В дополнение к этому предпринимаются специальные меры по минимизации неявных каналов коммуникации между приложениями (т.н. скрытые каналы). Современные реализации ядер безопасности на основе архитектуры MLS могут использовать аппаратные функции виртуализации, предоставляемые последними поколениями процессоров. Это позволяет, к примеру, реализовать гипервизор, способный выполнять гостевые ОС поверх ядра безопасности MLS в виртуализированной среде. Такой подход автоматически обеспечивает MLS-ядру изоляцию данных и контроль над информационными потоками, позволяя исключить появление скрытых каналов [34].

Внедрение предлагаемых в работе [34] методов обеспечения информационной безопасности на борту ВС может позволить сохранить надежность авиационных систем на высоком уровне, предотвратить несчастные случаи на воздушном транспорте и улучшить качество предоставляемых услуг.

На сегодняшний день самым эффективным методом обеспечения информационной безопасности ВС является деление бортового оборудования на безопасные домены, с помощью которых можно четко установить границы, где обмен информацией должен отвечать наивысшим требованиям безопасности, в то время как другие домены могут иметь более низкий уровень доверия и взаимодействовать с сетями общего пользования, не беспокоясь о том, что потенциальные угрозы навредят критическим системам ВС.

В итоге можно утверждать, что для обеспечения безопасной передачи данных необходимо и достаточно разместить на борту интеллектуальные устройства безопасности во всех местах, где происходит стыковка систем.

Обеспечение ИБ с помощью отдельных бортовых защищенных устройств характеризуется следующими основными преимуществами: отсутствие повышения нагрузки на центральные бортовые вычислители из-за программно-аппаратной поддержки функций обеспечения ИБ; возможность реализации механизмов обеспечения программной и аппаратной отказоустойчивости при возникновении угроз ИБ.

В результате системы смогут быстро и безопасно соединяться с внешними сетями, увеличить эффективность обмена данными благодаря более широкой полосе пропускания (например, совместное использование частот для беспроводных систем), а также облегчить доступность к бортовым системам для обслуживающего и технического персонала.

3.7. Методы выявления программных уязвимостей

В работе [13] рассмотрены такие методы вявления программных уязвимостей, как проверка безопасности программного кода в процессе сертификационных испытаний и тематических исследований по требованиям безопасности. В данной статье приведены примеры выявленных уязвимостей исходя из опыта работы испытательной лаборатории.

3.7.1. Виды сертификационных испытаний

В Российской Федерации основным легитимным способом выявления уязвимостей ПО является обязательная сертификация средств защиты информации по требованиям безопасности информации (по линии Минобороны России, ФСБ

России и ФСТЭК России) [14]. Это связано с тем, что при сертификации официально предоставляются необходимые спецификации, имеется обратная связь с разработчиком, а в ряде случаев предоставляется исходный программный код, компоновочная среда и т.д.

Сертификационные испытания и тематические исследования, регламентированные современной нормативной базой, проводятся путем:

– функционального тестирования на соответствие нормативным и методическим документам или документации (ТУ, формуляр, задание по безопасности);
– структурного декомпозиционного анализа программного обеспечения на отсутствие недекларированных возможностей [15].

Особенностями указанных подходов является следующее.

1. Функциональное тестирование программ касается проверки задекларированных детерминированных механизмов безопасности, т.е. проверяется факт их работы, не касаясь глубокого анализа защищенности. Однако используя личный опыт, квалифицированные эксперты способны построить тесты, позволяющие выявлять некоторые специфические ошибки безопасности проектирования, реализации, конфигураций, прототипов, интерфейсов и т.д.
2. При структурном анализе импортной продукции (если он предусмотрен) проводится, главным образом, проверка полноты/избыточности кода. При проверке программных средств защиты информации, отнесенной к гостайне, также должен проводиться еще статический и динамический анализ, который заключается в выполнении декомпозиции программной системы (формировании и контроле условной части маршрутов).

Однако нормативная база не ограничивает экспертов в использовании дополнительных методов и приемов проверки кода, например: инспекции кода, использовании статических анализаторов, изучении бюллетеней безопасности, организации фаззинг- и стресс-тестирования и др.

