УДК 004.4.056 + 681.518 + 005.334 Вестник УрФО № 3(25) / 2017, с. 34–42

Баринов А. Е., Скурлаев С. В., Соколов А. Н.

МЕТОДИКА ОЦЕНКИ РИСКОВ, ВЫЗВАННЫХ УЯЗВИМОСТЯМИ В ПРОГРАММНОМ ОБЕСПЕЧЕНИИ АВТОМАТИЗИРОВАННЫХ СИСТЕМ УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМИ ПРОЦЕССАМИ

В последнее время существенно участились атаки на автоматизированные системы управления технологическими процессами (АСУ ТП), вызванные уязвимостями в их программном обеспечении (ПО). Однако специфика АСУ ТП такова, что зачастую невозможно мгновенно установить исправление ПО даже при его наличии, а внеплановая остановка АСУ ТП для обслуживания является затратным мероприятием. Поэтому актуальным является вопрос оценки критичности уязвимостей в целях планирования установки исправлений. В статье описана модель влияния информационных потоков на технологические процессы (ТП) в АСУ ТП и предложена методика оценки критичности уязвимости, содержащая временной параметр, а также степень влияния на управляемые технологические функции.

Ключевые слова: *информационная безопасность АСУ ТП, метрика, уязвимость, оценка рисков, управление рисками, управление исправлениями, информационный поток.*

Barinov A. E., Skurlaev S. V., Sokolov A. N.

METHOD FOR RISK ASSESSMENT CAUSED BY VULNERABILITIES IN ICS SOFTWARE

Recently, attacks on ICS and SCADA caused by vulnerabilities in software have increased significantly. However, the specificity of ICS is that it is often impossible to immediately install the patch, even if it exists, and the unplanned stop of process control systems for maintenance is a costly event. Therefore, the issue of assessing the criticality of vulnerabilities for the purpose of planning the installation of patches is relevant. This paper describes the model of the influence of information flows on industrial process in the ICS and suggests a methodology for assessing the criticality of the vulnerability, containing time parameters, and the degree of influence on managed technological functions.

Keywords: *ICS Security, metric, vulnerability, risk assessment, risk management, patch management, information flow.*

АСУ ТП довольно давно используются во многих отраслях промышленности, однако актуальность их информационной безопасности существенно возросла лишь в последнее время с учащением инцидентов в промышленных системах. Обеспечение безопасности информации АСУ ТП наиболее критично для предприятий топливно-энергетического комплекса, поскольку инциденты информационной безопасности на таких объектах могут привести не только к серьезным экономическим последствиям (нарушение поставок нефти и газа, перебои в электроснабжении населения и т.п.), но и к экологическим или гуманитарным катастрофам. Кроме того, уязвимыми являются различные транспортные системы[1], металлургические, химические, ядерные производства, инфраструктура связи и т.п. Приведем лишь некоторые известные примеры кибератак на АСУ ТП, приведших к значительным негативным последствиям:

• Stuxnet (первая атака на АСУ ТП, поразившая ядерные объекты Ирана)[2];

• Crouching Yeti (атака на предприятия энергетики, машиностроения, фармацевтического сектора США и стран Евросоюза с целью кражи конфиденциальной информации о ТП)[3];

• BlackEnergy (семейство атак на украинскую энергетическую систему, вызвавших её сбой)[4].

Некоторые из угроз реализуются в промышленных сетях в силу их недостаточной изоляции от корпоративных сетей (BlackEnergy)[4] или через внешние носители (Stuxnet)[2]. Кроме того, уязвимости могут возникать непосредственно в оборудовании промышленных сетей: PLC, HMI и т.д. По имеющимся данным[5], за 2016 год обнаружено свыше 180 таких уязвимостей.

Специфика обеспечения информационной безопасности АСУ ТП в отличие от корпоративных систем имеет следующие особенности:

• индустриальные информационные системы могут работать годами, их остановка либо неприемлема, либо невозможна. Это усложняет установку исправлений для ПО с целью устранения уязвимостей;

• каждая АСУ ТП специфична по критичности ТП, степени потенциального ущерба и т.д.;

• многие АСУ ТП используют закрытые проприетарные протоколы, их производители либо не реализуют адекватных политик поиска, устранения и сопровождения уязвимостей, либо соответствующие релизы сняты с поддержки.

