Normal\_operation} = \{Attack\_httpd, Attack\_ftpd, \emptyset\}.$$

Действия для администратора главным образом сводятся к профилактическим или восстановительным мерам. Прежде чем привести полное множество его действий, напомним, что слово «compromised» переводится как «поставленный под угрозу». Итак, множество действий администратора таково:

$$
\begin{aligned}
&A^{Admin} =\\
=\{&Remove\_compromised\_account\_restart\_httpd,\\
&Restore\_website\_remove\_compromised\_account,\\
&Remove\_virus\_compromised\_account,\\
&Install\_sniffer\_detector, Remove\_sniffer\_detector,\\
&Remove\_compromised\_account\_restart\_ftpd,\\
&Remove\_compromised\_account\_sniffer, \emptyset\}
\end{aligned}
\tag{14.2}
$$

Например, в состоянии **Ftpd_attacked** (см. рис. 14.4) администратор имеет множество действий

$$A^{Admin}_{Ftpd\_attacked} = \{install\_sniffer\_detector, \emptyset\}.$$

Узел с находящимся под угрозой account'ом может быть попавшим в поле внимания администратора, а может быть и незамеченным. Когда он не замечен, мы моделируем ситуацию как ситуацию нахождения администратора в состоянии

бездействия $\emptyset$. Мы предполагаем, что администратор не знает, есть факт нападения или нет. Следует учитывать, что злоумышленник может иметь несколько целей и стратегий, о которых не знает администратор. Более того, не все действия нападающего могут наблюдаться.

14.1.3. Вероятности переходов

В изучаемом примере компьютерной сети значения для вероятностей изменения состояния сети даем, основываясь на собственной интуиции.

Для реальных сетей необходимые вероятности следует находить, естественно, используя дополнительные исследования и накапливая необходимую статистику. На рис. 14.3 и 14.4 изменения состояния сети представлены стрелами. Каждая стрела маркирована действием, переходной вероятностью и стоимостью/вознаграждением.

В формальной модели игры вероятность изменения состояния является функцией действий обоих игроков. Такие вероятности используются в компьютерной нелинейной программе NLP-1, применяемой для вычисления решения игры. Однако, чтобы реализовать разделения игры на игру с точки зрения злоумышленника (рис. 14.3) и с точки зрения администратора (рис. 14.4) принимается, что вероятности зависят от действий каждого игрока в отдельности. Например, на рис. 14.3 (вторая пунктирная стрела из вершины) считаем, что $P(\mathbf{Ftpd\_hacked}|\mathbf{Ftpd\_attacked}, Continue\_attacking) = 0,5$ как зависящую только от действия злоумышленника $Continue\_attacking$.

Когда сеть находится в состоянии **Normal_operation** и ни нападавший, ни администратор не принимают мер, сеть будет иметь тенденцию оставаться в том же самом состоянии. Эта ситуация моделируется как имеющая близкую к тождественной стохастическую матрицу, т. е. мы полагаем $P(\mathbf{Normal\_operation} \mid \mathbf{Normal\_operation}, \emptyset, \emptyset) = 1 - \varepsilon$ для некоторого малого $\varepsilon < 0,5$ и где $\emptyset$ обозначает бездействие. Тогда $P(s|\mathbf{Normal\_operation}, \emptyset, \emptyset) = \varepsilon/(N-1)$ для всяко-

