Уфа : УГАТУ, 2008 Вестник УГАТУ • Управление, ВТ и И Т. 10, № 2 (27). С. 210–214

УДК 004.7.056

М. С. ПОЛИТОВ, А. В. МЕЛЬНИКОВ

ДВУХФАКТОРНАЯ ОЦЕНКА ЗАЩИЩЕННОСТИ ИС

Сделана попытка найти наиболее эффективную методику и систему критериев для оценки уровня защищенности информационных систем. Проведен анализ существующих и успешно применяющихся сегодня критериев с выявлением их преимуществ и проблемных аспектов. Сформулирована новая система критериев оценивания уровня защищенности. *Информационная система ; критерии оценки уровня защищенности*

При планировании и развертывании информационных систем перед специалистами ставится задача создания не просто системы, которая выполняла бы определенные функции, но и накладывается ряд существенных требований, таких как надежность и сохранность, доверенных системе данных, эквивалентная стоимость которых часто превышает стоимость самой системы. На сегодняшний день актуальность необходимости защиты информации неоспорима, т. е. любая спроектированная и сданная в эксплуатацию система должна нести в себе функции защиты и предотвращения несанкционированного доступа. Очевидно, что защита информации должна носить комплексный характер, но также необходимо учитывать и возможность возникновения (или не возникновения) угроз, специфичных для данной конкретной информационной системы [4]. На этом этапе анализа важно не упустить существенных деталей и, в то же время, не переоценить некоторые из них, ибо это может повлечь неоправданные финансовые и материальные расходы на организацию системы предотвращения возникновения подобных ситуаций. Приступая к созданию системы информационной безопасности необходимо оценить, какие угрозы наиболее актуальны [3].

ТИПОВЫЕ ПОДХОДЫ К АНАЛИЗУ ЗАЩИЩЕННОСТИ

В настоящее время, видимо, не существует каких-либо стандартизированных методик анализа защищенности АС. Поэтому в конкретных ситуациях алгоритмы действий аудиторов существенно различаются. Однако типовую методику анализа защищенности корпоративной сети на сегодня предложить все-таки возможно. И хотя данная методика не претендует на всеобщность, ее эффективность многократно проверена на практике.

Типовая методика включает использование следующих методов:

- Изучение исходных данных по АС;
- Оценка рисков, связанных с осуществлением угроз безопасности в отношении ресурсов АС;
- Анализ механизмов безопасности организационного уровня, политики безопасности организации и организационно-распорядительной документации по обеспечению режима информационной безопасности и оценка их соответствия требованиям существующих нормативных документов, а также их адекватности существующим рискам;
- Ручной анализ конфигурационных файлов маршрутизаторов, МЭ и прокси-серверов, осуществляющих управление межсетевыми взаимодействиями, почтовых и DNS серверов, а также других критических элементов сетевой инфраструктуры;
- Сканирование внешних сетевых адресов ЛВС из сети Интернет;
- Сканирование ресурсов ЛВС изнутри;
- Анализ конфигурации серверов и рабочих станций ЛВС при помощи специализированных программных агентов.

Перечисленные методы исследования предполагают использование как активного, так и пассивного тестирования системы защиты. Активное тестирование системы защиты заключается в эмуляции действий потенциального злоумышленника по преодолению механизмов защиты. Пассивное тестирование предполагает анализ конфигурации ОС и приложений по шаблонам с использованием списков проверки. Тестирование может производиться либо вручную, либо с использованием специализированных программных

средств. Но здесь возникает проблема выбора и сравнения. Используя активные и пассивные методы тестирования, как по результатам оценить или сравнить уровни защищенности (уязвимости) разных конфигураций АС? Необходима некоторая, абстрагированная от конкретных свойств системы, шкала, в рамках которой и будет измеряться общий уровень безопасности. Как вариант предлагается использовать метод аналитической оценки и прогнозирования общего уровня защищенности, описанный в [1]. Данный метод позволяет оценить уровень защиты отдельных элементов АС.