Вышеперечисленное приводит к тому, что при обеспечении информационной безопасности АСУ ТП необходимы постоянный аудит и оценка рисков.

Установка исправления является критичным процессом, требующим принятий целого комплекса мер по тестированию исправления, вводу его в АСУ ТП, анализа необходимости остановки ТП и т.д.

Известны рекомендации[6], где предложен алгоритм принятия решения по установке исправления (рис. 1). Однако предложенный алгоритм не учитывает снижение эффективности системы при применении обходного решения, а также не предоставляет аналитического аппарата для проведения анализа рисков информационной безопасности и его сопоставления с операционными рисками.

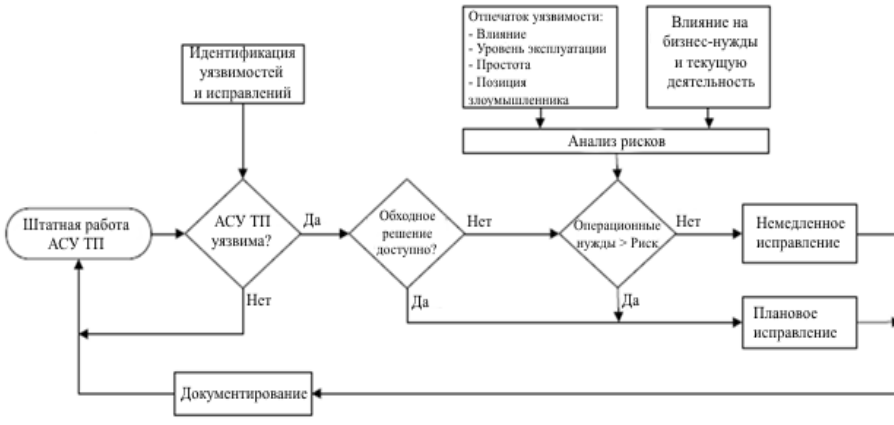

Рис. 1. Алгоритм принятия решения об установке исправления

Документ[6] вводит понятие отпечаток уязвимости на основе таких метрик, как:

• *уровень эксплуатации (D)* – доля узлов, подверженных рассматриваемой уязвимости (при этом допускается применение весов для определения приоритетов, подверженных уязвимости узлов);

• *влияние (I)* – степень контроля над системой и степень потенциально наносимого ей ущерба;

• *позиция злоумышленника (E)* – (включая его местоположение по отношению к периметру, требуемым привилегиям и т.д.);

• *простота (S)* – степень доступности средств для эксплуатации уязвимости и степень навыков злоумышленника.

3) описанный подход не сопоставляет время обнаружения уязвимости и расписание сервисных окон с целью определения оптимального момента для устранения уязвимостей.

Целью настоящей работы является построение модели оценки критичности уязвимости для формализации процесса её устранения.

В результате анализа процессов обработки информации в АСУ ТП и информационных потоков[7,8] разработана модель АСУ ТП (рис. 2), представленная в виде диаграммы классов UML.