3.7.2. Виды тестирования безопасности кода

Опираясь на методологию риск-менеджмента, при тестировании безопасности программного кода следует сформулировать вертикаль факторов информационной безопасности:

$$\{\text{ДЕФЕКТЫ}\} \rightarrow \{\text{УЯЗВИМОСТИ}\} \rightarrow \{\text{УГРОЗЫ}\} \rightarrow \{\text{РИСКИ}\}$$

В названном перечне, с точки зрения анализа кода, первичными являются именно *дефекты безопасности,* которые представляют собой потенциальные уязвимости, влияющие на целостность, доступность, конфиденциальность ресурсов. Дефекты, которые локализованы, описаны, эксплуатируемы, идентифицируются как *уязвимости.* Как правило, дефекты выявляются на этапе аудита безопасности кода, а уязвимости выявляются при сканировании информационной системы (сопоставлении идентифицируемых программ базе описаний уязвимостей или проверке кода программы на наличие сигнатуры уязвимости).

Роль и место указанных факторов в рамках модели управления безопасностью программного обеспечения (ПО) представлены в табл. 3.9.

В настоящее время методы и технологии выявления уязвимостей *не носят универсальный характер* и ориентированы на определенные классы уязвимостей и их причин (дефектов).

На практике выделяют три условных класса дефектов и уязвимостей:

1. «Некорректности программирования», классифицируемые как нефункциональные ошибки, сделанные при кодировании и влияющие на конфиденциальность, целостность, доступность ресурсов. Теоретически такие дефекты могут быть внесены умышленно.

При тестировании обычно полагается, что такие дефекты имеют стохастический характер, т.е. для выявления применяются методы функционального тестирования (обычно, фаззинг-тестирование). К примеру, по заявлению разработчиков, бета-версия Windows 8 прошла 1 миллиард запусков.

В настоящее время развивается направление прикладной верификации кода, позволяющей в рамках статического анализа найти «некорректности программирования»: переполнение буфера, избыточные переменные и объекты и др.

2. Дефекты, идентифицируемые как преднамеренные. Так как такие дефекты связаны с редкими входными данными, то в реальное время их можно выявить только ручными экспертными и полуавтоматизируемыми сигнатурными (эвристическими) методами [16].

Таблица 3.9. Управление безопасностью программ

Фактор	Управление	Контроль	Контрмеры
	Анализ/тестирование		
Дефекты	Выявление, локализация	Инспекционный контроль ПО	Безопасное программирование
Уязвимости	Идентификация, сканирование	Периодическое сканирование, контроль целостности, контроль источников происхождения компонентов и др.	Исправления
Угрозы	Формирование модели угроз	Мониторинг угроз	Обновления, блокировка, фильтрация и др.
Риски	Оценка риска	Оценка остаточного риска	Обработка риска

3. Ранее обнаруженные (известные) уязвимости, которые выявляются методами сканирования и экспертными методами, включающими также сбор и анализ бюллетеней, прототипов и т.д.

При отсутствии исходных данных применяются подходы реверс-инжиниринга и функциональные методы (по принципу «черного ящика»). Реверс-инжиниринг может проводиться путем:

– ретрансляции/дизассемблирования, прогона в отладочном режиме – для машинных и процедурных языков;
– высококачественной декомпиляцией – для языков с промежуточным кодом [16].

Надо понимать, что все методы имеют ограничения по использованию:

– функциональные методы ограничены величиной размерности входных данных, неэффективны при выявлении программных закладок и пригодны для небольших продуктов;

– структурные статические методы, кроме наличия исходных текстов, имеют ограничения на выявление дефектов, связанных с динамикой программы (циклами и т.д.);
– дизассемблирование – реально провести для небольших незащищенных программ;
– ручные экспертные методы предъявляют высокие требования к опыту и знаниям тестировщиков.

Примеры отдельных техник тестирования представлены в табл. 3.10 [15].

Важным моментом при выявлении уязвимостей является сочетание методов тестирования и методов *мониторинга* информационной безопасности (ИБ), включая реверсинг трафика и контроль событий ИБ.

Таким образом, использование различных техник проверки кода в рамках общей организации сертификации импортной программной продукции позволяет выявить ряд дефектов и уязвимостей, статистика по которым представлена ниже.

