УНИВЕРСИТЕТ ИТМО
НАУЧНО-ТЕХНИЧЕСКИЙ ВЕСТНИК ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ
май–июнь 2017 Том 17 № 3 ISSN 2226-1494 http://ntv.ifmo.ru/
SCIENTIFIC AND TECHNICAL JOURNAL OF INFORMATION TECHNOLOGIES, MECHANICS AND OPTICS
May–June 2017 Vol. 17 No 3 ISSN 2226-1494 http://ntv.ifmo.ru/en

УДК 004.056

АВТОМАТИЧЕСКИЙ АНАЛИЗ ЗАЩИЩЕННОСТИ ИНФОРМАЦИОННЫХ СИСТЕМ БЕЗ ИСПОЛЬЗОВАНИЯ ФОРМАЛЬНЫХ СПЕЦИФИКАЦИЙ

Д.А. Кавчук[a], Ю.Н. Матвеев[b,c]

[a] CloudLinux Inc., Пало-Альто, 94303, США
[b] ООО «ЦРТ-инновации», Санкт-Петербург, 196084, Российская Федерация
[c] Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация
Адрес для переписки: matveev@speechpro.com

Информация о статье
Поступила в редакцию 14.02.17, принята к печати 24.03.17
doi: 10.17586/2226-1494-2017-17-3-431-438
Язык статьи – русский

Ссылка для цитирования: Кавчук Д.А., Матвеев Ю.Н. Автоматический анализ защищенности информационных систем без использования формальных спецификаций // Научно-технический вестник информационных технологий, механики и оптики. 2017. Т. 17. № 3. С. 431–438. doi: 10.17586/2226-1494-2017-17-3-431-438

Аннотация

Предмет исследования. Рассмотрен метод анализа защищенности информационных систем, позволяющий оценить состояние защищенности системы с позиции наличия или отсутствия в ней «незакрытых» уязвимостей, которые можно проэксплуатировать с использованием общедоступных инструментальных средств. Предложенный метод позволяет проанализировать состояние исследуемой информационной системы без составления формальных спецификаций. Проверка проводится на «живой системе» в автоматическом режиме, и наблюдается реакция системы на атакующие воздействия, осуществляемые с использованием системы тестирования на проникновение Metasploit. **Метод.** На основании сопоставления поступающей входной информации строится дерево атак на исследуемую систему, а затем осуществляется его обход, благодаря чему обеспечивается возможность проверки многостадийных атак. Сокращение общего времени, затрачиваемого на анализ защищенности, достигается за счет разметки полученного дерева вероятностями успешного срабатывания его узлов, которая осуществляется посредством нейронной сети на основе радиальных базисных функций и последующего учета вероятностей при обходе дерева. Достоверность проводимого анализа обеспечивается фактической проверкой предположительных уязвимостей в процессе обхода построенного дерева. **Основные результаты.** Предложенный метод реализован в программной системе, и проведено экспериментальное исследование скорости и результативности ее работы. В ходе эксперимента оценивалось состояние защищенности набора информационных систем с использованием разработанной на основе метода программной системы и аналога. По введенному количественному показателю результативности разработанная система превосходит аналог в 1,5–6 раз, что доказывает эффективность предложенного метода. **Практическая значимость.** Разработанная на основе метода программная система может использоваться аналитиками и организациями в качестве самостоятельного средства тестирования на проникновение и анализа защищенности.

Ключевые слова
автоматический анализ защищенности, валидация уязвимостей, деревья атак, графы атак, искусственная нейронная сеть

AUTOMATIC SECURITY ANALYSIS OF INFORMATION SYSTEMS INDEPENDENTLY OF FORMAL SPECIFICATIONS

D.A. Kavchuk[a], Y.N. Matveev[b,c]

[a] CloudLinux Inc., Palo Alto, 94303, USA
[b] "STC-Innovation", Saint Petersburg, 196084, Russian Federation
[c] ITMO University, Saint Petersburg, 197101, Russian Federation
Corresponding author: matveev@speechpro.com

Article info
Received 14.02.17, accepted 24.03.17
doi: 10.17586/2226-1494-2017-17-3-431-438
Article in Russian

For citation: Kavchuk D.A., Matveev Y.N. Automatic security analysis of information systems independently of formal specifications. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2017, vol. 17, no. 3, pp. 431–438 (in Russian). doi: 10.17586/2226-1494-2017-17-3-431-438

