

ЭКЗАМЕН 2022/2023

ОРГАНИЗАЦИОННОЕ И ПРАВОВОЕ ОБЕСПЕЧЕНИЕ ИБ

Преподаватели:

Струков Владимир Ильич (Модуль 1)

Князева Маргарита Владимировна (Модуль 2)

Члены комиссии:

Ельчанинова Наталья Борисовна, к.т.н., доцент кафедры БИТ - председатель

Горбунов Александр Валерьевич, к.т.н., доцент кафедры ИБТКС

Котов Эдуард Михайлович, старший преподаватель кафедры ИАСБ

Семаков Артем Александрович, руководитель проектного офиса «Dunice»

Пример билета

МИНОБРНАУКИ РОССИИ

Федеральное государственное автономное образовательное
учреждение высшего образования
«ЮЖНЫЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

ЭКЗАМЕНАЦИОННЫЙ БИЛЕТ №1

по дисциплине:

«ОРГАНИЗАЦИОННОЕ И ПРАВОВОЕ ОБЕСПЕЧЕНИЕ
ИНФОРМАЦИОННОЙ БЕЗОПАСНОСТИ»

Институт компьютерных технологий и информационной безопасности

Направление/специальность: 10.03.01, 10.05.02, 10.05.03, 10.05.05

1. Правовые основы защиты информации ограниченного доступа. Формирование перечня сведений, составляющих коммерческую тайну. Актуальность защиты коммерческой информации, угрозы и методы защиты, правовая защита.

2. Источники поступления информации об угрозах. Threat Intelligence: цели, задачи, этапы. Платформы Threat Intelligence. Индикаторы компрометации (Indicators of Compromise, IoC). Пирамида индикаторов компрометации в зависимости от сложности получения данных.

Составитель _____ В.И. Струков

Составитель _____ М.В. Князева

« ____ » _____ 2022/2023 г.

1. Системный подход к управлению ИБ. Политики безопасности. Уровни.

Законодательный уровень:

Обзор нормативно-правовых актов РФ в области информационной безопасности

Административный уровень:

Разработка политик информационной безопасности (организационно-технические и режимные меры).

Политика информационной безопасности — набор законов, правил, практических рекомендаций и практического опыта, определяющих управленческие и проектные решения в области ЗИ. На основе ПИБ строится управление, защита и распределение критичной информации в системе. Она должна охватывать все особенности процесса обработки информации, определяя поведение ИС в различных ситуациях.

Для конкретной ИС политика безопасности должна быть индивидуальной. Она зависит от технологии обработки информации, используемых программных и технических средств, структуры организации и т.д.

Процедурный уровень: *перечень основных организационно-технические мероприятия по защите информации.*

Технический уровень: *конкретные меры и их реализация.*

1. Системный подход к управлению ИБ. Политики безопасности. Определение, сферы приложения, верхний, средний, нижний уровни политик безопасности.

Верхний уровень: документация на уровне руководства.

Средний уровень: частные политики безопасности, политики использования средств криптозащиты, политики антивирусной защиты, политики мониторинга и управления инцидентами и т.п.

Нижний уровень: действия по обеспечению ИБ на уровне сетевых сервисов, руководства, инструкции, регламенты, правила администрирования.

2. Защита критической информационной инфраструктуры России (КИИ). ГосСОПКА. Определение КИИ.

Приказ Федеральной службы безопасности Российской Федерации от 24.07.2018 № 367 "Об утверждении Перечня информации, представляемой в государственную систему обнаружения, предупреждения и ликвидации последствий компьютерных атак на информационные ресурсы Российской Федерации и Порядка представления информации в государственную систему обнаружения, предупреждения и ликвидации последствий компьютерных атак на информационные ресурсы Российской Федерации".

Цели и задачи проекта ГосСОПКА. Основные положения и нормативные акты, регулирующие деятельность по обнаружению, предупреждению и ликвидации последствий кибератак. Перечень мероприятий для субъектов КИИ.

2. Защита критической информационной инфраструктуры России (КИИ). ГосСОПКА. Определение КИИ.

В августе 2013 г. ФСБ представило проекты законов, регулирующих защиту ИТ-инфраструктуры критически важных объектов в России. Они предусматривают «дополнительные обременения» для частных владельцев критически важных ИТ-систем, а также уголовную ответственность за нарушения именно в данной области.

Под критической информационной инфраструктурой РФ (КИИ) подразумевается совокупность автоматизированных систем управления производственными и технологическими процессами критически важных объектов РФ и обеспечивающих их взаимодействие информационно-телекоммуникационных сетей, а также ИТ-систем и сетей связи, предназначенных для решения задач государственного управления, обеспечения обороноспособности, безопасности и правопорядка.

ГосСОПКА – это государственная система обнаружения, предупреждения и ликвидации последствий компьютерных атак на информационные ресурсы Российской Федерации.

2. Защита критической информационной инфраструктуры России (КИИ). ГосСОПКА. Определение КИИ.

Согласно указу Президента РФ от 22.12.2017 №620, ГосСОПКА предназначена для выполнения четырех основных задач:

- прогнозирование в области информационной безопасности РФ;
- взаимодействие организаций-владельцев информационных ресурсов, в том числе – субъектов критической информационной инфраструктуры, в ходе работ по обнаружению, предупреждению и ликвидации последствий кибератак;
- контроль степени защищенности информационных ресурсов РФ от кибератак;
- расследование компьютерных инцидентов.