3.7.3. Типовая статистика выявления уязвимостей в программном обеспечении

В процессе сертификации ПО испытательной лабораторией [15] было выявлено несколько десятков дефектов ПО, идентифицированных как критические уязвимости, и более тысячи дефектов безопасности, которые идентифицировать как преднамеренные не удалось. Под проверки подпала продукция 15 зарубежных производителей из 7 иностранных держав.

Таблица 3.10. Примеры техник тестирования средств защиты информации

Метод тестирования	Основные выявляемые дефекты и уязвимости
Функциональное тестирование	Дефекты реализации функций и ошибки документации
Фаззинг-тестирование	Дефекты реализации интерфейсов данных
Граничное тестирование	Ошибки граничных условий
Нагрузочное тестирование	Ошибки производительности
Стресс-тестирование	Отказ в обслуживании
Профилирование	Недостатки оптимизации кода
Статический семантический анализ (прикладная верификация)	Некорректности кодирования
Статический сигнатурный анализ	Заданные потенциально опасные фрагменты
Статический анализ отсутствия недекларируемых возможностей (НДВ)	«Мертвый код»
Динамический анализ отсутствия НДВ	«Мертвый код»
Мониторинг операционных процессов	Нарушения целостности процессов и ресурсов
Тестирование конфигураций	Ошибки администрирования
Сканирование уязвимостей	Известные опубликованные уязвимости
Тест на проникновение	Известные уязвимости, ошибки конфигурирования
Регрессионное тестирование	Повторные ошибки прошлых версий

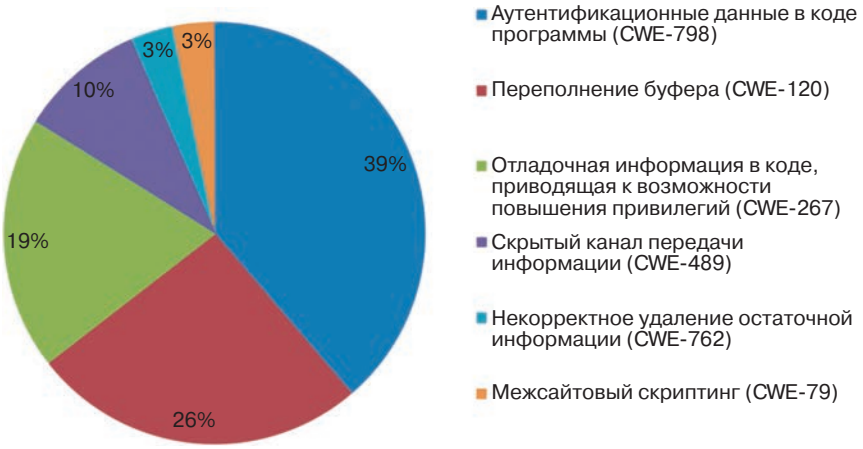

Рис. 3.20. Статистика по типам уязвимости

Статистика по типам уязвимостей

Испытания показали, что в ПО в явном виде встречаются программные закладки, маскируемые под отладочные средства (встроенные учетные записи и мастер-пароли, а также средства удаленного управления). Около 70% выявленных уязвимостей являются именно такими. В то же время зафиксирован ряд дефектов, которые трудно идентифицировать как преднамеренные, однако их можно эксплуатировать при проведении компьютерных атак, например, межсайтовый скриптинг (Cross-Site Scripting – XSS).

Статистика уязвимостей по типам программ

Из зарубежной продукции в рамках сертификации были проверены операционные системы, антивирусные решения, системы обнаружения вторжений, системы хранения информации и автоматизации предприятий, сетевые устройства. Статистика соответствует общемировой – большинство уязвимостей обнаружено в прикладных системах.

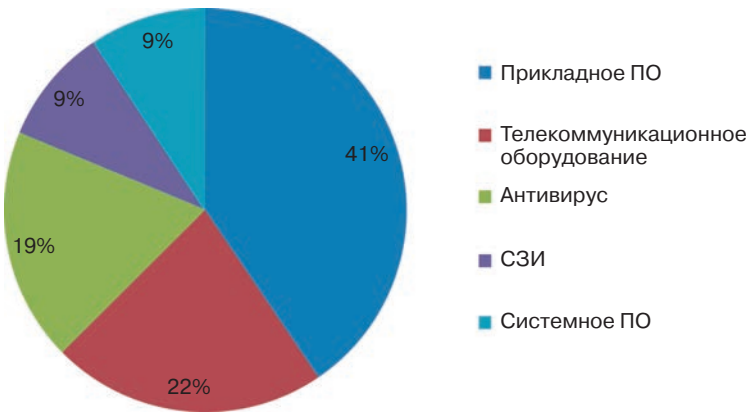

Рис. 3.21. Статистика уязвимостей по типам ПО

Следует отметить, что во всех образцах телекоммуникационного оборудования были обнаружены встроенные учетные записи (CWE-798).