Abstract

Subject of Research. The paper considers the method for security analysis of information systems. The method enables to evaluate the security state of information system under research in terms of the presence of unpatched vulnerabilities, which could be exploited with the public instruments. The proposed method allows for the state analysis of information system under research with no need to compose any formal specifications. The validation is carried out upon the live system in automatic mode, and system reaction to the attacking influences, performed with the Metasploit penetration testing platform, is observed. **Method.** The attack tree for the system under research is being constructed on the basis of the input data matching. The tree traversal follows. This provides the possibility of multi-stage attack validation. The decrease of total security analysis time period is achieved due to marking the constructed tree with probabilities of its nodes successful triggering and probability accounting during tree traversal. This probabilistic elaboration is performed with the help of radial-basis artificial neural network. Reliability of performed analysis is provided with the actual validation of presumptive vulnerabilities during tree traversal. **Main Results.** The program system is implemented on the basis of the proposed method. The experiments on the processing rate and effectiveness are carried out. During the experiment the security state of information systems from the set was analyzed with the help of developed program and its analog. The developed system transcends the analog from 1.5 to 6 rate by the introduced quantitative index of effectiveness. This fact proves the efficiency of proposed method. **Practical Relevance.** Organizations and security analysts could apply the program system, implemented on the basis of proposed method, as the standalone penetration testing and security analysis instrument.

Keywords

automatic security analysis, vulnerability validation, attack trees, attack graphs, artificial neural network

Введение

Высокий темп развития информационных технологий и все возрастающая сложность современных информационных систем обусловливают необходимость мониторинга их защищенности от внешних и внутренних угроз в процессе эксплуатации систем. Анализ защищенности имеет целью выявление и подтверждение уязвимостей, являющихся основной причиной нарушения безопасности систем, с целью повышения уровня защищенности рассматриваемой системы и данных, хранимых и обрабатываемых в ней. На решение данной проблемы направлено множество разработок, среди которых присутствуют как формальные методики, так и инструментальные средства.

Среди формальных методик в данной области можно выделить подходы на основе деревьев атак, сетей Петри и графов атак. При этом последние нашли наибольшую распространенность среди исследователей.

Деревья атак были представлены Б. Шнайером в 1999 г. в работе [1]. При анализе защищенности с использованием данной модели выбирается цель атаки и создается концептуальная диаграмма действий, направленных на достижение цели, в виде дерева, корнем которого является сама цель. Описанный процесс носит формальный характер и выполняется вручную, что позволяет говорить о модели деревьев атак как о формальной методологии для оценки безопасности систем.

В более поздних работах [2–4] была представлена усовершенствованная модель деревьев атак. Улучшения заключаются во введении дополнительной логики объединения листьев дерева и в добавлении листьям атрибутов. Однако решение, предлагаемое усовершенствованной моделью деревьев атак, помимо унаследованных недостатков, также имеет очень узкую область применения – сетевые системы обнаружения вторжений.

В работе [5] предлагается использовать сеть Петри для анализа защищенности. Позициями такой сети, получившей в работе название сети атаки, являются ресурсы системы, переходами – события, приводящие к изменению состояния ресурсов. Перемещение маркеров по сети отображает ход атаки. Построение и анализ сетей атак осуществляется вручную. Таким образом, подход находит применение только при формальной разработке вероятных сценариев атак.

Модели, основанные на графах атак, находят применение в различных сферах. Также как и деревья атак, графы атак применяются для улучшения качества работы систем обнаружения вторжений [6, 7]. В сфере анализа защищенности можно выделить работы [8, 9].

В работе [8] представлен ориентированный на эксплойты граф, узлами которого являются эксплойты, а дугами – условия успешного их срабатывания. При этом эксплойты представлены в терминах предусловий и постусловий. Графы строятся не для всей сети в совокупности, а для выбранной цели атаки (ресурса или объекта). Подход графов атак [8] реализован в инструменте TVA (Topological Vulnerability Analysis) [10], который строит графы на основании потока входных данных, включающего модель сетевой конфигурации, базу знаний о смоделированных эксплойтах и выбранный сценарий атаки. Если модель конфигурации сети генерируется сканером безопасности, то база эксплойтов составляется и обновляется разработчиками вручную, а сценарий атаки требуется описать силами пользователя. Таким образом, привязка к фактическому состоянию защищенности сети осуществляется только в модели сетевой конфигурации, получаемой от сканера безопасности, который, в свою очередь, не застрахован от ложных срабатываний и пропуска уязвимостей.

В работе [9] представлен граф множественных переходов, который позволяет проанализировать защищенность сети с точки зрения возможности компрометации ее хостов из любой точки сети, т.е. по-

средством других ее хостов. Система NetSPA, в основу которой лег данный подход, требует ручной подготовки более чем половины входных данных для построения графа. Для получения оставшейся части входных данных применяются сканеры безопасности, результаты работы которых не подвергаются никакой валидации – т.е. найденная сканером уязвимость признается существующей всегда, что далеко не всегда верно. Это, в свою очередь, может привести к ложному признанию узла сети скомпрометированным.

Все представленные подходы обладают общими недостатками: необходимость составления вручную формальных спецификаций, определяемых конкретной моделью; отсутствие валидации данных, получаемых от сторонних инструментальных средств (если использование таковых предусмотрено моделью); отсутствие фактических исследований на «живой» системе и учета ее реакции на воздействие.