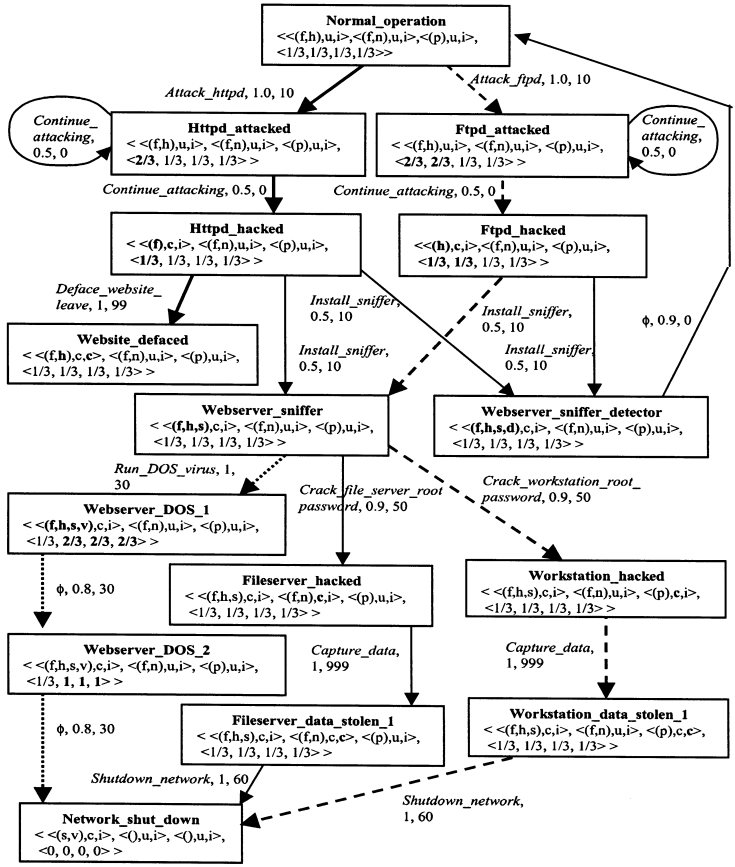


Рис. 14.3. Игра с точки зрения злоумышленника

го $s \neq$ **Normal_operation**, где N – число состояний. Есть также смены состояния, которые являются неосуществимыми. Например, для сети невозможно переместиться от нормального функционирования к состоянию завершения работы, не проходя при этом через некоторые промежуточные состояния.

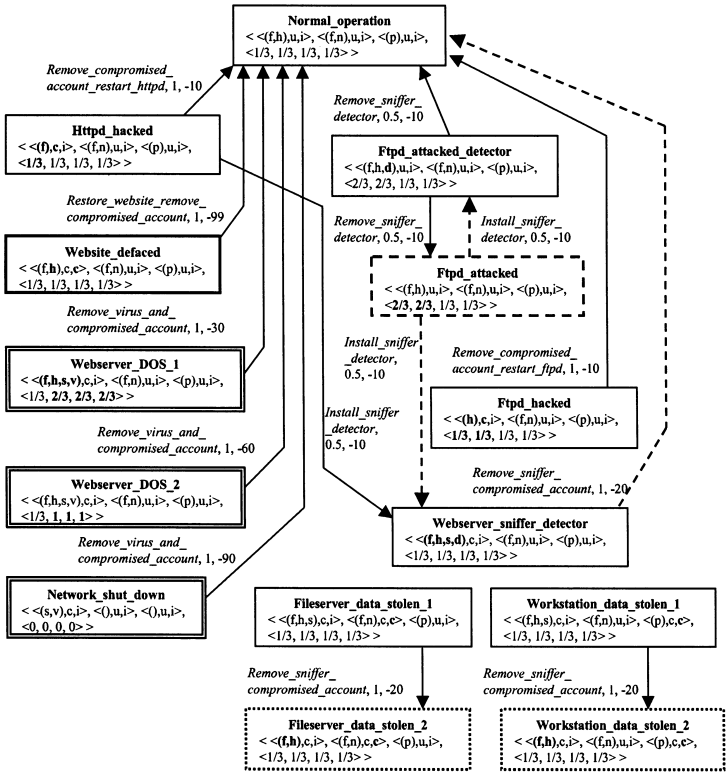


Рис. 14.4. Игра с точки зрения администратора

Неосуществимым сменам состояния приписываются нулевые вероятности перехода.

14.1.4. Платежи, затраты и вознаграждения

Затраты (отрицательные значения) и вознаграждения (положительные значения) связаны с действиями злоумышленника и администратора. Для действий злоумышленника имеем,

главным образом, вознаграждения, и такие вознаграждения выражаются в терминах количества повреждений, которые он нанес компьютерной сети.