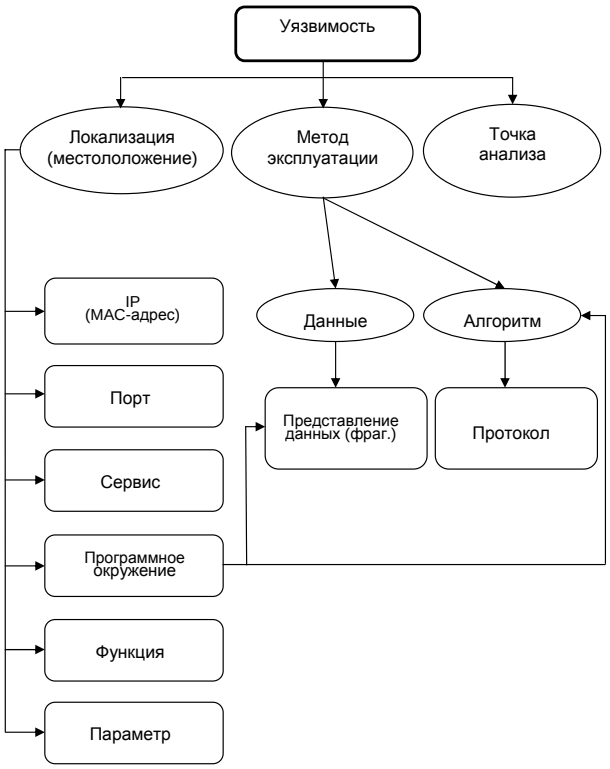

Рис. 1. Модель уязвимости компьютерной системы

КОМПЛЕКСНОЕ ОЦЕНИВАНИЕ СИСТЕМ

Но любая АС состоит из множества подсистем, которые могут иметь абсолютно разные уровни защиты. Рассмотрим критерии и методы совокупной оценки, но прежде введем понятие **уязвимости системы**, комбинируя определения методик аудита безопасности и систем оценивания рисков:

• **Уязвимость любой системы** определяется уязвимостью значимых ресурсов этой системы.

Но что такое уязвимость или неуязвимость системы? Сказать, что эта система уязвима — значит ничего не сказать по сути, ибо нельзя, оценивая плотность вещества, назвать какое-то плотным, а какое-то — нет. Уязвимость — понятие относительное. С точки зрения *А. В. Лукацкого* уязвимостью (vulnerability) стоит называть любую характеристику информационной системы, использование которой нарушителем может привести к реализации угрозы. При этом неважно, целенаправленно используется уязвимость или это происходит ненамеренно [2]. В качестве нарушителя может выступать любой субъект корпоративной сети, который попытался осуществить попытку несанкционированного доступа к ресурсам сети по ошибке, незнанию или со злым умыслом (рис. 1).

Как правило, использование этого понятия отдельно сопряжено с рядом трудностей, поскольку всегда возникает логичный вопрос «Если уязвима, то насколько?». Поэтому оценка уровня уязвимости/защищенности будет более правильна и конкретна.

Введем следующие определения и допущения:

1. Жизненный путь программно-технического средства будет оцениваться в количестве выпущенных производителем версий и модификаций;

2. Подсчет количества версий ведется не по числу реально используемых версий, а исходя из формальной системы образования порядкового номера версии. При этом не учитывается факт существования/отсутствия каждой отдельной.

3. Виды и типы уязвимостей классифицируем следующим образом:

• Low — уязвимости типа «поднятие локальных привилегий», но не до local system;

• Midle — уязвимости, мешающие нормальному функционированию системы и приводящие к возникновению DoS, уязвимости приводящие к поднятию локальных привилегий до local system;

• High — уязвимости, позволяющие злоумышленнику получить удаленный контроль над системой.

4. Отношение уязвимостей определенного класса к количеству версий будет измеряться в поинтах. Один поинт будет характеризовать количество уязвимостей данного типа, приходящиеся в среднем на одну версию программно-технического продукта.

Располагая эти оценки на единой шкале (в поинтах) можно говорить о вероятностном уровне защищенности конкретной информационной системы.

Если система имеет несколько целевых узлов, то совокупная уязвимость рассчитывается следующим образом:

$$\text{СУИС} = \text{К}_1 \cdot \text{УИС}_1 + \text{К}_2 \cdot \text{УИС}_2 + \ldots + \\ + \text{К}_i \cdot \text{УИС}_i\,, \quad (1)$$

где i — порядковый номер информационной подсистемы;

СУИС — совокупная уязвимость информационной системы;

К$_i$ — коэффициент долевого участия важности каждой конкретной системы в общей значимости всей ИТ — инфраструктуры. Измеряется в процентах.