В настоящей работе предлагается метод автоматического анализа защищенности, позволяющий оценить фактическое состояние защищенности исследуемой информационной системы без составления формальных спецификаций системы, а только на основании взаимодействия с системой и наблюдения ее реакции на взаимодействие. Метод также позволяет обнаруживать возможность проведения многостадийных атак на исследуемую информационную систему.

Метод автоматического анализа защищенности

На основании исследования информации в открытых источниках актуальной становится задача разработки метода анализа защищенности, который не требует составления формальных спецификаций и решает следующие задачи:

- валидация данных, полученных от сканера безопасности;
- проверка на «живой» системе, т.е. наблюдение за поведением исследуемой системы в процессе проведения атакующих воздействий и ее реакцией на них;
- проверка многостадийных атак, т.е. тех ситуаций, когда успешное проведение одной атаки создает не существовавшие ранее благоприятные условия для успешного проведения другой.

В предложенном в данной работе методе автоматического анализа защищенности используется модель деревьев атак, дополненная такими категориями входных данных, как информация об исследуемой информационной системе, о доступных эксплойтах и результатах запуска эксплойтов. Метод позволяет учесть в качестве входных данных об исследуемой системе любую информацию, в том числе и набор установленного программного обеспечения (ПО).

Дерево атак на исследуемую информационную систему строится следующим образом. Узлами дерева являются эксплойты, выбранные на основании интеллектуального анализа соответствия уязвимостям и характеристикам исследуемой системы. Дуги дерева представляют собой переходы от одной стадии атаки к другой. Таким образом, каждый путь в дереве представляет собой сценарий атаки.

Результирующее дерево атак, полученное после обхода построенного, содержит только сработавшие узлы (эксплойты). Цепочки эксплойтов представляют собой все подтвержденные проверкой на «живой» системе пути атаки – от корня к листьям – на исследуемую информационную систему.

Шаги метода можно сформулировать следующим образом [11, 12].

1. Получение информации об исследуемой системе:
   $osInf = \{S, Ver, SP, serv, ports\}$,
   где S – семейство операционной системы (ОС); $Ver = \{ver_i\}$ – конечное множество (к.м.) предположительных версий ОС; $SP = \{sp_i\}$ – к.м. предположительных *service pack* ОС; $serv = \{serv_i\}$ – к.м. сервисов; $ports = \{port_i\}$ – к.м. портов.
2. Получение информации об уязвимостях системы:
   $vulnInf = \{vuln_i\}$,
   где $vuln_i$ – предположительная уязвимость системы.
3. Получение информации о доступных средствах эксплуатации системы (эксплойтах):
   $explInf = \{expl_i\}$,
   где $expl_i$ – доступный эксплойт. Информация об эксплойте, как правило, включает
   $expl_i = \{rank_i, REF, targets, description_i\}$,
   где $rank_i$ – ранг эксплойта (показатель эффективности $REF = \{ref_i\}$ – к.м. ссылок на уязвимости; $targets = \{target_i\}$ – к.м. целей; $description_i$ – описание эксплойта.
4. Построение дерева атак:
   $G = (V, E)$,
   где $V = \{exp_i\}$ – множество вершин, $exp_i = F((expl_i.targets, osInf.S) | (expl_i.REF, vulnInf))$ – допустимый эксплойт; $E = \{T_i\}$ – множество ребер, T_i – допустимый переход от одного эксплойта к другому.

При определении допустимых переходов T от одного эксплойта к другому учитываются следующие правила:

1. если эксплойт может сработать без предварительных условий, то он размещается в узле самого верхнего уровня, т.е. следующего за корнем;
2. если же для успешного срабатывания эксплойта необходима реализация некоторых предварительных условий, то он размещается в поддереве, корнем которого является реализующий данные условия узел.

Допустимые эксплойты exp определяются на основании анализа соответствия уязвимостям системы и ее характеристикам (версия ОС и т.д.). Анализ соответствия и сопоставление эксплойта характеристикам исследуемой системы осуществляется с использованием лексического анализатора, разработанного для языка описания целей эксплойтов (платформ, для атаки на которые эксплойт предназначен). В результате анализа особенностей языка описания целей была построена морфологическая модель, ориентирующаяся на полное покрытие лексики рассматриваемого. Разработанный на основании модели анализатор позволяет определять принадлежность лексемы одному из классов лексем: система, версия, ветвь, тип или язык [13]. На основе данных, предоставляемых лексическим анализатором, делается вывод о соответствии рассматриваемого эксплойта исследуемой системе.

5. Уточнение дерева атак:

$P_i = neuronet(vector_i)$,

где P_i – вероятность успешного срабатывания эксплойта, $vector_i = <normRank_i, coeffR_i, coeffT_i>$.

Разметка построенного дерева атак вероятностями срабатывания эксплойтов осуществляется с использованием искусственной нейронной сети на основе радиальных базисных функций (RBF). Данный тип сетей успешно применяется для решения задач классификации, кластеризации и прогнозирования, а их основное преимущество заключается в скорости обучения [14].