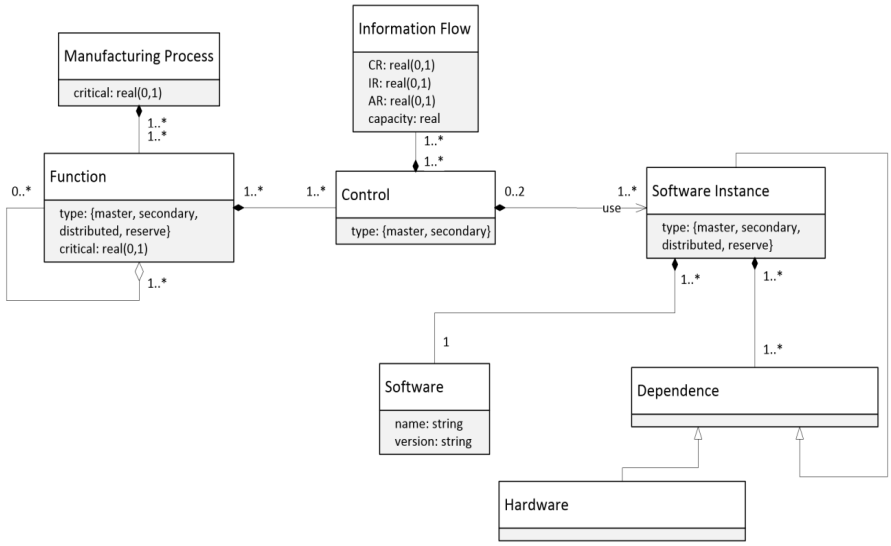

Рис. 2. Модель АСУ ТП

При формировании значений для этих составляющих документ[6] полагается на экспертную оценку значений по шкале с градациями: *низкая, средняя, высокая*. Это не позволяет полноценно выполнить процесс анализа рисков и сопоставить его с функциональными нуждами АСУ ТП. Поэтому очевидны следующие недостатки:

1) низкая степень гранулярности показателей не даёт возможности приоритезации для двух уязвимостей со сходными показателями, появившихся почти одновременно;

2) экспертная оценка показателей не позволяет адекватно сопоставить риск, вносимый рассматриваемыми уязвимостями, и риск функциональных нарушений в АСУ ТП. Таким образом, сложно оценить, какие именно производственные функции и процессы деградируют, а также степень их деградации. Кроме того, для сопоставления с числовыми значениями используется устаревший стандарт CVSS 2.0;

Под технологическим процессом (Manufacturing Process) будем понимать относительно автономную часть производства, выполняющую совокупность функций, направленных на реализацию продукта. Промышленное предприятие может реализовывать несколько независимых ТП, для которых, как правило, можно выделить независимые АСУ ТП. Критичность (Critical) отдельно взятого ТП может характеризовать степень ущерба при нарушении его нормальной работоспособности.

В настоящее время крупные страны имеют рекомендательные документы по классификации АСУ ТП. В частности, в России и США подобная классификация производится по величине потенциального финансового ущерба и количеству потенциальных жертв. В рамках представленной модели оценку предлагается производить величиной, верхнее значение которой соответствует максимально возможному уровню критичности.

Функция (function) в представленной модели определена как физический элемент технологического процесса (включая производственное, контрольное оборудование и т.п.). Функции образуют иерархическую структуру. Такой подход обеспечивает:

• возможность разбиения каждой функции верхнего уровня (например, подачи железной руды в доменном производстве) на более простые функции, контролируемые отдельными автоматизированными системами (например, очистку руды и т.п.);

• повторяемость использования функций (например, общие производственные функции (электроснабжение, вентиляция) могут обслуживать несколько технологических процессов). Эти общие функции считаются основными для того же уровня, который они обслуживают.

В модель также включены функции:

• *резервные* (reserve) – задействуются при отказе основных (master);

• *распределённые* (distributed) – работают параллельно (например, несколько одновременно работающих линий подачи);

• *вспомогательные* (secondary) – нарушение их работы на некоторый интервал времени не оказывает существенного влияния на ТП. К таким функциям можно отнести, например, функцию мониторинга температуры технологического оборудования.

Под *элементом контроля* (control) понимается промышленный контроллер или его автономная часть, принимающая сигналы от контролируемых функций и передающая им команды управления. При этом функции верхнего уровня контролируются программными средствами, обеспечивающими координацию всего ТП, а функции низовой автоматизации – программным обеспечением исполняющих устройств и средств, ими управляющих. Вполне очевидно, что элемент контроля может контролировать несколько функций, а одна функция может принимать сигналы (или отправлять данные) более чем одному элементу контроля. Элемент контроля для исполнения своих функций создаёт *информационные потоки* (Information Flow), обрабатываемые программным обеспечением или получаемые от него для передачи другим элементам контроля или для хранения с целью повторного использования. В свою очередь элемент контроля использует некоторое множество *установок ПО* (Software Instance) определённых версий и конфигураций. При этом каждая установка ПО может обслуживать не более двух элементов контроля (основной и резервный) или не обслуживать ни один, если является зависимостью (различное обеспечивающее ПО, например, СУБД для SCADA, операционные системы и т.п., и оборудование, на которое установлено ПО). Элемент контроля может быть *основным* (master), который непосредственно обслуживает производственные функции или *вспомогательным* (secondary). Наиболее очевидный пример вспомогательного элемента контроля – самодиагностика систем.