Некоторые затраты, однако, трудно оценить количественно. Например, потеря информации о маркетинговой стратегии конкуренту может вызвать крупные денежно-кредитные потери. Стертый корпоративный веб-сайт может нанести ущерб репутации компании и привести к потери части своих клиентов

В рассматриваемой модели ограничиваемся временными затратами администратора на восстановительные работы. Вознаграждение за действия злоумышленника также можно оценивать временем, которое необходимо администратору на восстановление поврежденной службы или данных, т. е. на перевод одного состояния сети в другое.

Например, когда обрушен некоторый сервис, администратору могут потребоваться 10 или 15 минут времени, чтобы определить причину и перезапустить сервис. В рис. 14.4 администратору нужно 10 минут, чтобы удалить оказавшуюся под угрозой учетную запись пользователя и перезапускать httpd (из состояния **Httpd_hacked** перейти к состоянию **Normal_operation**). Для нападавшего это количество времени было бы его вознаграждением.

Чтобы подчеркнуть серьезность потери важных финансовых данных в рассматриваемом примере сети, назначается очень высокое вознаграждение за действие нападавшего, которое приводит к состоянию, в котором злоумышленник получает эти данные. Например, при переходе из состояния **Workstation_hacked** к состоянию **Workstation_data_stolen_1** на рис. 14.3 вознаграждение равно 999. Есть также некоторые переходы, в которых потери для администратора не такие же как величина вознаграждения злоумышленника. Именно такие переходы делают игру игрой с общей суммой вместо игры с нулевым суммой.

14.1.5. Стратегии

Стационарные стратегии описаны в § 12.2. Наша цель – найти смешанные стационарные равновесия Нэша.

В рассматриваемом примере компьютерной сети p^1, p^2 соответствуют стратегиям злоумышленника и администратора, $v^1_{p^1,p^2}$ и $v^2_{p^1,p^2}$ – это их ожидаемые вознаграждения.

Стохастическая игра с общей суммой имеет хотя бы одно равновесие Нэша в смешанных стационарных стратегиях. Для их нахождения использована программа NLP-1 [49].

14.2. Атаки и защита сети

В этом параграфе опишем один из сценариев нападения и защиты компьютерной сети. Два других можно посмотреть в [59].

На рис. 14.3 показано, как злоумышленник видит изменения состояний сети в результате его действий, а на рис. 14.4 представлена точка зрения администратора. На рисунках состояния изображаются как прямоугольники, содержащие символическое имя и значения параметров для этого состояния. Каждый переход помечен действием, вероятностью перехода, выгодой или стоимостью в минутах восстановительных усилий, затрачиваемых администратором в случае повреждений в сети.

Атакуется веб-сервер сети. На веб-сервере, как правило, запущен httpd и ftpd. Нападающий завладевает root[2] shell, используя прием переполнения буфера (bufer overflow). Как только нападавший получает root shell, он может стереть вебсайт и уйти. Для данного сценария изменения состояния обозначены жирными стрелками на рис. 14.3.

В состоянии **Normal_operation** злоумышленник осуществляет действие *Attack_httpd*. С вероятностью 1

[2]Root – это учетная запись администратора, или superusera. Получив доступ к этому профилю, вы имеете целый ряд возможностей, не доступных в обычном режиме работы.