Для расчета вероятности P_i успешного срабатывания эксплойта на вход нейронной сети подается трехкомпонентный вектор $vector_i$, включающий следующие значения:

1. нормированное значение показателя эффективности (ранга) эксплойта, получаемое по формуле

$normRank=\frac{currRank}{bestRank}$,

где $currRank$ – ранг рассматриваемого эксплойта, $bestRank$ – максимальный ранг.

2. оценка степени соответствия эксплойта набору уязвимостей системы, вычисляемая по формуле

$coeffR=\frac{coREF}{allREF}$,

где $coREF$ – количество совпадений описаний уязвимостей для рассматриваемого эксплойта с набором уязвимостей исследуемой системы, $allREF$ – общее количество уязвимостей исследуемой системы.

3. оценка степени соответствия эксплойта исследуемой ОС, определяемая широко известным методом взвешенных сумм [15], где производится свертка параметров ОС с учетом их относительной важности:

$coeffT=0,55\cdot OS+0,3\cdot ver+0,15\cdot sp$,

где OS – совпадение типа ОС из описания эксплойта с типом ОС исследуемой системы; ver – совпадение версии ОС из описания эксплойта с версией ОС исследуемой системы; sp – совпадение *service pack* ОС из описания эксплойта с *service pack* ОС исследуемой системы.

Каждая из переменных OS, ver, sp может принимать значения 0 либо 1. Для получения их значений используются данные, предоставляемые лексическим анализатором. Весовые коэффициенты были получены эмпирически с учетом их относительной важности в формировании общей оценки.

Обученная на массиве данных вида [входной вектор → желаемый отклик], состоящем из 27 примеров, сеть имеет три нейрона входного слоя, 10 нейронов скрытого слоя и один нейрон выходного слоя.

6. Обход дерева атак.

Обход дерева производится в ширину по пути с наибольшей вероятностью срабатывания эксплойтов. При обработке очередного узла осуществляется запуск соответствующего узлу эксплойта с параметрами, наиболее подходящими исследуемой системе и набору открытых портов и активных сервисов. Результат запуска используется для принятия решения о дальнейших действиях: если эксплойт не срабатывает, производится усечение всего поддерева с корнем в данном узле, и далее обход оставшейся части дерева продолжается по убыванию вероятности путей.

Таким образом, результирующее дерево атак на исследуемую систему включает только успешные сценарии атак и позволяет получить наглядное представление о состоянии защищенности исследуемой системы.

Несмотря на то, что метод позволяет учесть в качестве входных данных об исследуемой системе любую информацию, в том числе и набор установленного ПО, на данный момент разработаны алгоритмы подбора эксплойтов, учитывающие данные об ОС, под управлением которой работает информационная система, и о наборе уязвимостей информационной системы.

Программная система автоматического анализа защищенности

С целью оценки эффективности метода была разработана программная система автоматического анализа защищенности, структура которой показана на рис. 1.

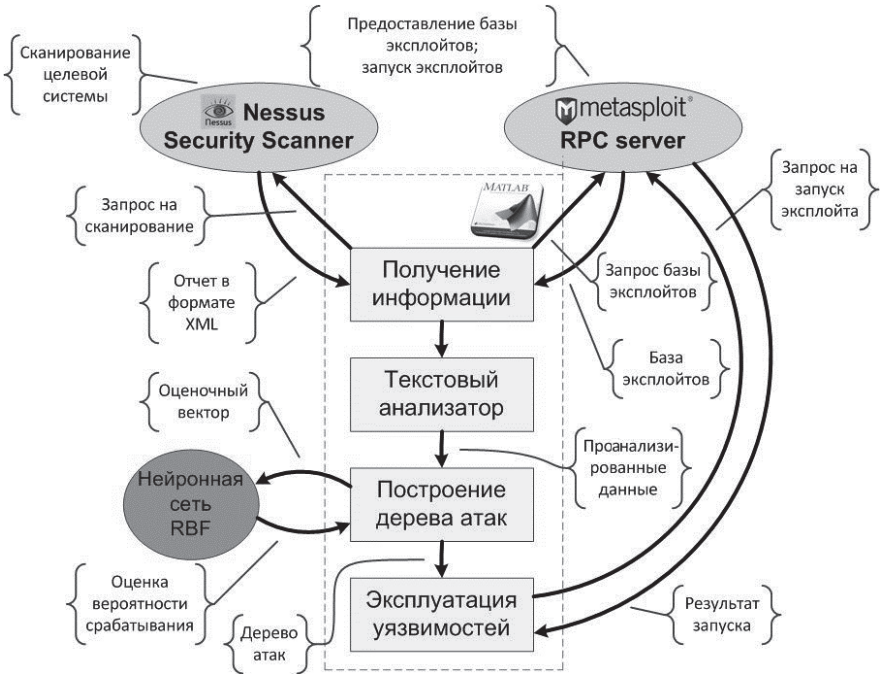

Рис. 1. Структура программной системы

В работе системы участвуют три взаимодействующих компонента: сканер безопасности Nessus используется для получения информации о характеристиках и уязвимостях исследуемой системы; система тестирования на проникновение Metasploit Framework – для получения базы доступных эксплойтов и их запуска; управляющая программа-анализатор реализует разработанный метод анализа защищенности. Управляющая программа написана в среде MATLAB, взаимодействие управляющей программы со сторонним ПО реализовано на языке Python.