В понятие *информационный поток* (Information Flow) входят:

• поток данных, формируемый функциями или элементом контроля;

• поток данных, формируемый командами, получаемыми от ПО;

• набор данных, хранящихся непосредственно на контролирующем элементе.

Информационные потоки предназначаются для обмена информацией между различными элементами контроля ТП или реализации элементами контроля своих внутренних функций (например, данные о текущем состоянии ТП, хранящиеся внутри контроллера, его конфигурация и т.д.). Очевидно, что один элемент контроля может производить и потреблять разнородные информационные потоки, например основной поток команд для оборудования, данные о текущем состоянии, информацию о структуре АСУ ТП для взаимодействия с другими элементами контроля и т.д. Можно отметить, что информационная ёмкость (capacity) этих потоков является различной, также как и требования по обеспечению различных метрик информационной безопасности: *конфиденциальности* (CR), *целостности* (IR) и *доступности* (AR). Любая информация в АСУ ТП может быть представлена в виде некоторого информационного потока. Кроме того, получать и использовать один информационный поток могут несколько контролирующих элементов.

Управляющую роль для элемента контроля исполняет *установка ПО* (Software Instance). Установка ПО реализует свою работу посредством ПО (Software) определённой версии и конфигурации, а также и его зависимостей. Установка ПО может быть *основной* (master), если её отказ приводит к нарушению работы соответствующего элемента контроля, *резервной* (reserve) и *параллельной* по аналогии с функциями (distributed). С целью

упрощения модели характеристики оборудования детально не анализируются.

Выше отмечено, что к составляющим отпечатка уязвимости относятся такие метрики как *уровень эксплуатации* (D), *влияние* (I) , *позиция злоумышленника* (E), *простота* (S).

Стоит отметить, что две последние метрики отражают характеристики злоумышленника и могут быть относительно легко сформированы из описания данных уязвимости в CVSS 3.0. Первые две метрики являются более сложными. Рассмотрим процесс формирования этих метрик. Здесь и далее будем полагать, что значение 0 означает неприменимость метрики или отсутствие риска, а 1 – полный контроль над системой со стороны злоумышленника.

Уязвимое ПО может использоваться в работе функции только посредством контролирующих элементов, а контролирующий элемент, в свою очередь, может обслуживать разные по критичности функции и различные по требованиям информационной безопасности информационные потоки. Поэтому параметры влияния и уровень эксплуатации должны рассчитываться отдельно для каждого контролирующего элемента.

Для каждого информационного потока характерно 3 метрики информационной безопасности: *конфиденциальность* (CR), *целостность* (IR) и *доступность* (AR). Оценка критичности каждой из этих метрик необходима для оценки общего влияния уязвимости на АСУ ТП. Каждая метрика информационного потока оказывает влияние на характеристики эффективности *производственных функций f(j)*. Обозначим *показатели эффективности производственных функций как* $p_{if(j)}$. Каждый показатель $p_{if(j)}$ зависит от параметров информационных потоков, которые на него влияют, а также имеет предельное значение $p_{if(j)t}$, при котором производственная функция деградирует. Поскольку каждый информационный поток входит в несколько производственных функций, а информационная ёмкость потока может меняться со временем эксплуатации системы, задача определения оптимальных параметров информационных потоков носит статистический и оптимизационный характер. Определим базовые параметры информационных потоков и метрики информационной безопасности, которые они формируют.

Метрика *доступности* (AR) – является одной из самых очевидных и определяет уровень доступности как долю времени работы системы с заданными характеристиками. Эта метрика формируется на этапе проектировании системы, поэтому при оценке величины AR можно опираться на проектные значения.[9] Исследование зависимости показателей эффективности производственных функций от значения метрики доступности является широко исследованным вопросом, например[7].