Равновесия Нэша

Таблица 14.1

		Strategies		State Values	
	State	Attacker	Administrator	Attacker	Administrator
1	**Normal_operation**	[1.00 0.00 0.00]	[0.33 0.33 0.33]	210.2	-206.8
2	**Httpd_attacked**	[1.00 0.00 0.00]	[0.33 0.33 0.33]	202.2	-191.1
3	**Ftpd_attacked**	[0.65 0.00 0.35]	[1.00 0.00 0.00]	176.9	-189.3
4	**Ftpd_attacked_detector**	[0.40 0.12 0.48]	[0.93 0.07 0.00]	165.8	-173.8
5	**Httpd_hacked**	[0.33 0.10 0.57]	[0.67 0.19 0.14]	197.4	-206.4
6	**Ftpd_hacked**	[0.12 0.00 0.88]	[0.96 0.00 0.04]	204.8	-203.5
7	**Website_defaced**	[0.33 0.33 0.33]	[0.33 0.33 0.33]	80.4	-80.0
8	**Webserver_sniffer**	[0.00 0.50 0.50]	[0.33 0.33 0.34]	716.3	-715.1
9	**Webserver_sniffer_detector**	[0.34 0.33 0.33]	[1.00 0.00 0.00]	148.2	-185.4
10	**Webserver_DOS_1**	[0.33 0.33 0.33]	[1.00 0.00 0.00]	106.7	-106.1
11	**Webserver_DOS_2**	[0.34 0.33 0.33]	[1.00 0.00 0.00]	96.5	-96.0
12	**Network_shut_down**	[0.33 0.33 0.33]	[0.33 0.33 0.33]	80.4	-80.0
13	**Fileserver_hacked**	[1.00 0.00 0.00]	[0.35 0.34 0.31]	1065.5	-1049.2
14	**Fileserver_data_stolen_1**	[1.00 0.00 0.00]	[1.00 0.00 0.00]	94.4	-74.0
15	**Workstation_hacked**	[1.00 0.00 0.00]	[0.31 0.32 0.37]	1065.5	-1049.2
16	**Workstation_data_stolen_1**	[1.00 0.00 0.00]	[1.00 0.00 0.00]	94.4	-74.0
17	**Fileserver_data_stolen_2**	[0.33 0.33 0.33]	[0.33 0.33 0.33]	80.4	-80.0
18	**Workstation_data_stolen_2**	[0.33 0.33 0.33]	[0.33 0.33 0.33]	80.4	-80.0

и вознаграждением 10 он переводит сеть в состоянии **Httpd_attacked**. В результате действия злоумышленника в данном состояние наблюдается увеличенный трафик между компьютером злоумышленника и веб-сервером. Начав действие *Continue_attacking*, злоумышленник с вероятностью успеха 0,5 получает права пользователя или доступ к корню через падение httpd. В результате система переходит в состояние **Httpd_hacked**. Поскольку он получил корневой доступ к веб-серверу, он может повредить веб-сайт, перезапустить httpd и уйти, перемещая сеть в состояние **Website_defaced**.

14.3. Результаты моделирования

Применение программы NLP-1 позволило найти равновесие Нэша для различных сценариев нападения на компьютерную сеть, представленную на рис. 14.1.

Стратегия игрока состоит из распределения вероятности по набору действия для каждого состояния. Результаты даны в таблице 14.1. О чем говорят строчки этой таблицы?

Например, для состояния **Httpd_hacked**

$$p_*^{Attacker} = [0,33; 0,10; 0,57], \quad p_*^{Admin} = [0,67; 0,19; 0,14].$$

Цифры в квадратных скобках программа NLP-1 сопровождает указанием множества соответствующих действий. Для злоумышленника эти действия таковы:

$$\{Deface\_website\_leave, Install\_sniffer, \emptyset\}.$$

Стратегия злоумышленника в рассматриваемом состоянии говорит, что он использует действие $Deface\_website\_leave$ с вероятностью 0,33 и $Install\_sniffer$ с вероятностью 0,10. Игнорируя бездействие $\emptyset$ и нормализуя вероятности, получаем вероятности 0,77 и 0,23 соответственно для первых двух действий. Даже при том, что инсталляция сниффера может позволить ему взломать корневой пароль и своровать любые данные, надо учитывать, что администратор может заметить его

присутствие и принять защитные меры. Он поэтому нанесет больший ущерб, если просто сотрет webserver и уйдет.