Статистика по методам тестирования

Подавляющее большинство уязвимостей было выявлено методами статического эвристического (сигнатурного) анализа. Для сравнения, следует отметить, что в практике проверки российского ПО доля уязвимостей (главным образом ошибок кодирования), выявленных функциональными методами, существенно выше (до 30%), чем для зарубежного. Это легко можно объяснить наличием сертифицированных систем менеджмента информационной безопасности (СМИБ) на зарубежных предприятиях.

Рис. 3.22. Статистика по методам выявления уязвимостей

Статистика дефектов в открытом коде

Следует указать, что современные программные комплексы включают модули программ с открытым кодом. Исследование показало, что такие программы тоже включают уязвимости. Ниже представленная статистика демонстрирует наличие уязвимостей в открытом коде (рис. 3.22).

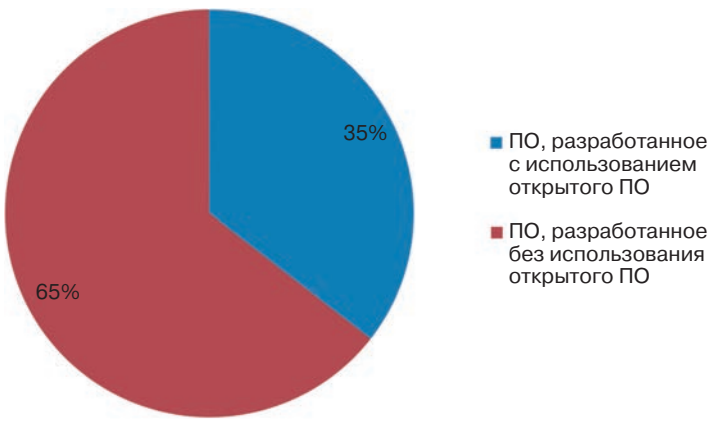

Рис. 3.23. Продукты с открытым кодом тоже содержат уязвимости

Краткие выводы по сертификационной статистике

– большинство импортных программных продуктов имели программные закладки аутентификационного характера и др.;
– подавляющее большинство уязвимостей было выявлено только в случае предоставления исходных кодов;
– большинство уязвимостей зафиксировано на уровне прикладных приложений (а не средств защиты информации);
– количество найденных уязвимостей, не идентифицированных как преднамеренные, зависит от существующей в организации системы менеджмента информационной безопасности (жизненного цикла безопасного производства программ).

Состояние проблемы в зарубежных странах

Так как наличие программных уязвимостей является основой реализации современных кибератак, то интересно познакомиться с зарубежным опытом в области кибербезопасности. Так, например, в США в настоящее время активизируется внимание к активным методам информационного противоборства. Можно отметить ряд тенденций:

1. В области ИБ в США очевиден упор на выявление и эксплуатацию уязвимостей. АНБ в настоящее время демонстрирует привлечение «хакерских» технологий», например, ведет несколько десятков крупномасштабных проектов, включая создание датацентров АНБ, центров обучения по кибербезопасности, привлечение хакеров на работу в АНБ.
2. США традиционно ведет политику «черных списков» для зарубежных разработчиков ПО. В настоящее время в США озабочены противодействием китайским технологиям в военном секторе. В стране имеется демонстративная система поставщиков в DoD.
3. Программное обеспечение в государственных структурах подлежит в обязательном порядке проверкам исходного кода, по результатам которого предусмотрено внедрение методов противодействия недоверенному ПО. В ряде других сфер (например, во всех платежных системах) аудит безопасности ПО «добровольно принудительный».
4. При сертификации критических систем в обязательном порядке проводится тестирование на проникновение (включая аудит безопасности кода). При сертификации средств защиты введено обязательное тестирование на проникновение, а также проведена трансформация методологии испытаний от показателей «качества» к показателям «безопасности». В последнем случае, можно отметить консолидацию европейских стран по изменению процедуры сертификационных испытаний. Например, сертификация во Франции – CSPN, которая заметно проще в вопросах оценки доверия, но отличительной особенностью которой является обязательное тестирование на проникновение.