Метрика *целостности* (IR) – может быть представлена на основе двух составляющих: минимальной *доли информации*, которая является *достаточной для управления ТП* ($\delta_{\varepsilon m}$) и *способности информации к восстановлению* ($\delta_{\varepsilon r}$). Формирование первой составляющей ($\delta_{\varepsilon m}$) можно наглядно продемонстрировать на следующем примере: при ежесекундном измерении температуры для получения сведений о динамике процесса зачастую достаточно одного из десяти измерений, следовательно, $\delta_{\varepsilon m} = 0{,}1$. Вторая составляющая ($\delta_{\varepsilon r}$) характеризует способность информации восстанавливаться либо за счёт встроенной избыточности (на уровне логики или избыточного кодирования), либо за счёт переотправки без потери производительности. В итоге, значение метрики IR можно определить как

$$IR = \delta_{\varepsilon m} \times \delta_{\varepsilon r}. \quad (1)$$

Метрика *конфиденциальности* (CR) является одной из самых сложных и чаще всего оценивается экспертным путём. Имеются работы, оценивающие конфиденциальность только на основе вероятностных характеристик[10] или стоимости информации. Оба метода не в полной мере могут подходить для АСУ ТП, так как для большинства ТП нарушение конфиденциальности информации не является критичным в краткосрочной перспективе (по сравнению с доступностью). С другой стороны, получая доступ к технологической информации, злоумышленник может получить не только информацию о ноу-хау, но и изучить структуру системы, что при нечастых исправлениях в АСУ ТП может облегчить ему дальнейшие атаки. Поэтому в предположении, что информация устаревает по экспоненте[11], определим следующие параметры:

k – доля полезной информации в текущем информационном потоке. К полезной не относится служебная информация, по которой нельзя получить сведения о структуре системы, информация о ноу-хау и т.д. (оценивается экспертно);

a – постоянная устаревания информации;

CR_B – базовый критерий конфиденциаль-

ности, определённый для данного информационного потока исходя из требований организации.

Вводя нормировку $\{k;\ \alpha;\ CR_B\} \in [0;1]$, получим значение метрики CR в виде

$$CR = \frac{\log_\alpha k \int_{t_c}^{t_i} v(t)dt + k\int_{t_c}^{t_i}\left(\int_{t_c}^{t} v(t)dt\right)\left(1-\alpha^t\right)dt}{(t_i - t_c + 1\log_\alpha k)\int_{t_c}^{t_i} v(t)dt} \times CR_B, \qquad (2)$$

где t_c – текущее время, t_i – предполагаемое время прекращения угрозы утечки информации, $v(t)$ – потенциальная скорость утечки информации, определяемая из статистических наблюдений за пропускной способностью потенциального канала утечки, учитывая его ёмкость и прогнозируемую загруженность.

Тогда оценка влияния уязвимости I_{if} на конкретный информационный поток *if* может быть выполнена, как[12]

$$I_{if}=1-(1-CR\times R_{CR})(1-IR\times R_{IR})(1-AR\times R_{AR}), \qquad (3)$$

где IR и CR определены, соответственно, в (1) и (2), а веса ранее определенных метрик R_{CR}, R_{IR}, R_{AR} связаны соотношением[13]

$$R_{CR}+R_{IR}+R_{AR}=1.$$

Итоговую оценку влияния I_m для m-го элемента контроля можно выполнить исходя из известного соотношения

$$I_m=1-\Pi^n_{if=1}(1-I_{if}), \qquad (4)$$

где n – число информационных потоков, а значение I_{if} определено в (3).

Очевидно, что влияние уязвимости тем выше, чем

- большее число узлов, реализующих ту или иную функцию, подвержено уязвимости,
- большее число функций обслуживается уязвимым ПО.

В этом случае *уровень эксплуатации* D_m для m-го элемента контроля может быть рассчитан как

$$D_m = D_{fm} \times D_{sm}, \qquad (5)$$

где D_{fm} – уровень функциональной эксплуатации, а D_{sm} – уровень программной эксплуатации.

На рис. 2 видно, что функция представляет собой древовидную структуру. Критичность основной функции D_i технологического процесса определяется его критичностью $T \in [0;1]$. Для формирования критичности конкретной функции дерево обходится вниз.

Если входящая в функцию подфункция является *основной*, то её критичность соответствует критичности функции верхнего уровня, (например конвейер – критичность всех элементов одинакова, т.к. остановка одного приводит к остановке всего конвейера).

Если функция f_i реализуется набором параллельных функций, то критичность каждой из них будет являться $D_i=m/n$, где n – число параллельных подфункций, реализующих данную функцию, а m – их минимальное число, необходимое для реализации функции на необходимом уровне.

Если функция является *вспомогательной* (например, мониторинг), то оценивается его степень влияния на основную функцию k, а критичность вспомогательной функции определяется как kD_i.

