

# 7 МЕТОДЫ И СИСТЕМЫ ЗАЩИТЫ ИНФОРМАЦИИ

УДК 004.056.4

## МОДЕЛЬ КОНКУРЕНЦИИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

А.В. Разумовский

Безопасность программного обеспечения зависит не только от реализованных в нем механизмов защиты, но также и от его способности конкурировать с другими образцами программного обеспечения, выживать в конкурентной борьбе с ними. Для оценки конкурентоспособности программы в будущем необходима модель ее развития в условиях конкуренции с другим программным обеспечением.

**Ключевые слова:** модель, безопасность, программное обеспечение, оценка, информация.

### Введение

Программное обеспечение (ПО) испытывает давление не только со стороны злоумышленников и вредоносных программ, но и со стороны конкурентов. Цель каждого программного продукта – максимизировать свою долю на рынке, увеличить количество копий, установленных на компьютерах пользователей. Чем популярнее становится программа, тем больше предпринимается попыток по ее взлому или компрометации. Логика злоумышленников в данном случае проста: чем популярнее ПО, тем больше инсталляций на компьютерах, тем выше вероятность успешной атаки. В качестве примера можно привести современные браузеры. До недавнего времени браузером компании Microsoft Internet Explorer пользовалось подавляющее число пользователей. Брешы в безопасности появлялись очень часто. По мере роста популярности другого браузера, Firefox от компании Mozilla, в последнем чаще стали находить уязвимости, чем до начала роста его популярности. Уязвимости стали появляться не потому, что разработчики стали меньше уделять внимания безопасности, а потому, что злоумышленники стали предпринимать больше попыток его взлома. Этот пример показывает, что конкуренция между программами напрямую влияет на безопасность программного продукта в целом.

Немаловажным моментом является сложность современного ПО в сочетании с требованием быстрых его изменений. В условиях спешки не всегда можно предусмотреть, как повлияет новый функционал на общую управляемость ПО.

Для прогнозирования роста популярности программного обеспечения, оценки его качества и безопасности будем использовать модель эволюции [2], адаптировав ее к программному обеспечению. Модель включает представление жизненного цикла ПО в виде эволюции во времени набора программ, разработанных по единому техническому заданию, но реализованных различными авторами. Экземпляры программ рассматриваются как конкурирующие объекты. Ввиду несовершенства современных технологий программирования в каждой программе есть ошибки, а, следовательно, и угрозы. С течением времени программы модифицируются с целью устранения угроз. В результате появляются новые программы, а часть программ удаляется из конкуренции. Таким образом, модель предполагает, что каждый экземпляр программы характеризуется тремя свойствами, зависящими от времени: качество, модификация и удаление. Каждая программа описывается вектором параметров, которые характеризуют функционал, определенный в техническом задании, алгоритм его реализации, архитектуру программ, изменения в коде за счет воздействия внешних угроз, таких как вирусы и черви, и т.д. Процесс эволюции заключается в сравнении качества программ и выборе лучшей программы в каждый момент времени (тиражировании лучшего экземпляра, модификации худших программ и удалении плохих программ).

### Оценка качества программ

Перед тем как перейти к описанию самой модели, необходимо каким-то образом оценить программы. Для сравнения программ между собой необходимо определить параметры или критерии, по которым можно сравнить программы. В этой модели выбор критериев оценки и последующую оценку производит эксперт. После оценки всех программ формируется оптимальный набор значений параметров. Оптимальный набор рассчитывается исходя из значений параметров всех программ. В него попадают наиболее часто встречающиеся значения параметров. Обозначим оптимальный набор значений как  $S_m$ . Набор значений параметров каждой программы обозначим как  $S_i$ . После оценки всех программ рассчитаем функцию приспособленности  $f(S)$ :

$$f(S) = \exp[-d_H(S_i, S_m)], \quad (1)$$

где  $d_H$  – разность между параметрами  $i$ -той программы и оптимальным набором параметров. В основе расчета величины лежит расстояние Ли [3].