Как правило, в Metasploit к любому эксплойту, даже к эксплойту на конкретное ПО, привязана информация об операционной системе. В связи с этим в текущей программной реализации метода дерево строится для информационной системы в целом, без конкретики в части ПО и сервисов. При обходе дерева активные эксплойты выбираются с учетом набора сервисов, функционирующих в ОС, а пассивные – для интернет-браузера, поскольку это ПО является предустановленным во всех широко распространенных ОС. Другие эксплойты, вошедшие в дерево, не запускаются ввиду отсутствия алгоритмов анализа соответствия по ПО.

Взаимодействие с Nessus и Metasploit Framework осуществляется в автоматическом режиме в блоке получения информации. Вступающий в работу далее блок текстового анализа решает две важные задачи:

1. извлечение данных об ОС, портах, сервисах и уязвимостях исследуемой системы из отчета Nessus и приведение их к пригодному для дальнейшей обработки виду;
2. лексический анализ поля Targets эксплойтов (целей).

После подготовки и формализации всех необходимых для проведения анализа защищенности данных система строит дерево атак. В блоке построения дерева подбираются удовлетворяющие исследуемой системе эксплойты, определяются связи между ними и посредством нейронной сети на основе радиальных базисных функций присваиваются вероятности срабатывания каждому попавшему в дерево эксплойту. В текущей реализации программной системы поддерживается только один тип связи: любой пассивный эксплойт (для реализации которого необходима активность со стороны исследуемой системы) – эксплойт 2-го уровня – может сработать только после успешного срабатывания активного эксплойта (не требующего действий со стороны исследуемой системы) – эксплойта 1-го уровня.

Блок эксплуатации уязвимостей реализует обход дерева атак в соответствии с определенными в методе правилами: сначала запускаются активные эксплойты по убыванию вероятности (наиболее вероятные запускаются первыми) и с учетом доступных портов и работающих сервисов. В случае успешного результата осуществляется запуск пассивных эксплойтов в той же манере; в противном случае обход дерева завершается.

На рис. 2 приведен пример результата работы программной системы автоматического анализа защищенности, в основу которой лег разработанный метод. Рис. 2 иллюстрирует не дерево, а граф, поскольку было сделано следующее допущение: любой пассивный эксплойт может сработать после любого активного.