Для *резервной* функции критичность определяется как $F(t_i) \times D_i$,где $F(t_i)$ – функция распределения отказов резервируемого элемента, полученная статистически. Для резервируемого элемента, критичность, соответственно, имеет вид $(1 – F(t_i)) \times D_i$. При этом если функция оказывается *подфункцией* для нескольких технологических процессов или функций, то её критичность определяется максимальной из расчётных критичностей. Если один контролирующий элемент управляет несколькими функциями с разной степенью критичности, то уровень функциональной эксплуатации может быть рассчитан, как

$$D_{fm}=\Pi^n_{i=1}(1-D_i).$$

Программное обеспечение также можно представить древовидной структурой. Расчет уровня программной эксплуатации D_{si} можно выполнить по следующему алгоритму:

1) если уязвимое ПО является *основным* компонентом, подверженным уязвимости, то уровень эксплуатации D_{si} считается равным 1;

2) если основной компонент резервируется другим компонентом на другом стеке ПО, то его уровень эксплуатации $D_{si}= m/n$, где m – сумма уровней эксплуатации уязвимых стеков ПО, а n – число суммарное число компонентов резервных и резервируемых;

3) если ПО является распределенным, то уровень программной эксплуатации $D_{si} \times F_s(t_i)$ соответствует доле установок ПО, потенциально подверженных атаке на основе статистических наблюдений;

4) если установка ПО – *вспомогательная* (например, для мониторинга состояния сервера, на котором установлено ПО управления АСУ ТП), то его влияние может быть оценено через коэффициент, то есть kD_{si};

5) если установка ПО – *резервная*, то уровень его программной эксплуатации может быть рассчитан как $D_{si} \times F_{se}(t_i)$, где $F_{se}(t_i)$ – распределение отказов основных компонентов,

и, соответственно, переходов на резервный компонент;

6) если установка ПО является *зависимостью* для нескольких стеков ПО, то уровень её программной эксплуатации определяется максимальным из уровней.

Итоговый уровень эксплуатации для элемента контроля определяется значением, появившимся на вершине дерева при обходе его снизу вверх.

Простота (S) является характеристикой не инфраструктуры, а непосредственно уязвимости. Исходя из анализа[6,12], можно выразить *простоту* как

$$S = (AC - 0{,}44) \times 1{,}52 + (ECM \times RC - 0{,}83) \times 6{,}14, \quad (6)$$

где AC – показатель сложности проведения атаки, ECM – зрелость кода для эксплуатации уязвимости, RC – степень детальности описания уязвимости. Указанные значения берутся из описаний уязвимостей в репозиториях и интерпретируются в соответствии с CVSS 3.0, либо модифицируются на уровне организации путём экспертных оценок. Переводные коэффициенты получены с учетом того, что параметр S находится в диапазоне [0;1].

Аналогично параметру S, выразим *позицию злоумышленника (E)* как

$$E = (AV \times PR \times UI - 0{,}03) \times 1{,}71, \quad (7)$$

где AV – вектор атаки, PR – требуемые для эксплуатации уязвимости привилегии, UI – требуемое для эксплуатации уязвимости пользовательское взаимодействие.

Таким образом, для каждого элемента контроля имеем 4 параметра *I, D, E, S*, рассчитанных для конкретной уязвимости в (4), (5), (6) и (7).

Оценку риска для j-го элемента контроля, подверженного конкретной уязвимости, можно оценить при этом как[14]:

$$C_j = \frac{\lfloor 10I_j \rfloor + \lfloor 10D_j \rfloor}{18} \times \frac{\lfloor 10S \rfloor + \lfloor 10E \rfloor}{18}.$$

Суммарная оценка риска по всем элементам контроля вычисляется как

$$C = \frac{\lfloor 10S \rfloor + \lfloor 10E \rfloor}{18} \times \sum_{j=1}^{n} \frac{\lfloor 10I_j \rfloor + \lfloor 10D_j \rfloor}{18},$$

где n – число элементов, контролирующих стеки, ПО которых подвержены данной уязвимости.