Для оценки совокупной уязвимости информационной системы воспользуемся логическими схемами, представленными ниже.

1. Модель последовательного соединения звеньев системы (рис. 2).

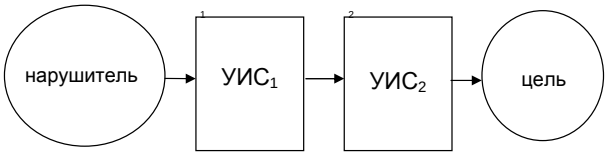

Рис. 2. Последовательная логическая схема «Нарушитель-Цель»

$$p(AB) = p(A) \cdot p(B)\,. \quad (2)$$

2. Модель параллельного соединения звеньев системы (рис. 3).

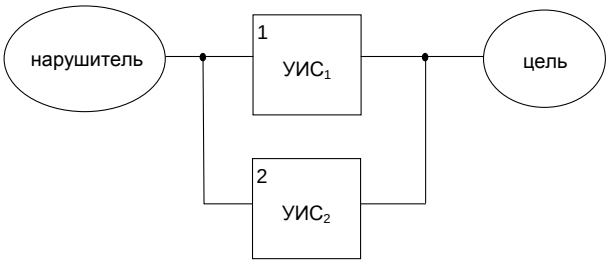

Рис. 3. Параллельная логическая схема «Нарушитель-Цель»

$$p(A + B) = p(A) + p(B) - p(A \cdot B)\,. \quad (3)$$

3. Комбинированная схема соединения звеньев системы (рис. 4).

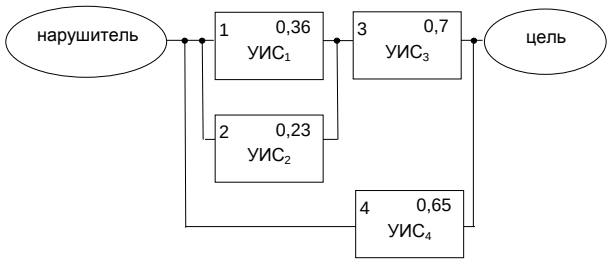

Рис. 4. Комбинированная логическая схема «Нарушитель-Цель»

Получаем формулу для совокупной уязвимости информационной системы:

$$\text{СУИС} = (\text{УИС}_1 + \text{УИС}_2 - \text{УИС}_1 \cdot \text{УИС}_2) \times \\ \times \text{УИС}_3 + \text{УИС}_4 - (\text{УИС}_1 + \text{УИС}_2 - \\ - \text{УИС}_1 \cdot \text{УИС}_2) \cdot \text{УИС}_3 \cdot \text{УИС}_4\,; \\ \text{СУИС} = (0{,}36 + 0{,}23 - 0{,}36 \cdot 0{,}23) \cdot 0{,}7 + \\ + 0{,}65 - (0{,}36 + 0{,}23 - 0{,}36 \cdot 0{,}23) \times \\ \times 0{,}7 \cdot 0{,}65 = 0{,}774\,. \quad (4)$$

Описанной выше методикой была сделана попытка оценить уровень уязвимости конкретного программного продукта с возможностью экстраполяционного прогноза этого уровня на ближайшие версии в пределах допустимой погрешности, без привязки к команде разработчиков, что, в свою очередь, не позволяет увидеть картины в целом.

КАЧЕСТВО КОМАНДЫ РАЗРАБОТЧИКОВ

Если нас интересует не один продукт, а целая линейка, то расчет этих показателей для каждого конкретного представляется весьма трудоемкой задачей. В данном случае логично попытаться оценить качество и уровень знаний самих разработчиков, поскольку, если у команды существуют определенные проблемы по каким-либо направлениям, то они будут находить свое выражение в каждом выпускаемом продукте. Для этого предлагается использовать данные, накопленные в ходе оценки уровня уязвимости ПО. Для этого построим зависимости общего количества уязвимостей от номера версии. Практическую реализацию предлагаемого метода будем демонстрировать на примере Web-сервера Apache:

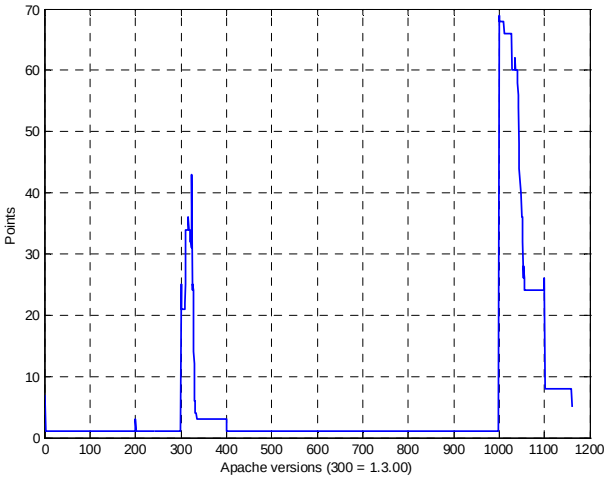

Рис. 5. Версионная уязвимость Web-сервера Apache

Как известно, смена основных номеров версий программного продукта связана с существенными изменениями кода и функциональными преобразования. В пределах этих версий идет доработка уже заложенного функционала и исправление ошибок. Для удобства анализа отсортируем по возрастанию значения графика на этих промежутках (рис. 6).

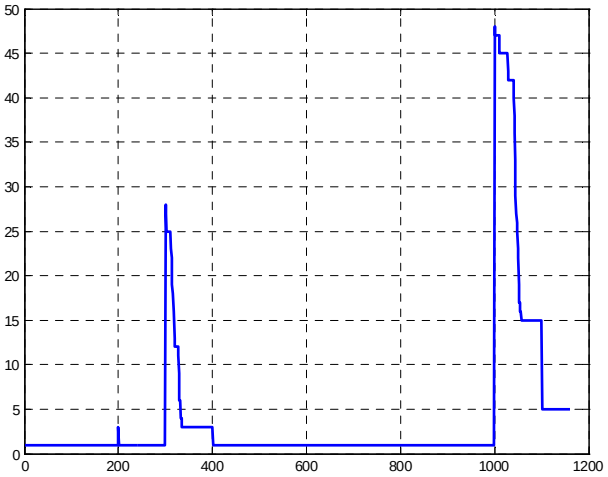

Рис. 6. Отсортированная версионная уязвимость Web-сервера Apache

На этапах существования промежуточных версий, как правило, наблюдается снижение общего уровня уязвимости продукта. Тогда логично попробовать аппроксимировать эти интервалы гиперболической зависимостью вида $y(x) = \frac{c}{x-x_0}$ с вертикальной асимптотой в точке начала основной версии. Делается предположение, что, исследуя скорость убывания общего количества уязвимостей по мере приближения к следующей основной версии, можно выявить проблемные области качества разработки и знания команды программистов.

Большая скорость изменения значений говорит о неоднородности обнаруженных ошибок, либо о их малом количестве, о высоком уровне команды, о малом количестве серьезных ошибок, о том, что у команды есть некоторые не очень существенные пробелы, которые выражаются в смещении ошибок в область с невысокой критичностью уязвимостей.

Малая скорость изменения значений гиперболы говорит об однородности ошибок, что, в свою очередь, показывает серьезные и ярко выраженные проблемы команды разработчиков в определенной сфере безопасного программирования.