Ответственность за функционирование системы ГосСОПКА лежит на Федеральной службе безопасности РФ. В свою очередь, организации с критической инфраструктурой (КИИ) должны развернуть у себя центры ГосСОПКА в соответствии с нормативной базой, разработанной ФСБ России.

ФЗ-187 от 26.07.2017 « О безопасности критической информационной инфраструктуры Российской Федерации».

Статья 7. Категорирование объектов критической информационной инфраструктуры: три категории значимости. Критерии значимости: Социальная значимость (5 показателей); Политическая значимость (2 показателя); Экономическая значимость (3 показателя); Экологическая значимость (1 показатель); Значимость для обеспечения обороны страны, безопасности государства и правопорядка (3 показателя).

Статья 8. Реестр значимых объектов критической информационной инфраструктуры.

Статья 10. Система безопасности значимого объекта критической информационной инфраструктуры.

2. Защита критической информационной инфраструктуры России (КИИ). ГосСОПКА. Определение КИИ.

Этапы реализации требований приказа ФСТЭК РФ №239:

1. Первичный сбор данных

На начальном этапе организация определяет, является ли она субъектом КИИ. Для этого она готовит перечни ИС, АСУ, ИТС и определяет сферы функционирования каждой из них.

Результат этой работы — вывод о необходимости (или отсутствии необходимости) выполнения организацией требований 187-ФЗ.

2. Проведение категорирования (Постановление Правительства Российской Федерации: от 08.02.2018 г. № 127 «Об утверждении Правил категорирования объектов критической информационной инфраструктуры Российской Федерации, а также перечня показателей критериев значимости объектов критической информационной инфраструктуры Российской Федерации и их значений»)

Организация создает комиссию по категорированию, определяет объекты категорирования, а затем определяет категории значимости для объектов КИИ.

После этого она согласует со ФСТЭК России перечень объектов КИИ с присвоенными (либо не присвоенными) категориями значимости.

3-1. Для субъектов КИИ, не имеющих значимые объекты КИИ

В том числе это относится к организациям, у которых нет значимых объектов КИИ, так как организация должна обеспечить соответствие 9 статьи ФЗ-187 (часть 2), в частности незамедлительно информировать о компьютерных инцидентах федеральные органы исполнительной власти.

3-2. Для субъектов КИИ, имеющих значимые объекты КИИ

Разработать план реагирования на компьютерные инциденты.

Выбрать способ реализации функций ГосСОПКА (<http://cert.gov.ru/>) Национальный координационный центр по компьютерным инцидентам)

Разработка плана реагирования на компьютерные инциденты (КИ) и принятия мер по ликвидации последствий компьютерных атак (КА) в соответствии с требованиями приказа ФСБ России №282.

Создание системы безопасности значимых объектов КИИ в соответствии с требованиями приказов ФСТЭК России №235, 239.

2. Защита критической информационной инфраструктуры России (КИИ). ГосСОПКА. Определение КИИ.

Каждому значимому объекту КИИ РФ присваивается регистрационный номер состоящий из групп цифр и прописных букв, разделенных косыми чертами, который имеет вид: XXXXXX/X/XX/X

- порядковый номер;
- федеральный округ, на территории которого находится значимый объект КИИ;
- сфера (область) деятельности, в которой функционирует значимый объект КИИ;
- тип значимого объекта КИИ.

**Приказ ФСТЭК России от 6 декабря 2017 г. № 227
Об утверждении Порядка ведения реестра значимых объектов КИИ РФ.*

Требования приказа ФСТЭК РФ №239

14 сентября 2020 года был опубликован Приказ Федеральной службы по техническому и экспортному контролю (ФСТЭК) о внесении изменений в **Требования по обеспечению безопасности значимых объектов критической информационной инфраструктуры РФ (КИИ)**.

29.3. Прикладное программное обеспечение, планируемое к внедрению в рамках создания (модернизации или реконструкции, ремонта) значимого объекта и обеспечивающее выполнение его функций по назначению (далее - программное обеспечение), должно соответствовать следующим требованиям по безопасности:

29.3.1. Требования по безопасной разработке программного обеспечения:

- наличие руководства по безопасной разработке программного обеспечения;
- проведение анализа угроз безопасности информации программного обеспечения;
- наличие описания структуры программного обеспечения на уровне подсистем и результатов сопоставления функций программного обеспечения и интерфейсов, описанных в функциональной спецификации, с его подсистемами (для программного обеспечения, планируемого к применению в значимых объектах 1 категории значимости).

29.3.2. Требования к испытаниям по выявлению уязвимостей в программном обеспечении:

- проведение статического анализа исходного кода программы;
- проведение фаззинг-тестирования программы, направленного на выявление в ней уязвимостей;
- проведение динамического анализа кода программы (для программного обеспечения, планируемого к применению в значимых объектах 1 категории значимости).

29.3.3. Требования к поддержке безопасности программного обеспечения:

- наличие процедур отслеживания и исправления обнаруженных ошибок и уязвимостей программного обеспечения;
- определение способов и сроков доведения разработчиком (производителем) программного обеспечения до его пользователей информации об уязвимостях программного обеспечения, о компенсирующих мерах по защите информации или ограничениях по применению программного обеспечения, способов получения пользователями программного обеспечения его обновлений, проверки их целостности и подлинности;
- наличие процедур информирования субъекта критической информационной инфраструктуры об окончании производства и (или) поддержки программного обеспечения (для программного обеспечения, планируемого к применению в значимых объектах 1 категории значимости).