В том же самом состоянии администратор может совершить действие

$$Remove\_compromised\_account\_restart\_httpd$$

или

$$Install\_sniffer\_detector.$$

Первое действие он совершает с вероятностью 0,67, второе – с вероятностью 0,19. Игнорируя третье действие, после нормализации эти вероятности принимают значения 0,88 и 0,22 соответственно. Это говорит ему, что лучше немедленно удалить подозрительную учетную запись (account), перезапустить httpd, чем продолжать «играть» со злоумышленником.

Таким образом, использование стохастических игр способно быть серьезным инструментом, с помощью которого можно проводить анализ степени безопасности той или иной компьютерной системы, моделировать сценарии различных атак на информационные ресурсы и вырабатывать рекомендации по их защите.

Заключение

Теория игр возникла с целью решения экономических задач. С 1950-х гг. ее начинают применять не только в экономике, но и в кибернетике, технике и биологии. После Второй мировой войны теорией игр заинтересовались военные ведомства, которые увидели в ней мощный аппарат для исследования стратегических решений.

Успехи теории игр в экономике можно оценивать по числу нобелевских лауреатов. Так, нобелевскими лауреатами по экономике за достижения в области теории игр и экономической теории стали Роберт Ауманн, Райнхард Зелтен, Джон Нэш, Джон Харсаньи, Уильям Викри, Джеймс Миррлис, Томас Шеллинг, Джордж Акерлоф, Майкл Спенс, Джозеф Стиглиц, Леонид Гурвиц, Эрик Мэскин, Роджер Майерсон, Ллойд Шепли, Элвин Рот.

Однако такое количество нобелевских премий в области экономики на фоне постоянных экономических и финансовых кризисов несколько удивляет и бросает тень на действенную эффективность применения теории игр в экономике. Тем не менее теория игр преподается практически во всех ведущих университетах мира при обучении студентов-экономистов. Количество соответствующих публикаций статей и учебников по теории игр, касающихся экономики, постоянно растёт.

В то же время отслеживание публикаций с помощью Интернет показывает, что, хотя публикаций, применяющих теорию игр в сфере защите компьютерных систем, не столь много,

всё же достаточно интенсивно ведут исследования по защите компьютерных систем с помощью теории игр специалисты быстро развивающихся стран, среди которых Китай и Индия. Заметны такие публикации и в США. Намного хуже ситуация в России.

Думается, что успехи теории игр в сфере защиты компьютерных систем скорее всего скрыты от любопытствующего гражданина. Это связано с тем, что ведущие государства, принимая решения по созданию кибервойск и издавая указы, ставящие задачи по обеспечению кибербезопасности своих информационных ресурсов, к которым относятся в первую очередь информационные системы и информационно-телекоммуникационные сети, находящиеся на территории страны и в дипломатических представительствах за рубежом, ограничивают доступ рядовых граждан к передовым научным разработкам в этой области.

Авторы надеются, что представленное учебное пособие будет способствовать распространению в России идей, касающихся организации защиты компьютерных систем с помощью теории игр.

Список литературы

[1] *Абденов А.Ж.* Защита информации в информационных системах // Информационно-телекоммуникационные системы: сб. матер. семинара по повышению квалификации молодых ученых. – Новосибирск: НГТУ, 2008. С. 7–38.

[2] *Абденов А.Ж., Заркумова Р.Н.* Выбор средства эффективной защиты с помощью методов теории игр // Вопросы защиты информации. 2010. №2. С. 26–31.

[3] *Аръков П.А.* Комплекс моделей для поиска оптимального проекта системы защиты информации // Известия Южного федерального университета. Технические науки. 2008. Т.85, №8. С. 30–36.

[4] *Брёкер Т., Ландер Л.* Дифференцируемые ростки и катастрофы. М. : Мир, 1977.

[5] *Васин А.А., Морозов В.В.* Введение в теорию игр с приложениями в экономике. М., 2003. 278 с.

[6] *Вахний Т.В., Гуц А.К.* Теоретико-игровое моделирование поведения азартного нарушителя при защите информационных ресурсов // Межвузовская научно-практическая конференция «Информационные технологии и автоматизация управления». Омск: ОмГТУ, 2009. С. 166–167.