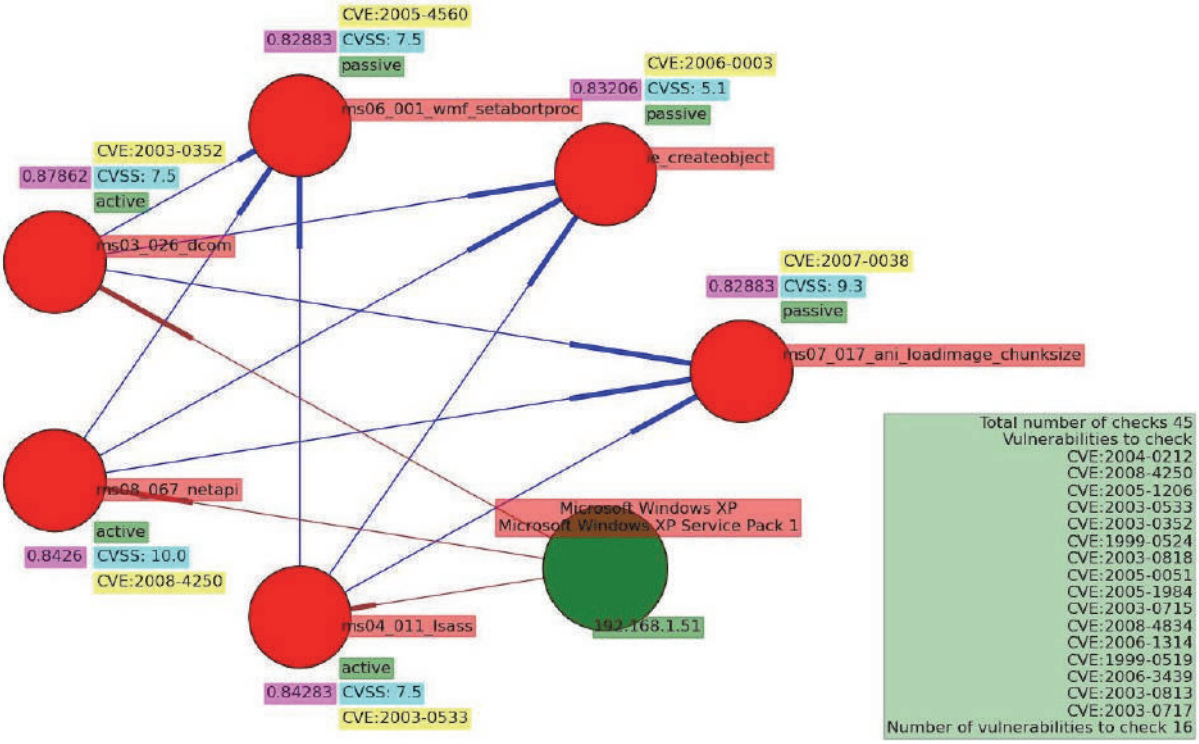

Рис. 2. Пример результирующего графа атак

Визуализация результатов работы системы осуществляется с использованием средств языка Python. Корневой узел выделен зеленым цветом и представляет собой состояние системы до начала процесса анализа защищенности. Узел маркирован информацией о платформе, под управлением которой работает исследуемая информационная система, и ее IP-адресом. Узлы, представляющие собой эксплойты, выделены красным цветом и маркированы следующей информацией: короткое наименование эксплойта в системе Metasploit Framework, его тип, вероятность срабатывания, привязанный идентификатор CVE [16] и базовая метрика CVSS [17]. Дуги красного цвета направлены от корня к активным эксплойтам, дуги синего цвета – от активных эксплойтов к пассивным эксплойтам.

Расположенный справа блок информации отображает общее количество проверок, выполненное системой, и список уязвимостей, предоставленный сканером безопасности для валидации.

Экспериментальное исследование метода

В ходе экспериментального исследования в целях доказательства работоспособности предложенного метода оценивалась защищенность информационных систем, работающих на платформах семейства Windows. В качестве аналога для сравнительного анализа был выбран инструмент автоэксплуатации *Metasploit db_autopwn* как наиболее подходящий по принципу работы.

Благодаря введенным вероятностным оценкам, разработанной программной системой было проведено меньше проверок, чем ПО *Metasploit db_autopwn*; при этом во всех случаях было подтверждено большее, либо равное количество предположительных уязвимостей, что доказывает эффективность разработанной системы в сравнении с данным аналогом. По количественному показателю результативности разработанная система превосходит аналог минимум в 1,5 раза и максимум в 6 раз.

Экспериментальные результаты для всех платформ в совокупности позволяют выявить распределение вероятности нахождения всех фактических уязвимостей за заданное число проверок, а также распределение количества сработавших эксплойтов по заданному числу проверок. Графики распределений представлены на рис. 3, 4 соответственно.

Из графика на рис. 3 видно, что при 50 проверках вероятность обнаружения и подтверждения всех уязвимостей составляет 0,8, при 75 проверках – 0,95, при 100 проверках – 0,999. Таким образом, можно утверждать, что первой сотни отобранных по вероятности срабатывания эксплойтов достаточно для проверки всех присутствующих в исследуемой системе уязвимостей, и дальнейшего увеличения количества проверок не требуется.

График на рис. 4 иллюстрирует, что наибольшее количество уязвимостей подтвердилось в промежутке от 40 до 50 заданных проверок, а на первых 20 проверках уже было подтверждено 30% от общего количества фактических уязвимостей.

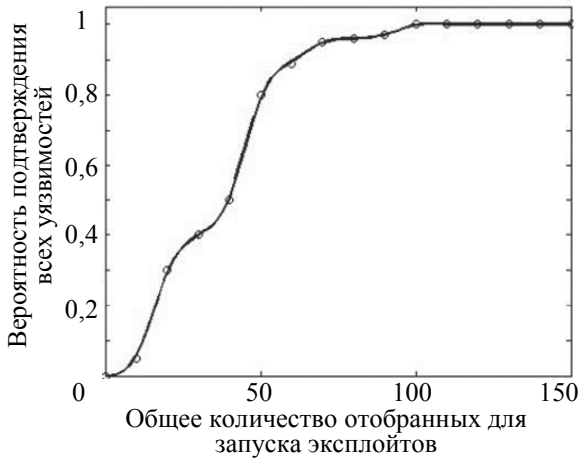

Рис. 3. График распределения вероятности

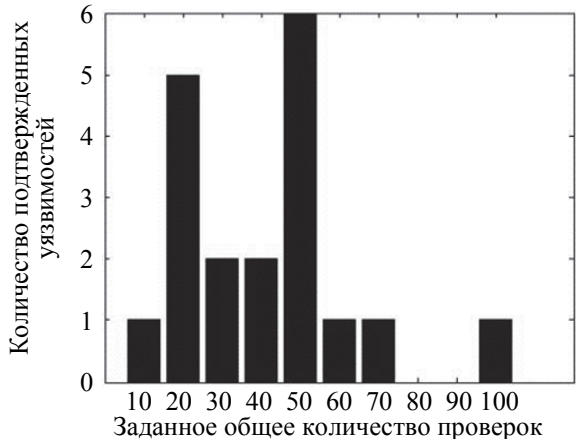

Рис. 4. График распределения количества подтвержденных уязвимостей

Заключение

Разработан метод автоматического анализа защищенности информационных систем, позволяющий провести проверку с учетом реакции исследуемой системы на атакующие воздействия и без использования формальных спецификаций.