3. Управление информационной безопасностью. Построение СОИБ и СУИБ на предприятии.

Техническое проектирование является необходимым условием для реализации комплексного подхода к созданию систем обеспечения информационной безопасности. В отсутствии технического проекта возможно лишь реализация фрагментарных мер и механизмов безопасности, за счет которых в современных условиях невозможно решение основных вопросов обеспечения информационной безопасности.

Целью проектирования системы обеспечения информационной безопасности (СОИБ) является выработка рекомендаций, организационных и технических решений по обеспечению безопасности информационных ресурсов хранимых, обрабатываемых и передаваемых по каналам связи в компьютерных сетях и информационных системах организации.

СОИБ в современных организациях имеет сложную многокомпонентную, многоуровневую, территориально и логически распределенную архитектуру. Компоненты СОИБ очень тесно интегрированы в информационную инфраструктуру организации.

3. Управление информационной безопасностью. Построение СОИБ и СУИБ на предприятии.

В состав СОИБ обычно входят следующие компоненты и подсистемы, тесно интегрированные между собой и с другими компонентами ИТ-инфраструктуры:

- подсистема защиты периметра сети;*
- подсистема обеспечения безопасности межсетевых взаимодействий;*
- подсистема мониторинга и аудита безопасности;*
- подсистема обнаружения и предотвращения атак;*
- подсистема резервного копирования и восстановления данных;*
- подсистема анализа защищенности и управления политикой безопасности;*
- подсистема контроля целостности данных;*
- криптографическая подсистема;*
- инфраструктура открытых ключей;*
- подсистема защиты от вредоносного ПО;*
- подсистема фильтрации контента и предотвращения утечки конфиденциальной информации;*
- подсистема установки обновлений ПО;*
- подсистема администрирования безопасности.*

3. Управление информационной безопасностью. Построение СОИБ и СУИБ на предприятии.

Условно АСУИБ можно разделить на три функциональных уровня:

- уровень сбора,
- уровень ядра,
- уровень управления.



3. Управление информационной безопасностью. Построение СОИБ и СУИБ на предприятии.

Условно АСУИБ можно разделить на три функциональных уровня:

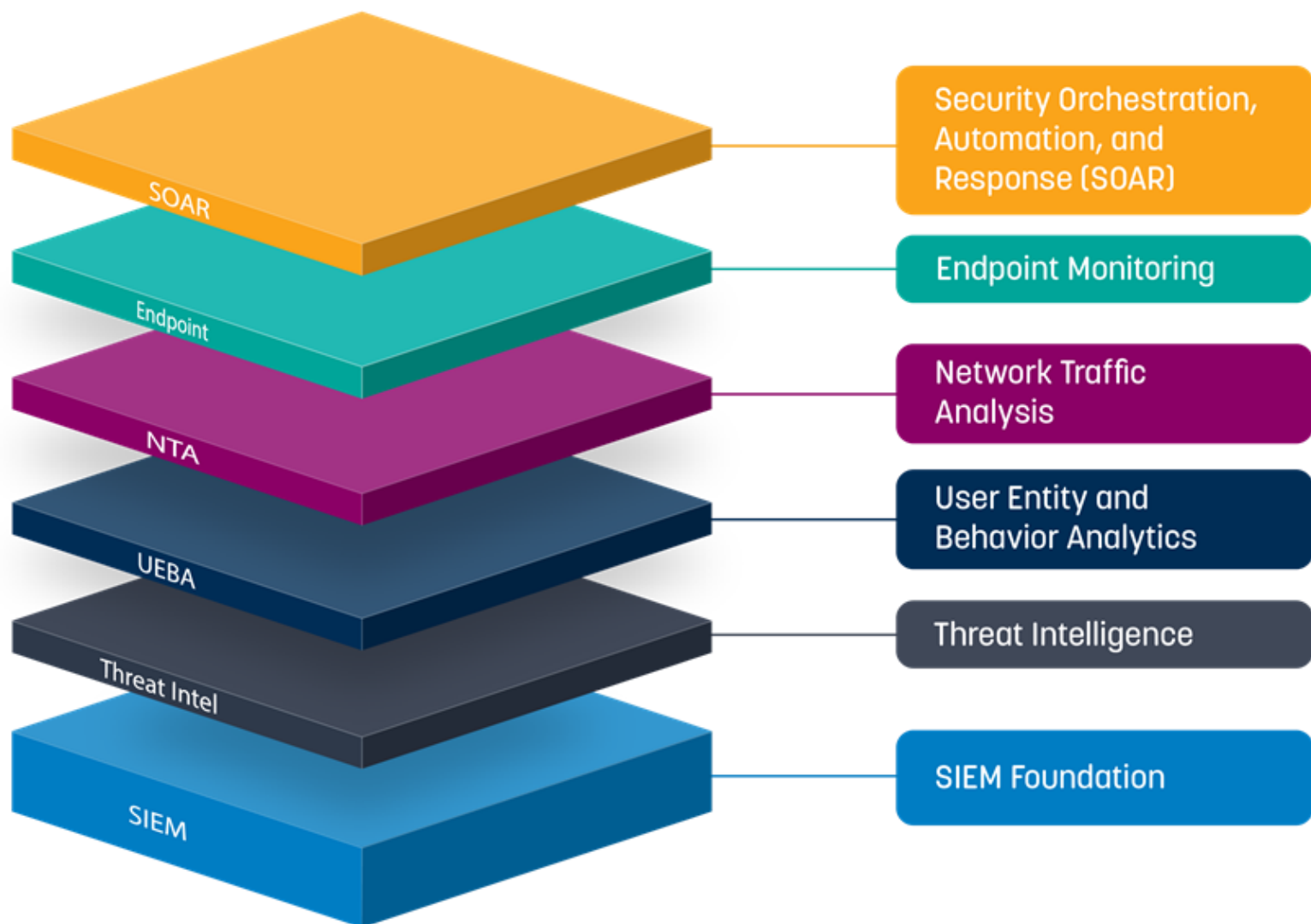
- уровень сбора,
- уровень ядра,
- уровень управления.