3.8. Five-Level Problem – пути снижения уязвимостей критических информационных систем

Как мы видим [Белоус А.И. Кибербезопасность объектов топливно-энергетического комплекса. Концепции, методы и средства обеспечения. – Инфра-Инженерия, 2020], современная экономика и национальная безопасность любого государства сегодня зависит от информационно-коммуникационных технологий (Information and communications Technology – ITC). В Национальной стратегии США по обеспечению безопасности киберпространства [18] отмечено, что анализ угроз и снижение существующих уязвимостей в киберпространстве представляет собой особенно сложную задачу из-за большого количества различных категории пользователей киберпространства. Безопасность киберпространства требует согласования совместных действий на нескольких уровнях и со стороны различных групп пользователей, потому что в мире буквально сотни миллионов вычислительных устройств и систем связаны между собой сетью Интернет. Для решения этой проблемы в вышеупомянутой стратегии предлагается ***пятиуровневый подход (A Five-Level Problem)***.

Первый уровень – уровень Домашних Пользователей И Малого Предпринимательства (Home User/Small Busness). «Хотя компьютеры домашних пользователей и не являются частью глобальной критической инфраструктуры, они могут стать частью сети дистанционно управляемых вычислительных устройств, которые затем используются злоумышленниками для атаки на критические инфраструктуры».

Незащищенные домашние компьютеры и компьютеры малого бизнеса сегодня действительно уязвимы для злоумышленников, которые могут их использовать без ведома владельца. Такие машины затем могут быть использованы злоумышленниками для запуска, например, атак типа «отказ в обслуживании» на ключевые интернет-узлы и другие важные предприятия или критические инфраструктуры.

Второй уровень – Крупные Предприятия – Lavge Enterprises (корпорации, государственные учреждения, университеты) как цели для кибератак. Многие такие предприятия являются составной частью других критически важных государственных инфраструктур. По прогнозам международного разведывательного сообщества, сети крупных предприятий будут все чаще становиться мишенью злоумышленников.

Третий уровень – Critical Sector/Infrustructures– это Критически Важные Секторы Инфраструктуры.

Объединение усилий организаций из разных секторов (экономики, обороны, правительства, научных кругов), нацеленность на решение общих проблем кибербезопасности абсолютно необходимы для снижения нагрузки на отдельных пользователей и предприятия. Такое сотрудничество часто приводит к созданию общих институтов и механизмов, которые, в свою очередь, могли бы иметь киберуязвимости, эксплуатация которых могла бы непосредственно влиять на деятельность предприятий-членов и сектора в целом. Отдельные предприятия также могут существенно снизить киберриски, участвуя в группах, которые разрабатывают лучшие практики, оценивают инновационные технологические предложения в области киберзащиты, проводят сертификацию продуктов и услуг, а также осуществляют обмен информацией.

Четвертый уровень – National Issues and Vulnerabilities – это национальные проблемы и критические уязвимости систем энергетики, обороны, транспорта. Все секторы национальной экономики имеют общий доступ к интернету. Соответственно, все они находятся под угрозой, если используемые ими механизмы (например, протоколы и маршрутизаторы) не являются безопасными. Имеются слабые места (уязвимости) в широко используемом программном обеспечении, а аппаратные продукты также могут создавать на национальном уровне проблемы, требующие скоординированной работы по исследованию и разработке усовершенствованных защитных технологий. Кроме того, проблема отсутствия обученных и сертифицированных специалистов по кибербезопасности также заслуживает внимания на национальном уровне.

Пятый уровень – Global (Глобальный).

Интернет (всемирная паутина) – это глобальная планетарная информационная сеть взаимосвязанных информационных систем. Международные общие стандарты в принципе могут обеспечить достаточно безопасное взаимодействие между компьютерными системами всего мира. Это означает, что пути и методы решения проблемы кибербезопасности, принятые на одном континенте, потенциально могут повлиять на безопасность компьютеров на другом континенте. Международное сотрудничество необходимо не только *для обмена разнообразной информацией*, связанной с киберпространством, но и для того, чтобы организовать эффективную *борьбу с киберпреступниками*. Совершенно очевидно, что без такого сотрудничества коллективная способность обнаруживать, сдерживать и сводить к минимуму последствия кибератак будет значительно уменьшена.