[7] *Вахний Т.В., Гуц А.К.* Теоретико-игровой подход к выбору оптимальных стратегий защиты информационных ресурсов // Математические структуры и моделирование. 2009. Вып. 19. С. 104–107.

[8] *Вахний Т.В., Гуц А.К., Константинов В.В.* Программа, реализующая игровой подход при выборе оптимального набора средств защиты компьютерной системы // Математические структуры и моделирование. 2011. Вып. 24. С. 98–101.

[9] *Вахний Т.В., Гуц А.К., Константинов В.В.* Программное приложение для выбора оптимального набора средств защиты компьютерной информации на основе теории игр // Вестник Омского университета. 2013. Вып. 4 (70). С. 201–206.

[10] *Воробьев А.А.* Методы оценивания и обеспечения гарантированного уровня защиты информации от несанкционированного доступа в вычислительной сети автоматизированной системы управления: автореф. дис... канд. техн. наук / А.А. Воробьев. СПб., 1997. 15 с.

[11] *Воробьев А.А., Куликов Г.В., Непомнящих А.В.* Оценивание защищенности автоматизированных систем на основе методов теории игр // Информационные технологии. 2007. №2. С. 1–24.

[12] *Воробьев А.А.* Теоретико-игровой подход к оцениванию качества системы защиты информации от несанкционированного доступа в автоматизированных системах // Информатика–машиностроение. 1999. №3.

[13] *Воробьев А.А.* Анализ моделей процессов защиты информации от несанкционированного доступа в автоматизированных системах // Информатика–машиностроение. 1999. №2.

[14] *Воробьев Н.Н.* Основы теории игр. Бескоалиционные игры. М. : Наука, 1984.

[15] *Воробьев Н.Н.* Теория игр для экономистов-кибернетиков. М. : ФМ, 1985.

[16] *Гаценко О.Ю.* Защита информации. СПб. : Сентябрь, 2001.

[17] *Гроздов Г.Г.* Методика построения модели распределенной атаки на автоматизированную систему // Науково-практичний журнал. Сучасна спеціальна техніка. 2009. №3. С. 82–90.

[18] *Гроздов Г.Г.* Модифицированный способ решения задачи формирования эффективной комплексной системы защиты информации автоматизированной системы : монография. К. : ГУИКТ, 2009. 32 с.

[19] *Гроздов Г.Г.* Методика построения теста на проникновение в автоматизированную систему, основанная на математической теории игр // Наукові записки українського науково-дослідного інституту зв'язку. 2010. №3. С. 88–94.

[20] *Гроздов Г.Г.* Стратегии построения эффективной комплексной системы защиты информации в автоматизированных системах // Наукові праці ДонНТУ. 2011. Вип.14(188). Серія «Інформатика, кібернетика та обчислювальна техніка». С. 24–30.

[21] *Гурвич В.А.* Разрешимость позиционных игр в чистых стратегиях // Журнал вычислительной математики и математической физики. 1975. Т. 15, №2. С. 358–371.

[22] *Гуц А.К., Лавров Д.Н.* Описание DDoS-атаки с помощью катастрофы «сборка» // Математические структуры и моделирование. 2013. Вып.27. С. 42–45.

[23] *Дрешер М.* Стратегические игры. Теория и приложения. М. : Советское радио, 1964.

[24] *Дюбин Г.Н., Суздаль В.Г.* Введение в прикладную теорию игр. М. : Наука, 1981.

[25] *Дятчин В.В., Тутубалин П.И., Бормотов К.В.* Применение теоретико-игровой модели для размещения конфиденциальной информации на серверах корпоративной информационной системы // Исследования по информатике. 2007. Т. 11. С. 89–94.

[26] *Завгородний В.И.* Комплексная защита информации в компьютерных системах: учебное пособие. М. : Логос; ПБОЮЛ Н.А. Егоров, 2001.

[27] *Заркумова Р.Н.* Применение методов теории игр при выборе средства эффективной защиты // Сб. науч. тр. НГТУ. 2009. №4(58). С. 41–46.