Уровень ядра. Здесь происходит сбор, анализ и обнаружение корреляции между событиями безопасности, поступающими от систем обеспечения информационной безопасности и различных компонент ИТ-инфраструктуры. Работа уровня ядра настраивается в соответствии с политикой ИБ, принятой в организации.

Обычно уровень ядра строится на основе одной из представленных на рынке SIEM-систем (Security Information and Event Management).

SIEM-система — это программный или программно-аппаратный комплекс, его основной функцией является обработка большого количества событий безопасности, порождаемых различными системами, и генерация на их основе инцидентов информационной безопасности. От миллионов сообщений и log-записей осуществляется переход к небольшому количеству настоящих инцидентов ИБ.

3. Управление информационной безопасностью. Построение СОИБ и СУИБ на предприятии.



3. Управление информационной безопасностью.

Построение СОИБ и СУИБ на предприятии.

Уровень ядра предназначен для сбора, анализа и корреляции событий безопасности, поступающих от систем безопасности и выполняет следующие функции:

- сбор событий от систем обеспечения информационной, физической и экономической безопасности, ИТ-систем;
- анализ событий в соответствии с предустановленными правилами корреляции для генерации инцидентов;
- анализ событий в соответствии с пользовательскими правилами корреляции для генерации инцидентов;
- автоматическая генерация инцидентов при срабатывании одного из соответствующих правил корреляции, которая может основываться как на отдельном событии, так и на агрегации и сопоставлении разнородных событий и сетевых потоков;
- создание пользовательских правил корреляции с использованием внутреннего помощника ядра;
- поддержка следующих типов пользовательских правил:

- правила, выполняющие проверку совпадений над событиями безопасности в режиме реального времени;
- правила, выполняющие проверку совпадений над сетевыми потоками в режиме реального времени;
- правила, выполняющие проверку совпадений над атрибутами, общими для -событий безопасности и сетевых потоков;
- правила, срабатывающие при внесении изменений в открытые инциденты;
- правила, выполняющие анализ аномалий в сетевых потоках, в том числе поведенческий анализ;

- контроль и расследование инцидентов информационной безопасности в режиме реального времени или выполнение запросов к историческим данным;
- развитая система запросов и фильтрации данных о событиях информационной безопасности и сетевых потоках;
- централизованное управление ядром из единой веб-консоли:

- возможность в режиме реального времени просматривать информационные потоки от подключенных источников событий безопасности;
- возможность использования предустановленных отчетов;
- возможность создания пользовательских отчетов;
- возможность создания автоматической генерации отчетов по графику и рассылку по заранее заданным адресам электронной почты в форматах PDF, HTML, RTF, XLS, XML;
- возможность настройки оповещения о произошедших инцидентах;
- возможность настройки автоматического обновления ядра;
- возможность настройки резервного копирования и восстановления ядра;
- управление правами пользователей при помощи ролей;
- возможность автоматической инвентаризации сетевых узлов.

3. Управление информационной безопасностью. Построение СОИБ и СУИБ на предприятии.

Уровень сбора.

Уровень сбора предназначен для сбора, нормализации и отправки на уровень ядра событий безопасности, поступающих от антивирусных приложений, систем защиты от несанкционированного доступа, различных систем мониторинга и обнаружения вторжений, систем межсетевого экранирования, защиты почты, DLP, сетевого оборудования, серверов и т. д.

Уровень сбора продукта может состоять из следующих модулей:

Агент сбора

предназначен для сбора, нормализации и отправки на уровень ядра событий безопасности, поступающих от информационных систем и систем защиты.

Агент доступности

предназначен для проверки сетевой доступности удаленных узлов и сервисов.

Агент инвентаризации

предназначен для предоставления актуальной информации обо всех программно-аппаратных компонентах комплексной системы информационной безопасности.

Агент инвентаризации и контроля целостности (Агент ИКЦ)

предназначен для автоматизации процесса инвентаризации и контроля целостности состава аппаратных средств, системного и прикладного программного обеспечения, установленного на серверы и рабочие станции.

3. Управление информационной безопасностью. Построение СОИБ и СУИБ на предприятии.

Уровень управления.

Уровень управления АСУИБ может иметь различный вид. В самом простом случае это набор организационных мер, организационно-распорядительной документации и те средства визуализации и отчетности, которые предоставляет SIEM-система.

Более сложный вариант уровня управления предполагает создание программной надстройки, которая позволяет автоматизировать многие функции управления информационной безопасностью, такие как визуализация данных и создание отчетов о состоянии ИБ на предприятии, управление и хранение истории инцидентов ИБ, ведение реестров информационных активов, поддержание в актуальном состоянии документации в области управления ИБ и т. д. Такой уровень управления интегрируется или основывается на других средствах автоматизации, используемых на предприятии (например, на системе Service Desk), либо же строится отдельно.