Литература к главе 3

1. https://russianelectronics.ru/rossiyane-vyyavili-neustranimuyu-uyazvimost-vo-vseh-proczessorah-intel-poslednih-let/
2. https://www.anti-malware.ru/threats/programs-vulnerability
3. https://ru.wikipedia.org/wiki/%D0%A3%D1%8F%D0%B7%D0%B2%D0%B8%D0%BC%D0%BE%D1%81%D1%82%D1%8C_(%D0%BA%D0%BE%D0%BC%D0%BF%D1%8C%D1%8E%D1%82%D0%B5%D1%80%D0%BD%D0%B0%D1%8F_%D0%B1%D0%B5%D0%B7%D0%BE%D0%BF%D0%B0%D1%81%D0%BD%D0%BE%D1%81%D1%82%D1%8C)
4. https://ru.wikipedia.org/wiki/SiXSS
5. https://ru.wikipedia.org/wiki/%D0%9F%D0%BE%D0%B2%D1%8B%D1%88%D0%B5%D0%BD%D0%B8%D0%B5_%D0%BF%D1%80%D0%B8%D0%B2%D0%B8%D0%BB%D0%B5%D0%B3%D0%B8%D0%B9
6. https://ru.wikipedia.org/wiki/%D0%A3%D1%8F%D0%B7%D0%B2%D0%B8%D0%BC%D0%BE%D1%81%D1%82%D1%8C_%D0%BD%D1%83%D0%BB%D0%B5%D0%B2%D0%BE%D0%B3%D0%BE_%D0%B4%D0%BD%D1%8F
7. https://ru.wikipedia.org/wiki/%D0%91%D0%B5%D0%B7%D0%BE%D0%BF%D0%B0%D1%81%D0%BD%D0%BE%D1%81%D1%82%D1%8C_%D0%B4%D0%BE%D1%81%D1%82%D1%83%D0%BF%D0%B0_%D0%BA_%D0%BF%D0%B0%D0%BC%D1%8F%D1%82%D0%B8
8. https://ru.wikipedia.org/wiki/%D0%A1%D0%BE%D1%81%D1%82%D0%BE%D1%8F%D0%BD%D0%B8%D0%B5_%D0%B3%D0%BE%D0%BD%D0%BA%D0%B8