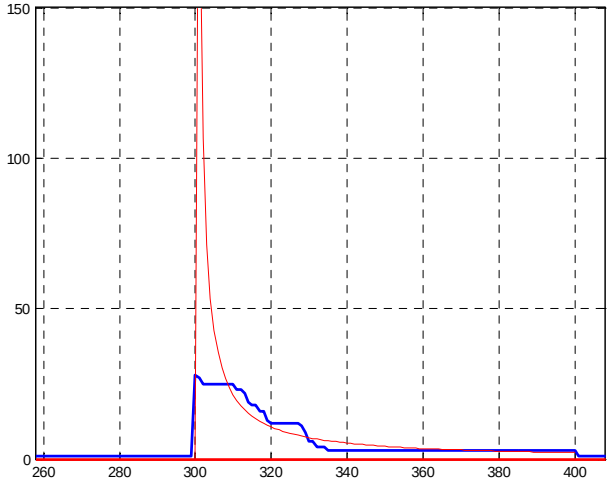

Рис. 7. Аппроксимация первого пика графика

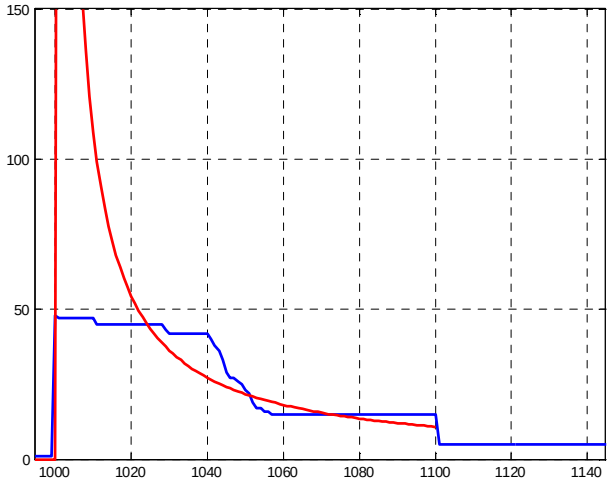

Рис. 8. Аппроксимация второго пика графика

Чем меньше скорость изменения значений гиперболы, тем выше вероятность вер-

ного экстраполяционного прогноза по математическому ожиданию в виду однородности общего уровня уязвимости. Т. е. можно установить границы значений скорости изменения, в пределах которых возможен верный прогноз (рис. 7, 8).

Определим функцию, характеризующую скорость убывания значений графика гиперболы, т. е. найдем первую производную от уравнения гиперболической функции:

$$y'(x) = (\frac{c}{x - x_0})' = -\frac{c}{(x - x_0)^2}. \qquad (5)$$

Из (5) видно, что скорость изменения значений существенно зависит от постоянного коэффициента С. Можно производить оценку вышеописанных параметров исходя из шкалы коэффициентов С. Из формулы (5) следует, что параметр С может изменяться в пределах $[0; +\infty]$.

Поскольку каждая из вышеописанных методик повышает достоверность результирующих оценок другого, то, комбинируя их в двухуровневую модель оценки общего уровня уязвимости, можно достичь более адекватного результата.

ВЫВОД

Использование методики экстраполяционного прогнозирования уровня уязвимости следующей генерации конкретной информационной системы позволяет с определенным уровнем вероятности определить, каким количеством уязвимостей и какого класса она будет обладать в этой самой итерации. Ценность такой информации увеличивается благодаря тому, что получить ее удается заранее (до того, как новая генерация информационной системы выйдет в свет). Обладая данными знаниями заранее и имея возможность оценить по вышеописанной методике профессиональный уровень команды разработчиков в области написания «безопасного» кода, можно принимать решение о переходе на качественно иную структуру и комбинацию подсистем во всей информационной системе в целом, либо же необходимо просто что-то исключить или чем-то дополнить уже имеющуюся (например, дополнительными средствами защиты, которые в конечном итоге закроют «узкие» места в системе и дадут прирост общего уровня защищенности).

СПИСОК ЛИТЕРАТУРЫ

1. **Politov, M. S.** Information systems security analysis problems / M. S. Politov // Computer Science and Information Technologies. Ufa : USATU, 2005. Vol. 2.
2. **Politov, M. S.** Complex system vulnerability estimation / M. S. Politov // Computer Science and Information Technologies. Ufa : USATU, 2007. Vol. 2.
3. **Филин, С. А.** Информационная безопасность / С. А. Филин. Альфа-Пресс, 2006.
4. Common Criteria for Information Technology Security Evaluation. V. 1.0. 31.01.96.

ОБ АВТОРАХ

Политов Михаил Сергеевич, асп. каф. сист. прогр. Челяб. гос. ун-та (ЧелГУ). Дипл. инж. по упр. и информ. в техн. системах (ЮУрГУ, 2005). Готовит дис. в обл. защиты информации.

Мельников Андрей Витальевич, прорек. по науч. раб. того же ун-та. Дипл. инж. по ЭВМ. Д-р техн. наук. Иссл. в обл. защиты информации.