Некоторые SIEM-системы имеют встроенные средства интеграции с корпоративным Service Desk, для других придется разрабатывать их дополнительно.

3. Управление информационной безопасностью.

Построение СОИБ и СУИБ на предприятии.

Основные задачи автоматизации, выполняемые **Уровнем управления**, в зависимости от комплектации, это:

Управление информационными активами организации:

- поддержание в актуальном состоянии информации о бизнес-процессах, информационных системах, технических сервисах, ПО и информации, используемым в организации;
- инвентаризация и контроль целостности существующих активов организации, в том числе управление агентами инвентаризации и контроля целостности.

Управление жизненным циклом инцидентов информационной безопасности в организации:

- оповещение о возникновении инцидентов информационной безопасности;
- визуализация возникших инцидентов информационной безопасности на гео-карте или карте сети;
- назначение ответственного за инцидент и контроль принятия решения по возникшим инцидентам информационной безопасности;

Управление записями/логами информационной безопасности. Структурированное хранение и индексация логов записей от различных систем, поиск данных и расследование инцидентов ИБ, сбор и хранение улик в соответствии с законодательством УК РФ;

Управление соответствием требованиям информационной безопасности организации:

- законодательным требованиям;
- заданным ключевым показателям эффективности;
- внутренним и отраслевым стандартам организации;
- международным стандартам.

Управление рисками информационной безопасности организации за счет автоматизации:

- подготовки модели угроз и модели нарушителя;
- подготовки перечня информационных активов и объектов среды;
- определения источников угроз для каждого из типов объектов среды;
- проведения оценки возможности реализации угроз информационной безопасности и тяжести последствий;
- оценки рисков информационной безопасности с использованием качественной или количественной шкалы;
- подготовки документов, фиксирующих результаты оценки рисков.

Управление знаниями. Помощь в принятии решений по управлению ИБ организации, в том числе по возникающим инцидентам информационной безопасности за счет:

- автоматически регулярно пополняемой базы знаний и рекомендаций, имеющей отраслевой характер;
- интеллектуального принятия решений на основе ранее предпринятых действий и обучаемой математической модели;

4. Угрозы информационной безопасности.

Определение угрозы, источники угрозы, уязвимость, критичность реализации угрозы, окно безопасности, модель угроз.

Критерии классификации угроз (по аспекту ИБ – CIA, STRIDE, Гексада Паркера, 5A). Использование DFD диаграмм для построения модели угроз информационной безопасности. Microsoft Threat Modeling Tool. Примеры.

<https://learn.microsoft.com/ru-ru/azure/security/develop/threat-modeling-tool>

4. Угрозы информационной безопасности.

Стандартная модель безопасности информации «CIA»:

1.конфиденциальность

(confidentiality) - сохранение информации в тайне, невозможность раскрытия информации без согласия заинтересованных сторон;

2.целостность (integrity) - непротиворечивость и правильность информации, защита информации от неавторизованной модификации;

3.доступность (availability) - обеспечение наличия информации и работоспособности основных услуг для пользователя в нужное для него время.

Модель угроз «Гексада Паркера» **Parkerian Hexad** вводит еще три **состояния:**

4.подлинность (authenticity) - в применении к пользователю соответствие участника взаимодействия своему имени;

5.управляемость, или владение (possession or control) - гарантия того, что законный владелец является единственным лицом, во власти которого изменить информацию или получить к ней доступ на чтение

6.полезность (utility) - удобство доступа; нахождение информации в такой форме, что ее владелец не должен тратить неоправданных усилий.

4. Угрозы информационной безопасности.

Модель безопасности информации «5A»:

1. Authentication

(аутентификация: кто ты?)

2. Authorization (авторизация: что тебе можно делать?)

3. Availability (доступность: можно ли получить работать с данными?)

4. Authenticity (подлинность: не повреждены ли данные злоумышленником?)

5. Admissibility (допустимость: являются ли данные достоверными, актуальными и полезными?)

Модель угроз «STRIDE+LM» компонент, используемой Microsoft методологии SDL (Secure Development Lifecycle).

1. Spoofing (аутентификация)

2. Tampering (изменение, целостность)

3. Repudiation (отказ от ответственности)

4. Information Disclosure (утечка данных)

5. Denial of Service (отказ в обслуживании)

6. Elevation of Privilege (захват привилегий, авторизация)

+

7. Lateral Movement (горизонтальное или боковое движение)

4. Угрозы информационной безопасности.

Элементы модели STRIDE+LM

Спуфинг (Spoofing) Предполагает незаконный доступ к данным пользователя (включая имя пользователя и пароль), используемыми для аутентификации, и их последующего применения.

Незаконное изменение (Tempering) Предполагает вредоносное изменение данных. Примеры включают несанкционированные изменения, внесенные в постоянные данные, например хранящиеся в базе данных, а также изменение данных при их передаче между двумя компьютерами через открытую сеть, например Интернет.

Отказ (Repudiation) Речь идет о пользователях, которые отрицают выполнение действий, если другие пользователи не могут доказать обратное. Например, пользователь может выполнить незаконную операцию в системе, где отсутствует возможность трассировки запрещенных операций. Неподдельность означает способность системы учитывать угрозы отказа.

Раскрытие информации (Information Disclosure) Предполагает раскрытие сведений пользователям, которые не должны иметь к ним доступ, например возможность прочитать файл, к которому этим пользователям не предоставлен доступ, или возможность для злоумышленника считать данные при их передаче между двумя компьютерами.