9. https://searchinform.ru/informatsionnaya-bezopasnost/osnovy-ib/ugrozy-informatsionnoj-bezopasnosti/
10. https://encyclopedia.kaspersky.ru/knowledge/software-vulnerabilities/
11. https://ru.wikipedia.org/wiki/%D0%9C%D0%B5%D0%B6%D1%81%D0%B0%D0%B9%D1%82%D0%BE%D0%B2%D0%B0%D1%8F_%D0%BF%D0%BE%D0%B4%D0%B4%D0%B5%D0%BB%D0%BA%D0%B0_%D0%B7%D0%B0%D0%BF%D1%80%D0%BE%D1%81%D0%B0
12. https://ru.wikipedia.org/wiki/Shatter_attack
13. Марков А.С., Цирлов В.Л. Опыт выявления уязвимостей в зарубежных программных продуктах // Вопросы кибербезопасности. – 2013. – № 1 (1). – С. 42–48.
14. Марков А.С., Цирлов В.Л. Сертификация программ: мифы и реальность // Открытые системы. СУБД. – 2011. – № 6. – С. 26–29.
15. Марков А.С., Цирлов В.Л., Барабанов А.В. Методы оценки несоответствия средств защиты информации. – М.: Радио и связь, 2012. – 192 с.
16. Марков А.С., Фадин А.А. Статический сигнатурный анализ безопасности программ // Программная инженерия и информационная безопасность. – 2013. № 1 (1). – С. 50–56.
17. Барабанов А.В., Марков А.С., Фадин А.А. Сертификация программ без исходных текстов // Открытые системы. СУБД. – 2011. – № 4. – С. 38–41.
18. Department of Defense Fiscal Year (FY) 2014 President's Budget Submission. – US Department of the Army, 2013. –679 p.
19. Винокуров А.В. Анализ уязвимостей комплексов с беспилотными летательными аппаратами и классификация угроз безопасности циркулирующей в них информации // i-methods. – 2016. – Т. 8. – № 1. – С. 5–9. – Изд-во: ООО «Институт инноваций и наукоемких технологий» (Санкт-Петербург).
20. Круглов Е. Перспективы развития американских авиационных средств РЭБ и тактика их применения в современных вооруженных конфликтах // Зарубежное военное обозрение. – 2014. № 2 (803). – С. 57–63.
21. Цветнов В.В., Демин В.П., Куприянов А.И. Радиоэлектронная борьба: радиомаскировка и помехозащита: Учебное пособие. – М.: Изд-во МАИ, 1999. – 240 с.
22. Бардаев Э.А. Винокуров А.В., Задвижкин А.А, Колованов А.В., Лисицын В.В. Принципы и модели построения системы защиты информации в робототехнических комплексах от внешних деструктивноинформационных воздействий // Вопросы кибербезопасности. – 2019. – № 6 (34).
23. Климов С.М. Методы и модели противодействия компьютерным атакам. – Люберцы: КАТАЛИТ, 2008. – 316 с.
24. Винокуров А.В. Бухонский М.И., Дейкун Г.И. Защита командно-программной информации управления беспилотными летательными аппаратами // Инновационные технологии в образовательном процессе. Материалы XIX Всероссийской научно-практической конференции. – Краснодар: КВВАУЛ, 2017. – С. 38–45.
25. Дейкун Г.И. Инновационные технологии в образовательном процессе // Материалы XIX Всероссийской научно-практической конференции. – Краснодар: КВВАУЛ, 2017. – С. 38–45.
26. Овчаренко М.В., Винокуров А.В. Методологические основы построения имитоустойчивой аппаратуры передачи данных // Информационные ресурсы России. – 2015. – № 5. – С. 38–41.

27. Патент 164498 Российская Федерация, МПК7 G 09 C 1/00, H 03 M 13/23. Устройство имитостойкого кодирования. – А.В. Винокуров, М.В. Овчаренко; заявитель и патентообладатель КВВУ имени генерала армии С.М. Штеменко. – № 20155141723; заявл. 30.09.2015; опубл. 10.09.2016. – Бюллетень № 25. – 2 с.
28. Сырямкин В.И., ШидловскийВ.С. Корреляционно-экстремальные радионавигационные системы.– Томск: Изд-во Томского ун-та, 2010. – 316 с.
29. Цветков В.В., Устинов А.А., Оков И.Н. Устойчивый к канальным ошибкам видеокодек подвижных изображений на основе трехмерного ортогонального преобразования с обеспечением конфиденциальности и аутентификации передаваемых видеоданных // Информация и космос. – 2015. – № 2. – С. 52–59.
30. Nasrullah, Sang J., Akbar M.A., Cai B., Xiang H., Hu H. Joint image compression and encryption using IWT with SPIHT, Kd-tree and chaotic maps // Applied Sciences. – 2018. – Vol. 8 (10). doi: 10.3390/app8101963
31. Agarwal S. Secure Image Transmission Using Fractal and 2D-Chaotic Map // Journal of Imaging. – 2018. – Vol. 4 (1). doi:10.3390/ jimaging4010017
32. Younas I., Khan M. A New Efficient Digital Image Encryption Based on Inverse Left Almost Semi Group and Lorenz Chaotic System // Entropy. – 2018. – Vol. 20 (12). doi:10.3390/e20120913
33. Патент 2595953 Российская Федерация, МПК7 H 04 L 9/00 Способ арифметического кодирования с шифрованием. – В.Б. Васильев [и др.]; заявитель и патентообладатель – Акционерное общество «Концерн радиостроения «Вега»; заявл. 04.08.2015; опубл. 27.08.2016. – Бюллетень № 24.
34. Косьянчук В.В., Сельвесюк Н.И, Зыбин Е.Ю., Хамматов Р.Р., Карпенко С.С. Концепция обеспечения информационной безопасности бортового оборудования воздушного судна // Вопросы кибербезопасности. –2018. – № 4 (28).