Таким образом, возможность расчёта риска для отдельно взятого элемента контроля позволяет выделить те участки инфраструктуры, где устранение уязвимости необходимо выполнить в первую очередь. Разработанная методика при этом позволяет провести оценку влияния уязвимостей программного обеспечения на информационную безопасность АСУ ТП с целью их приоритезации и последующего устранения. Представленная модель является многопараметрической. В качестве одного из параметров базовых аналитических выражений используется параметр t_i – предполагаемый момент времени прекращения угрозы утечки информации. Он позволяет оценить интервал времени воздействия потенциальной угрозы с момента возникновения до сервисных окон обслуживания, если для установки исправления требуется остановка технологического процесса. Такой подход позволяет адекватно спланировать установку исправлений и ответить на вопрос, нужно ли проводить исправления в срочном порядке или следует выбрать для этого соответствующее сервисное окно.

Оценки параметров эффективности производственных функций, используемые в модели, позволяют также оценить критичность уязвимости в зависимости от степени деградации производственных функций. Перспективным направлением моделирования представляется оценка взаимного влияния уязвимостей, когда в системе действует несколько уязвимостей низкого риска, но совместное их воздействие может спровоцировать высокий риск для инфраструктуры. Имеющиеся в этой области работы либо не отражают специфику АСУ ТП[13], либо оценивают влияние уязвимостей на количественных характеристиках без учёта особенностей применимости к конкретной системе.

Статья выполнена при поддержке Правительства РФ (Постановление №211 от 16.03.2013 г.), соглашение № 02.A03.21.0011.

Примечания

1. Андрей Заикин. Почему защита АСУ ТП сегодня стала критически важной? [Электронный ресурс] // https://www.securitylab.ru/analytics/484730.php. (Дата обращения: 30.10.2017).

2. Keizer, Greg. Is Stuxnet the 'best' malware ever? Infoworld. 16 September 2010. [Электронный ресурс] // https://www.infoworld.com/article/2626009/malware/is-stuxnet-the--best--malware-ever-.html. (Дата обращения: 30.10.2017).

3. KL ICS CERT. Nigerian phishing: industrial companies under attack [Электронный ресурс] // https://ics-cert.kaspersky.com/reports/2017/06/15/nigerian-phishing-industrial-companies-under-attack/. (Дата обращения: 30.10.2017).

4. Securelist. Kaspersky Lab's Global Research & Analysis Team. BlackEnergy APT Attacks in Ukraine employ spearphishing with Word documents [Электронный ресурс] // https://securelist.com/blackenergy-apt-attacks-in-ukraine-employ-spearphishing-with-word-documents/73440/. (Дата обращения: 30.10.2017).

5. ICS-CERT Monitor November/December 2016, National Cybersecurity and Communications Integration Center, 2016.

6. Recommended Practice for Patch Management of Control Systems (December 2008). U.S. Department of Homeland Security.

7. Friedberg, I. Towards a Resilience Metric Framework for Cyber-Physical Systems. 4th International Symposium for ICS & SCADA Cyber Security Research 2016, Queen's Belfast University, UK, 23 - 25 August 2016, pp. 19-22.

8. Tebbe, C. Ontology and life cycle of knowledge for ICS security assessments. 4th International Symposium for ICS & SCADA Cyber Security Research 2016, Queen's Belfast University, UK, 23 - 25 August 2016, pp. 32-41.

9. ITIL Service Design TSO, Norwich, United Kingdom, 2011.

10. Jonsson, E. An attempt to quantitative modelling of behavioural security, 11th International Information Security Conference, Cape Town, South Africa, May 1995.

11. Cobb, J. What is the half-life of your information? [Электронный ресурс] // http://captricity.com/blog/half-life-information/. (Дата обращения: 30.10.2017).

12. Common Vulnerability Scoring System v3.0: Specification Document (v1.7).

13. Баринов А.Е., Рябцева О.В., Соколов А.Н. Адаптивная оценка клиентского риска в облачных инфраструктурах. Вестник УрФО. Безопасность в информационной сфере. № 1(23) / 2017, С 14–19.

14. Cayirci, E., Garaga, A., Santana de Oliveira, A. et al. J Cloud Comp (2016) 5: 14. doi:10.1186/s13677-016-0064-x.

References

1. Andrey Zaikin. Pochemu zashchita ASU TP segodnya stala kriticheski vazhnoy? [Elektronnyy resurs] // https://www.securitylab.ru/analytics/484730.php. (Data obrashcheniya: 30.10.2017).