Отказ в обслуживании (Denial of Service) DoS-атака — это буквально отказ в обслуживании для допустимых пользователей, что делает веб-сервер временно недоступным или непригодным для использования. Следует защищаться от определенных типов угроз DoS-атак, просто чтобы повысить доступность и надежность системы.

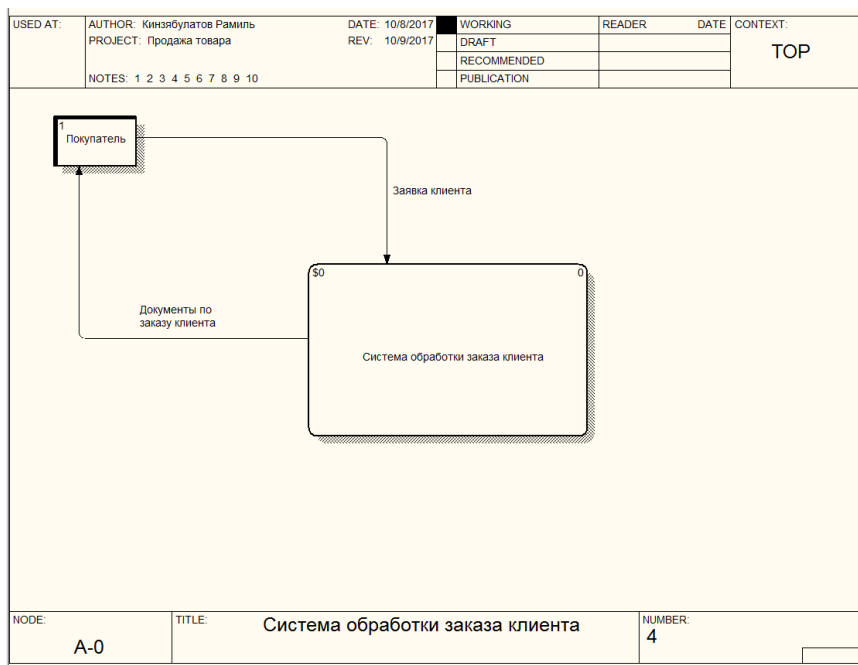
Несанкционированное получение привилегий (Elevation of Privilege) Непривилегированный пользователь получает привилегированный доступ, т. е. достаточные полномочия для нарушения работоспособности или уничтожения всей системы. Угроза повышения привилегий включает ситуации, в которых злоумышленник эффективно обходит все средства защиты системы, сам становясь частью надежной системы. Тем самым он создает действительно опасную ситуацию.

Горизонтальное перемещение (Lateral Movement) включает методы получения противником доступа и контроля над удаленными системами, подключенными к атакованной сети, а так же, в некоторых случаях, запуска вредоносных инструментов на удаленных системах, подключенных к атакованной сети.

4. Угрозы информационной безопасности. DFD диаграммы для моделирования угроз ИБ.

DFD — общепринятое сокращение от англ. **data flow diagrams** — диаграммы потоков данных. Так называется методология графического структурного анализа, описывающая внешние по отношению к системе источники и адресаты данных, логические функции, потоки данных и хранилища данных, к которым осуществляется доступ.

Диаграмма потоков данных (data flow diagram, DFD) — один из основных инструментов структурного анализа и проектирования информационных систем, существовавших до широкого распространения UML.



4. Угрозы информационной безопасности. DFD диаграммы для моделирования угроз ИБ.

STRIDE — это аббревиатура от названий угроз нарушения каждого из перечисленных состояний ИС:

- **spoofing**, аутентичности;
- **tampering**, целостности;
- **repudiation**, апеллируемости;
- **information disclosure**, конфиденциальности;
- **denial of service**, доступности;
- **elevation of privilege**, авторизованности.



Модель актуальных угроз ИС:
диаграммы потоков данных (DFD)
Элементы верхнеуровневой DFD для моделируемой системы.

- **процесс**, — компонент ИС, осуществляющий обработку или передачу информации;
- **интерактор**, — внешний, по отношению к ИС компонент, осуществляющий информационный обмен с каким-либо компонентом ИС;
- **хранилище**, — компонент ИС, осуществляющий хранение или передачу информации;
- **поток данных**, — канал обмена информацией между процессами, интеракторами и хранилищами;
- **граница доверия**, — проходит через потоки данных и отделяет доверенные компоненты ИС от недоверенных.

4. Угрозы информационной безопасности. DFD диаграммы для моделирования угроз ИБ.

На основании *актуальности угроз* для каждого из типов компонентов оценивается:

- угрозы для процессов;
- S и R для интеракторов;
- T,R,I,D для хранилищ;
- T,I,D для потоков данных.

- На основании информации о *пересечении потоками данных границ доверия* строится список актуальных угроз для модели.
- Далее производится декомпозиция каждого из элементов диаграммы так, как если бы этот компонент был отдельной ИС, причем потоки данных, циркулирующие внутри него, также размечаются границами доверия, если это необходимо.
- По полученной DFD вновь строится список актуальных угроз, дополняющий предыдущий. Этот процесс повторяется рекурсивно до тех пор, пока достигнутая степень детализации модели не перестанет вносить новые актуальные угрозы, либо пока возможна декомпозиция.