Разработанная на основе предложенного метода программная система доказывает работоспособность метода и его эффективность в сравнении с выбранным аналогом. Полученных экспериментальных результатов удалось добиться благодаря интеллектуальной валидации полученных от сканера безопасности данных и введению вероятностных оценок для узлов дерева, а также благодаря проверке многостадийных атак. Преимуществами системы в сравнении с аналогом также являются полностью автоматический анализ и визуализация сценариев атаки с указанием их критичности (оценки CVSS). Метод и программная система могут быть использованы аналитиками при тестировании на проникновение, а также организациями для анализа состояния защищенности их информационных систем. В качестве дальнейшего направления исследования можно рассмотреть разработку алгоритма построения более сложных связей между эксплойтами-узлами дерева и, как следствие, получение многоуровневых деревьев и графов с глубокими сценариями атак, а также разработку алгоритмов подбора эксплойтов с учетом информации об установленном на исследуемой системе программном обеспечении.

Литература

1. Schneier B. Attack trees // Dr. Dobb's Journal. 1999. V. 24. N 12. P. 21–29.
2. Camtepe S.A., Yener B. A formal method for attack modeling and detection // Technical Report TR-06-01. Rensselaer Polytechnic Institute, 2006.
3. Camtepe S.A., Yener B. Modeling and detection of complex

References

1. Schneier B. Attack trees. *Dr. Dobb's Journal*, 1999, vol. 24, no. 12, pp. 21–29.
2. Camtepe S.A., Yener B. A formal method for attack modeling and detection. *Technical Report TR-06-01*. Rensselaer Polytechnic Institute, 2006.
3. Camtepe S.A., Yener B. Modeling and detection of complex

attacks // Proc. 3rd Int. Conf. on Security and Privacy in Communications Networks and the Workshops. Nice, France, 2007. P. 234–243. doi: 10.1109/SECCOM.2007.4550338
4. Дородников Н.А., Арустамов С.А. Разработка вероятностной поведенческой модели для защиты вычислительной сети с использованием деревьев атак // Научно-технический вестник информационных технологий, механики и оптики. 2016. Т. 16. № 5. С. 960–962. doi: 10.17586/2226-1494-2016-16-5-960-962
5. McDermott J.P. Attack net penetration testing // Proceedings New Security Paradigms Workshop. Ballycotton, Ireland, 2000. P. 15–21.
6. Sheyner O., Haines J., Jha S., Lippmann R., Wing J.M. Automated generation and analysis of attack graphs // Proceedings of the IEEE Symposium on Security and Privacy. Oakland, 2002. P. 273–284. doi: 10.1109/SECPRI.2002.1004377
7. Jha S., Sheyner O., Wing J.M. Two formal analyses of attack graphs // Proc. 15th IEEE Workshop on Computer Security Foundations. Cape Breton, Canada, 2002. P. 49–63. doi: 10.1109/CSFW.2002.1021806
8. Jajodia S., Noel S., O'Berry B. Topological analysis of network attack vulnerability / In: Managing Cyber Threats: Issues, Approaches and Challenges. Springer-Verlag, 2005. P. 248–266.
9. Jajodia S., Noel S. Topological vulnerability analysis: a powerful new approach for network attack prevention, detection and response / In: Algorithms, Architectures, and Information Systems Security. Eds B. Bhattacharya, S. Sur-Kolay, S. Nandy, A. Bagch. Springer, 2009. 384 p.
10. Ingols K., Lippmann R., Piwowarski K. Practical attack graph generation for network defence // Proc. 22nd Annual Computer Security Applications Conference. Miami Beach, USA, 2006. P. 121–130. doi: 10.1109/ACSAC.2006.39
11. Tumoyan E., Kavchuk D. The method of optimizing the automatic vulnerability validation // Proc. 5th Int. Conf. on Security of Information and Networks (SIN '12). NY, 2012. P. 205–208. doi: 10.1145/2388576.2388586
12. Тумоян Е.П., Кавчук Д.А. Метод оптимизации автоматической проверки уязвимостей удаленных информационных систем // Безопасность информационных технологий. 2013. №1. С. 25–30.
13. Кавчук Д.А. Лексический анализ в задачах моделирования сетевых атак // XI Всероссийская научная конференция молодых ученых, студентов и аспирантов «Техническая кибернетика, радиоэлектроника и системы управления». Таганрог, 2012. Т. 2. С. 87.
14. Хайкин С. Нейронные сети: полный курс. 2-е изд. М.: Вильямс, 2006. 1104 с.
15. Подиновский В.В., Потапов М.А. Метод взвешенной суммы критериев в анализе многокритериальных решений: pro et contra // Бизнес-информатика. 2013. №3(25). С. 41–48.
16. Waltermire D., Scarfone K. Guide to using vulnerability naming schemes. NIST Special Publication (SP) 800-51. National Institute of Standards and Technology, 2011.
17. Chambers J., Thompson J. Common Vulnerability Scoring System: Final Report and Recommendations. National Infrastructure Advisory Council, 2004.