### Модель эволюции программного обеспечения

Рассматривается множество программ  $X_i$ , которое разделяется на подмножества  $X_j$  близких по характеристикам экземпляров программ, каждая группа  $X_j$  состоит из множества экземпляров программ  $x_i$ , изменяющихся во времени. Динамика развития каждой группы  $X_j$  определяется следующим образом:

$$\frac{dX_j}{dt} = (A_i Q_i - D_i) X_j + \sum_{k \neq i} \omega_{ik} X_k + \Phi_i. \quad (2)$$

Каждая группа программ  $X_j$  стремится максимально увеличить свою численность, для чего каждый экземпляр старается улучшить свои параметры, приближая их к оптимальным. При этом суммарное количество всех экземпляров программ не должно превышать  $X_i$ ,

$$\sum_j X_j \leq X_i. \quad (3)$$

Величина  $A_i$  описывает создание новых копий программы и зависит от приспособленности программы  $f(S_i)$ . Чем ближе параметры программы к оптимальным, тем меньше значение  $f(S_i)$ , соответственно, увеличение количества программ происходит более быстро. При этом увеличивается количество программ, входящих в группу  $X_j$ .

$$A_i = f(S_m) - f(S_i),$$

где  $f(S_i)$  – функция приспособленности (1),  $f(S_m)$  – функция приспособленности программы с идеальными параметрами. Значение  $A$  задается экспертом и задает начальную численность группы.

Величина  $Q_i$  – фактор качества. Более качественные программы распространяются с большей скоростью. Данная величина основывается на количестве экземпляров  $x$  в группе  $X_j$ . В модели качество каждой программы измеряется через расстояние Ли. Таким образом, качество всей группы можно обозначить как сумму качеств программ, входящих в группу  $X_j$ :

$$Q_i = \sum_i d_H,$$

где  $d_H$  – расстояние Ли. Величина  $D_i$  описывает запланированное удаление из группы и обратно пропорциональна функции приспособленности  $f(S_i)$  конкретного экземпляра программы,

$$D_i = \frac{1}{f(S_i)}.$$

Функция приспособленности экспоненциально убывает с ростом разницы между параметрами (1), поэтому коэффициент удаления уменьшается также по экспоненте, т.е., чем выше приспособленность, тем больше экземпляров добавляется в группу (A) и тем меньше экземпляров отмирает (D).

Величина  $\omega$  – процент программ из группы, которые подвергаются воздействиям угроз. Данная величина учитывает воздействие угроз, которым с определенной частотой подвергаются программы. Некоторые из таких воздействий вызывают изменения в работе программ. Таким образом, темп изменений отражает процент программ из общей группы, которые подверглись воздействию угроз. Статистика заражений говорит о том, что процент зараженных программ колеблется на уровне от 0,01 до 0,25 % от всего количества программ [4]. В данной работе мы предполагаем, что заражение экземпляров будет совпадать со статистическими.

$\Phi$  – количество программ, которые удаляются из группы при достижении группой предельного количества экземпляров программ  $i_{\max}$ :

$$\Phi = |i_{\max} - X_j|$$

Экспертное задание A,  $\Phi$  отражает ограниченность ресурсов в модели. В нашем случае внешним ресурсом выступают пользователи, которые не могут все одновременно приобрести какую-либо программу из группы. Параметры Q, D,  $\omega$  рассчитываются непосредственно в процессе взаимодействия программ в группе.

### Результаты моделирования

Результаты компьютерного моделирования, представленные на рисунке, показывают, что чем больше параметры программного обеспечения отличаются от оптимальных, тем сильнее падает численность этих программ, причем критичным является начальное количество программ: если оно достаточно, то программа может изменить один или несколько своих параметров, чтобы приблизиться к оптимуму.