[28] *Захаров А.И., Лидский Э.А., Михалева У.А.* Качественные решения при выборе атаки/защиты информационного ресурса // Надежность. 2005. №3. С.12-20.

[29] *Кун Г.У.* Позиционные игры и проблема информации // Сб. : Позиционные игры. М: ФМ, 1967. С. 13–40.

[30] *Лукацкий А.В.* Обнаружение атак. СПб : BHV, 2001. 611 с.

[31] *Льюс Р.Д., Райфа Х.* Игры и решения. М. : Иностранная литература, 1961.

[32] *Мак-Кинси.Д.* Введение в теорию игр. К. : КВИРТУ, 1959. 347 с.

[33] Матричные игры / под. ред. Н.Н. Воробьева. М : Гос. изд-во физ.-мат. лит-ры, 1961. 280 с.

[34] *Медведев Н.В., Гришин Г.А.* Оптимизация тактики защиты компьютерных сетей с использованием математического аппарата теории стратегических игр // Вестник Московского государственного технического университета им. Н.Э. Баумана. Серия : Приборостроение. 2010. №4. С. 117–124.

[35] *Михалева У.А.* Оценка уязвимостей в программном обеспечении организации на основе смешанных стратегий теории игр : автореф. дис... канд. техн. наук. Уфа., 2010. 17 с.

[36] *Моисеев В.С., Козар В.В., Тутубалин П.И., Бормотов К.В.* Двухкритериальная теоретико-игровая модель с заданным упорядочиванием смешанных стратегий // Вестник КГТУ. 2005. №1 С. 40–45.

[37] *Моисеев Н.Н.* Математические задачи системного анализа. М. : Наука, 1981.

[38] *Нейман Дж. фон., Моргенштерн О.* Теория игр и экономическое поведение. М. : Наука, 1970. 983 с.

[39] *Нестеров С.А.* Разработка методов и средств проектирования инфраструктуры обеспечения информационной безопасности автоматизированных систем: автореф. дис... канд. техн. наук. СПб., 2002. 18 с.

[40] *Нестеров С.А.* Об использовани иконечных игровых моделей для оценки экономической эффективности систем защиты информации // Тр. науч.-техн. конф. «Безопасность информационных технологий». 2001. Т. 1. С. 31–33.

[41] *Оуэн Г.* Теория игр. М. : Наука, 1971.

[42] *Петросян Л.А. и др.* Теория игр : учеб. пособие для ун-тов / Л.А. Петросян, Н.А. Зенкевич, Е.А. Семина. М. : Высш. шк.; Книжный дом «Университет», 1998. 304 с.

[43] *Писарук Н.Н.* Введение в теорию игр. Минск : БГУ, 2013. 233 с.

[44] Позиционные игры / под. ред. Н.Н. Воробьева и И.Н. Врублевской. М : Гос. изд-во физ.-мат. лит-ры, 1967. 522 с.

[45] *Редькина Т.В., Сурнева О.Б.* Использование теории игр в практике защиты информации // Передача, прием, обработка и отображение информации о быстропротекающих процессах: сб. ст. М. : РПА «АПР», 2009. С. 382–386.

[46] *Суздаль В.Г.* Теория игр для флота. М. : Военноиздат, 1976. 317 с.

[47] *Alpcan T., Basar T.* Network security: a decision and game-theoretic approach. Cambridge/New York: Cambridge University Press, 2011.

[48] *Beckery S., Seibert J., Zage D., Nita-Rotaru C., State R.* Applying Game Theory to Analyze Attacks and Defenses in Virtual Coordinate Systems.
URL: http://www.cs.purdue.edu/homes/jcseiber/dsn2011_ game-theory.pdf

[49] *Filar J., Vrieze K.* Competitive Markov Decision Processes. Springer-Verlag, 1997.