attacks. *Proc. 3rd Int. Conf. on Security and Privacy in Communications Networks and the Workshops*. Nice, France, 2007, pp. 234–243. doi: 10.1109/SECCOM.2007.4550338
4. Dorodnikov N.A., Arustamov S.A. Probabilistic behavioral model for computer network protection based on attack trees. *Scientific and Technical Journal of Information Technologies, Mechanics and Optics*, 2016, vol. 16, no. 5, pp. 960–962. (In Russian) doi: 10.17586/2226-1494-2016-16-5-960-962
5. McDermott J.P. Attack net penetration testing. *Proceedings New Security Paradigms Workshop*. Ballycotton, Ireland, 2000, pp. 15–21.
6. Sheyner O., Haines J., Jha S., Lippmann R., Wing J.M. Automated generation and analysis of attack graphs. *Proc. IEEE Symposium on Security and Privacy*. Oakland, 2002, pp. 273–284. doi: 10.1109/SECPRI.2002.1004377
7. Jha S., Sheyner O., Wing J.M. Two formal analyses of attack graphs. *Proc. 15th IEEE Workshop on Computer Security Foundations*. Cape Breton, Canada, 2002, pp. 49–63. doi: 10.1109/CSFW.2002.1021806
8. Jajodia S., Noel S., O'Berry B. Topological analysis of network attack vulnerability. In *Managing Cyber Threats: Issues, Approaches and Challenges*. Springer-Verlag, 2005, pp. 248–266.
9. Jajodia S., Noel S. Topological vulnerability analysis: a powerful new approach for network attack prevention, detection and response. In *Algorithms, Architectures, and Information Systems Security*. Eds B. Bhattacharya, S. Sur-Kolay, S. Nandy, A. Bagch. Springer, 2009, 384 p.
10. Ingols K., Lippmann R., Piwowarski K. Practical attack graph generation for network defence. *Proc. 22nd Annual Computer Security Applications Conference*. Miami Beach, USA, 2006, pp. 121–130. doi: 10.1109/ACSAC.2006.39
11. Tumoyan E., Kavchuk D. The method of optimizing the automatic vulnerability validation. *Proc. 5th Int. Conf. on Security of Information and Networks, SIN '12*. New York, 2012, pp. 205–208. doi: 10.1145/2388576.2388586
12. Tumoyan E.P., Kavchuk D.A. Optimizing automated vulnerability assessments of remote information systems. *IT Security*, 2013, no. 1, pp. 25–30. (In Russian)
13. Kavchuk D.A. Lexical analysis in the tasks of modeling network attacks. *Proc. XI All-Russian Scientific Conf. on Technical Cybernetics, Radio Electronics and Control Systems*. Taganrog, Russia, 2012, vol. 2, p. 87. (In Russian)
14. Haykin S. *Neural Networks: A Comprehensive Foundation*. Pearson Education, 1999, 823 p.
15. Podinovski V., Potapov M. Weighted sum method in the analysis of multicriterial decisions: pro et contra. *Business Informatics*, 2013, no. 3, pp. 41–48. (In Russian)
16. Waltermire D., Scarfone K. Guide to using vulnerability naming schemes. *NIST Special Publication (SP) 800-51*. National Institute of Standards and Technology, 2011.
17. Chambers J., Thompson J. *Common Vulnerability Scoring System: Final Report and Recommendations*. National Infrastructure Advisory Council, 2004.

Авторы

Кавчук Дарья Александровна – программист, CloudLinux Inc., Пало-Альто, 94303, США, dasha.kavchuk@yandex.ru

Матвеев Юрий Николаевич – доктор технических наук, главный научный сотрудник, ООО «ЦРТ-инновации», Санкт-Петербург, 196084, Российская Федерация; заведующий кафедрой, Университет ИТМО, Санкт-Петербург, 197101, Российская Федерация, matveev@speechpro.com

Authors

Daria A. Kavchuk – software developer, CloudLinux Inc., Palo Alto, 94303, USA, dasha.kavchuk@yandex.ru

Yuri N. Matveev – D.Sc., Chief Scientific Officer, "STC-Innovation", Saint Petersburg, 196084, Russian Federation; Head of Chair, ITMO University, Saint Petersburg, 197101, Russian Federation, matveev@speechpro.com