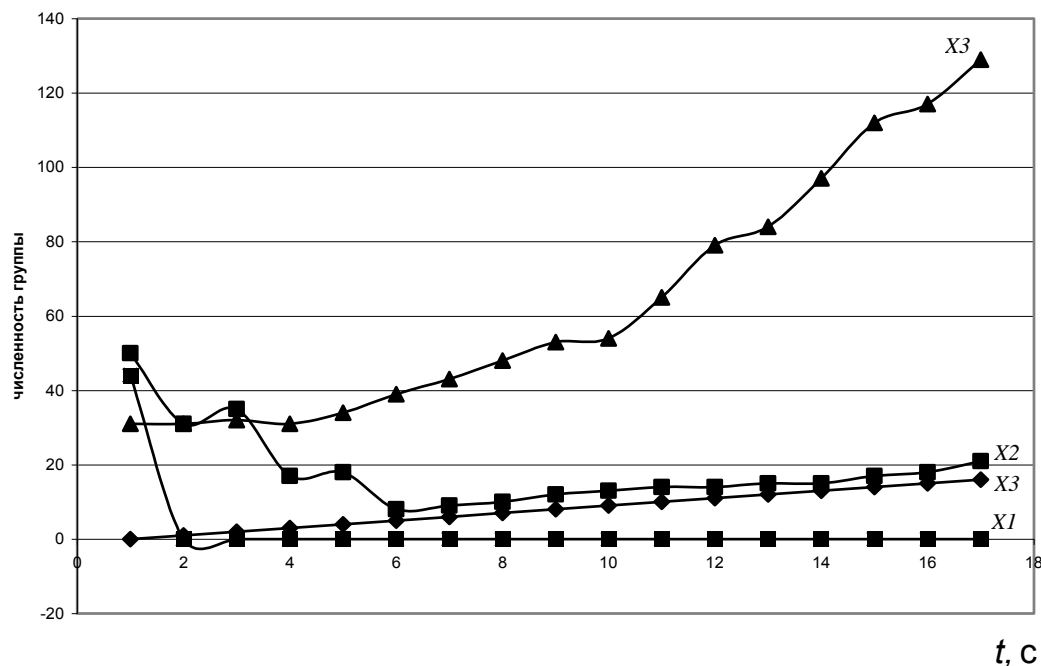


Рисунок. Результат компьютерного моделирования. Верхний график – группа, которая первой достигла ограничения численности, заданного параметром  $\Phi$

Рассмотренная модель может использоваться для оценки качества программного обеспечения, прогнозирования роста количества его копий во времени. Также ее можно использовать при расчете оптимальных параметров на этапе проектирования программного обеспечения, оценки различных сложных структур (узлы сети), для прогнозирования развития сложных программных комплексов.

### **Заключение**

Описанная модель является дополнительным инструментом при разработке современного ПО. Эта модель позволит на ранних этапах разработки моделировать возможные проблемы, в том числе безопасности, разрабатывать более гибкую архитектуру приложений, что в конечном счете приведет к повышению надежности и безопасности программного обеспечения.

### **Литература**

1. Редько В.Г. Эволюционная кибернетика. – М.: Наука, 2001. – 156 с.
2. Эйген М. Самоорганизация материи и эволюция биологических макромолекул. – М.: Мир, 1973. – 216 с.
3. Ушаков А.В., Хабалов В.В., Дударенко Н.А. Математические основы теории систем: элементы теории и практикум / Под ред. Ушакова А.В. – СПб: СПбГУ ИТМО, 2007.
4. Обзор вирусной обстановки от компании «Доктор Веб» за первое полугодие 2009 года [Электронный ресурс]. – Режим доступа: <http://news.drweb.com/show/?i=399&c=10&p=0>, свободный

*Разумовский Андрей Владимирович* – Санкт-Петербургский государственный университет информационных технологий, механики и оптики, аспирант, xrew@yandex.ru