2. Keizer, Greg. Is Stuxnet the 'best' malware ever? Infoworld. 16 September 2010. [Elektronnyy resurs] // https://www.infoworld.com/article/2626009/malware/is-stuxnet-the--best--malware-ever-.html. (Data obrashcheniya: 30.10.2017).

3. KL ICS CERT. Nigerian phishing: industrial companies under attack [Elektronnyy resurs] // https://ics-cert.kaspersky.com/reports/2017/06/15/nigerian-phishing-industrial-companies-under-attack/. (Data obrashcheniya: 30.10.2017).

4. Securelist. Kaspersky Lab's Global Research & Analysis Team. BlackEnergy APT Attacks in Ukraine employ spearphishing with Word documents [Elektronnyy resurs] // https://securelist.com/blackenergy-apt-attacks-in-ukraine-employ-spearphishing-with-word-documents/73440/. (Data obrashcheniya: 30.10.2017).

5. ICS-CERT Monitor November/December 2016, National Cybersecurity and Communications Integration Center, 2016.

6. Recommended Practice for Patch Management of Control Systems (December 2008). U.S. Department of Homeland Security.

7. Friedberg, I. Towards a Resilience Metric Framework for Cyber-Physical Systems. 4th International Symposium for ICS & SCADA Cyber Security Research 2016, Queen's Belfast University, UK, 23 - 25 August 2016, pp. 19-22.

8. Tebbe, C. Ontology and life cycle of knowledge for ICS security assessments. 4th International Symposium for ICS & SCADA Cyber Security Research 2016, Queen's Belfast University, UK, 23 - 25 August 2016, pp. 32-41.

9. ITIL Service Design TSO, Norwich, United Kingdom, 2011.

10. Jonsson, E. An attempt to quantitative modelling of behavioural security, 11th International Information Security Conference, Cape Town, South Africa, May 1995.

11. Cobb, J. What is the half-life of your information? [Elektronnyy resurs] // http://captricity.com/blog/half-life-information/. (Data obrashcheniya: 30.10.2017).

12. Common Vulnerability Scoring System v3.0: Specification Document (v1.7).

13. Barinov A.Ye., Ryabtseva O.V., Sokolov A.N. Adaptivnaya otsenka kliyentskogo riska v oblachnykh infrastrukturakh. Vestnik UrFO. Bezopasnost' v informatsionnoy sfere. № 1(23) / 2017, s. 14–19.

14. Cayirci, E., Garaga, A., Santana de Oliveira, A. et al. J Cloud Comp (2016) 5: 14. doi:10.1186/s13677-016-0064-x.

БАРИНОВ Андрей Евгеньевич, старший преподаватель кафедры защиты информации, младший научный сотрудник, и.о. директора НОЦ «Информационная безопасность» ФГАОУ ВО «Южно-Уральский государственный университет (национальный исследовательский университет)». Россия, 454080, г. Челябинск, пр. Ленина, 76. E-mail: barinovae@susu.ru

ANDREY Barinov, Federal State Autonomous Educational Institution of Higher Education "South Ural State University (national research university)", Chelyabinsk, Russian Federation. E-mail: barinovae@susu.ru

СКУРЛАЕВ Сергей Вадимович, старший преподаватель кафедры защиты информации ФГАОУ ВО «Южно-Уральский государственный университет (национальный исследовательский университет)». Россия, 454080, г. Челябинск, пр. Ленина, 76. E-mail: svskurlaev@susu.ru

SERGEY Skurlaev, Federal State Autonomous Educational Institution of Higher Education "South Ural State University (national research university)", Chelyabinsk, Russian Federation. E-mail: svskurlaev@susu.ru

СОКОЛОВ Александр Николаевич, канд. техн. наук, доцент, заведующий кафедрой защиты информации ФГАОУ ВО «Южно-Уральский государственный университет (национальный исследовательский университет)». Россия, 454080, г. Челябинск, пр. Ленина, 76. E-mail: ANSokolov@inbox.ru

ALEXANDER Sokolov, Federal State Autonomous Educational Institution of Higher Education "South Ural State University (national research university)", Chelyabinsk, Russian Federation. E-mail: ANSokolov@inbox.ru