[50] *Dayanand K., Magesh S.* Defence Strategy against Flooding Attacks Using Nash Equilibrium Game Theory // International Conference on Computing and Control Engineering (ICCCE 2012), 12 & 13 April, 2012. URL: http://www.iccce.co.in/Papers/ICCCECS100.pdf

[51] Decision and Game Theory for Security / Eds. Jens Grossklags Jean Walrand. Third International Conference, GameSec 2012 Budapest, Hungary, November 5-6, 2012 Proceedings, Springer, 2012.

[52] *Fudenberg D., Tirole J.* Game Theory. MIT Press, 1991.

[53] *Gordon L.A., Loeb M.P, Lucyshyn W. et l.* 2006 CSI/FBI computer crime and security survey. San Francisco: Computer Security Institute, 2006.

[54] *Gueye A.* A Game Theoretical Approach to Communication Security. A dissertation submitted in partial satisfaction of the requirements for the degree of Doctor of Philosophy. University of California, Berkeley, 2011.

[55] *Hamilton S.N., Miller W.L., Ott A., Saydjari O.S.* Challenges in applying game theory to the domain of information warfare // Proceedings of the Fourth Information Survivability Workshop, 2002.

[56] *Hamilton S.N., Miller W.L., Ott A., Saydjari O.S.* The role of game theory in information warfare // Proceedings of the Fourth Information Survivability Workshop, 2002.

[57] *Khirwadkar T.* Defense against network attacks using game theory // Thesis for the degree of Master of Science in Computer Science. Graduate College of the University of Illinois at Urbana-Champaign, Urbana, Illinois, 2011. 66 p.

[58] *Kuhn H.W.* Extensive Games and the Problem of Information / In Contributions to the Theory of Games. – Eds. Harold W. Kuhn and A. Tucker, Vol. 2, Princeton University Press, 1953. P. 193–216.

[59] *Lye, Kong-wei and Wing J.* Game Strategies in Network Security // School of Computer Science, Carnegie Mellon University, Pittsburgh, 2002. 14 p.

[60] *Lye, Kong-wei and Wing J.* Game Strategies in Network Security // International Journal of Information Security. 2005. V. 4. No. 1-2. P. 71-86.

[61] *McKelvey R.D., McLennan A.M., Turocy T.L.* Gambit: Software tools for game theory. Technical report, Version 0.2007.01.30, 2006.

[62] *Marn-Ling Shing, Chen-Chi Shing, Kuo Lane Chen, Huei Lee.* A Game Theory Approach in Information Security Risk Study // 2010 International Conference on E-business, Management and Economics IPEDR. 2011. V.3. P. 201-203.

[63] *Mendelson E.* Introducing Game Theory and Its Applications. New York: Chapman & Hall/CRC Press Co, 2004.

[64] *Nochenson A., Heimann C.F.L.* Simulation and Game-Theoretic Analysis of an Attacker-Defender Game // Decision and Game Theory for Security Third International Conference, GameSec 2012 Budapest, Hungary, November 5-6, 2012, Proceedings. P. 138-151.

[65] *Shapley L.S.* Stochastic games // Proc. Nat. Acad. Science. 1953. V.39. P. 1095-1100.

[66] *Shun-Chieh Lin, Shian-Shyong Tseng.* Constructing detection knowledge for DDoS intrusion tolerance // Expert Systems with Applications. 2004. V. 27, Issue 3. P. 379–390.

Учебное издание

Александр Константинович Гуц
Татьяна Владимировна Вахний

Теория игр и защита компьютерных систем

Сертификат соответствия № РОСС RU.AE88.H01449
Срок действия с 26.07.2012 г. по 25.07.2015 г.

Редактор Л.М. Кицина
Технический редактор Н.С. Серопян

Подписано в печать 30.12.2013. Формат 60 × 84 1/16.
Печ. л. 10,0. Усл. печ. л. 9,3. Уч.-изд. л. 8,3. Тираж 150 экз.
Заказ 302.

Издательство Омского государственного университета
644077, Омск-77, пр. Мира, 55а
Отпечатано на полиграфической базе ОмГУ
644077, Омск-77, пр. Мира, 55а