Инструмент: Microsoft Threat Modelling Tool

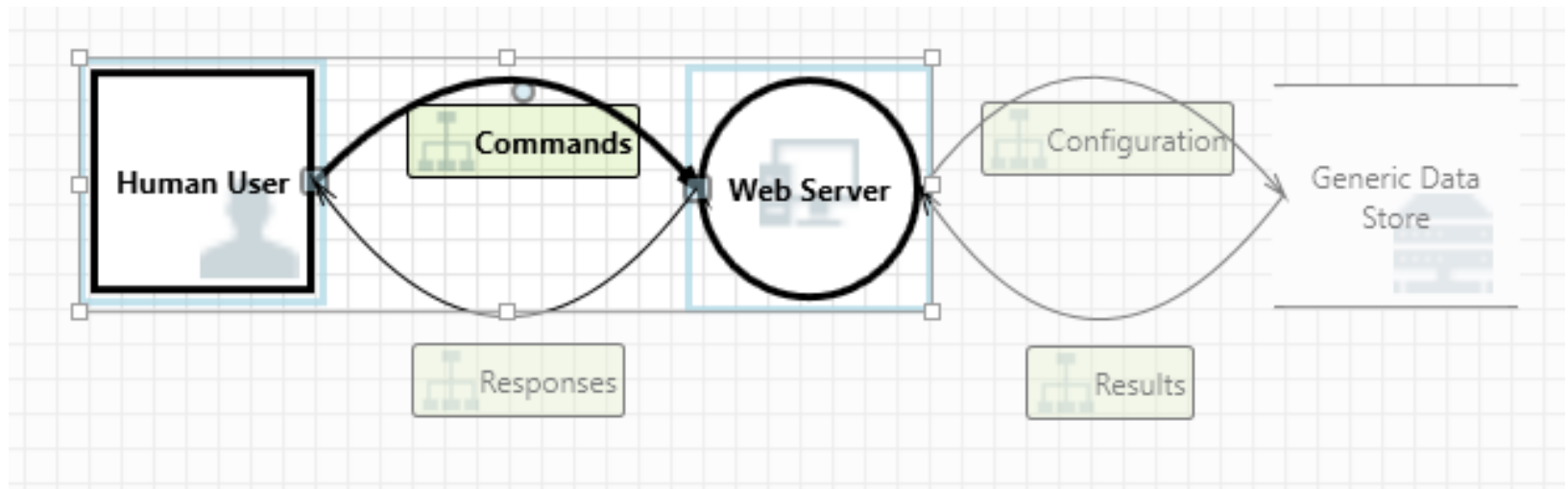
<https://learn.microsoft.com/ru-ru/azure/security/develop/threat-modeling-tool>

4. Угрозы информационной безопасности. DFD диаграммы для моделирования угроз ИБ.

В окне свойств угроз появляются дополнительные сведения об угрозах.

Созданная угроза помогает ему понять потенциальные изъяны проектирования.

Классификация STRIDE дает представление о потенциальных векторах атаки, тогда как в дополнительном описании содержатся сведения о том, что именно идет не так, и возможные способы решения проблемы.



Threat Properties

ID:	0	Diagram:	Diagram 1	Status:	Not Started
Title:	Spoofing the Human User External Entity				
Category:	Spoofing				
Description:	Human User may be spoofed by an attacker and this may lead to unauthorized access to Web Server. Consider using a standard authentication mechanism to identify the external entity.				
Justification:					
Interaction:	Commands				
Priority:	High				

5. Стандарт OWASP (Open Web Application Security Project). Рейтинг OWASP Top 10 (2021) десяти наиболее опасных рисков информационной безопасности для веб-приложений.



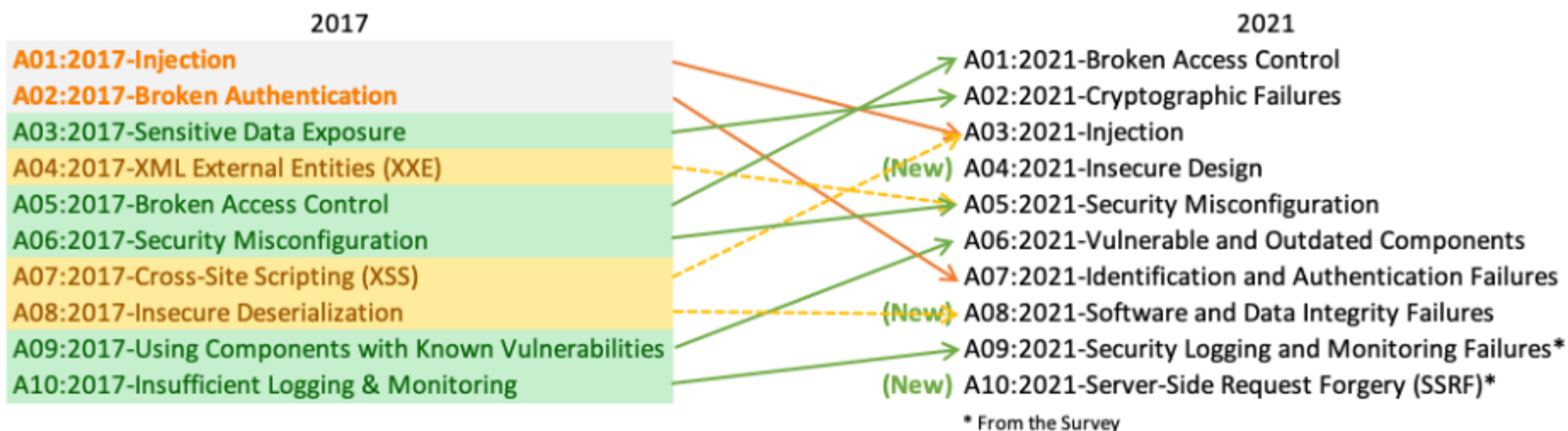
Open Web Application Security Project (OWASP) – это некоммерческая организация, а также открытое интернет-сообщество, целью которого является защита организаций посредством разработки безопасного кода, тестирования на проникновение и сопровождения разрабатываемых приложений на всех этапах проекта.

OWASP использует различные ресурсы (люди, технологии, процессы), чтобы решить существующие и возникающие проблемы в разработке безопасных приложений.

Это происходит с помощью внедрения библиотек и применения инструментов безопасности, предоставляемых OWASP.

<https://owasp.org/Top10/>

5. Стандарт OWASP (Open Web Application Security Project). Рейтинг OWASP Top 10 (2021) десяти наиболее опасных рисков информационной безопасности для веб-приложений.



Open Web Application Security Project 2021 представляет собой нумерованный список угроз:

- A1: Неверное управление доступом
- A2: Криптографические ошибки и сбои
- A3: Инъекция
- A4: Небезопасный дизайн
- A5: Неверная конфигурация безопасности
- A6: Использование уязвимых и устаревших компонент
- A7: Ошибки идентификации и аутентификации
- A8: Ошибки целостности программного обеспечения и данных
- A9: Ошибки логирования и мониторинга безопасности
- A10: Подделка запросов на стороне сервера

5. Стандарт OWASP (Open Web Application Security Project).

A1: Неверное управление доступом (Broken Access Control). Пример.

Управление доступом применяет политику таким образом, что пользователи не могут действовать за пределами своих предполагаемых разрешений. Сбои обычно приводят к несанкционированному раскрытию информации, модификации или уничтожению всех данных или к выполнению бизнес-функции за пределами полномочий пользователя.

Нарушения предполагают:

- Обход проверок управления доступом путем изменения URL-адреса (изменение параметров или принудительный просмотр), внутреннего состояния приложения или HTML-страницы либо с помощью инструмента атаки, изменяющего запросы API.
- Разрешение просмотра или редактирования чужой учетной записи путем предоставления ее уникального идентификатора (небезопасные прямые ссылки на объекты).
- Доступ к API с отсутствующими элементами управления доступом для POST, PUT и DELETE.
- Манипуляции с метаданными, такие как воспроизведение или подделка маркера управления доступом JSON Web Token (JWT), а также использование файла cookie или скрытого поля для повышения привилегий или злоупотребления аннулированием JWT.
- Неправильная конфигурация CORS разрешает доступ к API из неавторизованных/ненадежных источников.
- Принудительный просмотр аутентифицированных страниц в качестве неаутентифицированного пользователя или привилегированных страниц в качестве обычного пользователя.

5. Стандарт OWASP (Open Web Application Security Project).

A1: Неверное управление доступом (Broken Access Control). Пример.

Примеры сценариев атаки

Сценарий № 1. Приложение использует непроверенные данные в вызове SQL, который обращается к информации об учетной записи:

```
pstmt.setString(1, request.getParameter("acct"));
```

```
ResultSet results = pstmt.executeQuery( );
```

Злоумышленник просто изменяет параметр «acct» браузера, чтобы отправить любой номер учетной записи, который он хочет. В случае неправильной проверки злоумышленник может получить доступ к учетной записи любого пользователя.

```
https://example.com/app/accountInfo?acct=notmyacct
```

Сценарий № 2. Злоумышленник просто заставляет просматривать целевые URL-адреса. Для доступа к странице администратора требуются права администратора.

```
https://example.com/app/getappInfo
```

```
https://example.com/app/admin_getappInfo
```

Если неавторизованный пользователь может получить доступ к любой странице, это недостаток.

6. Протокол автоматизации управления данными безопасности (SCAP).

Набор открытых стандартов, определяющих технические спецификации для представления и обмена данными по безопасности.

SCAP (Security Content Automation Protocol) — спецификация, которая определяет три процесса: поиск и исправление уязвимостей, автоматическую настройку конфигураций, а также оценку уровня безопасности.

Например, SCAP состоит из следующих стандартов:

- Типовые уязвимости и ошибки конфигурации (Common Vulnerabilities and Exposures CVE(r))
- Список типовых конфигураций (Common Configuration Enumeration CCE™)
- Список типовых платформ (Common Platform Enumeration CPE)
- Единая система определения величины риска уязвимостей (Common Vulnerability Scoring System CVSS)
- Расширяемый формат описания списка проверки конфигурации (Extensible Configuration Checklist Description Format XCCDF) <https://scap.nist.gov/specifications/xccdf/>
- Открытый язык описания уязвимостей и оценки (Open Vulnerability and Assessment Language OVAL™)
- Перечень общеизвестных слабых мест (Common Weakness Enumeration, CWE);
- Система оценки общеизвестных слабых мест (Common Weakness Scoring System, CWSS);

6. Протокол автоматизации управления данными безопасности (SCAP).

Протокол автоматизации управления данными безопасности (SCAP) представляет собой набор открытых стандартов, определяющих технические спецификации для представления и обмена данными по безопасности. Эти данные могут быть использованы для автоматизации процесса поиска уязвимостей, оценки соответствия технических механизмов контроля и измерения уровня защищенности.

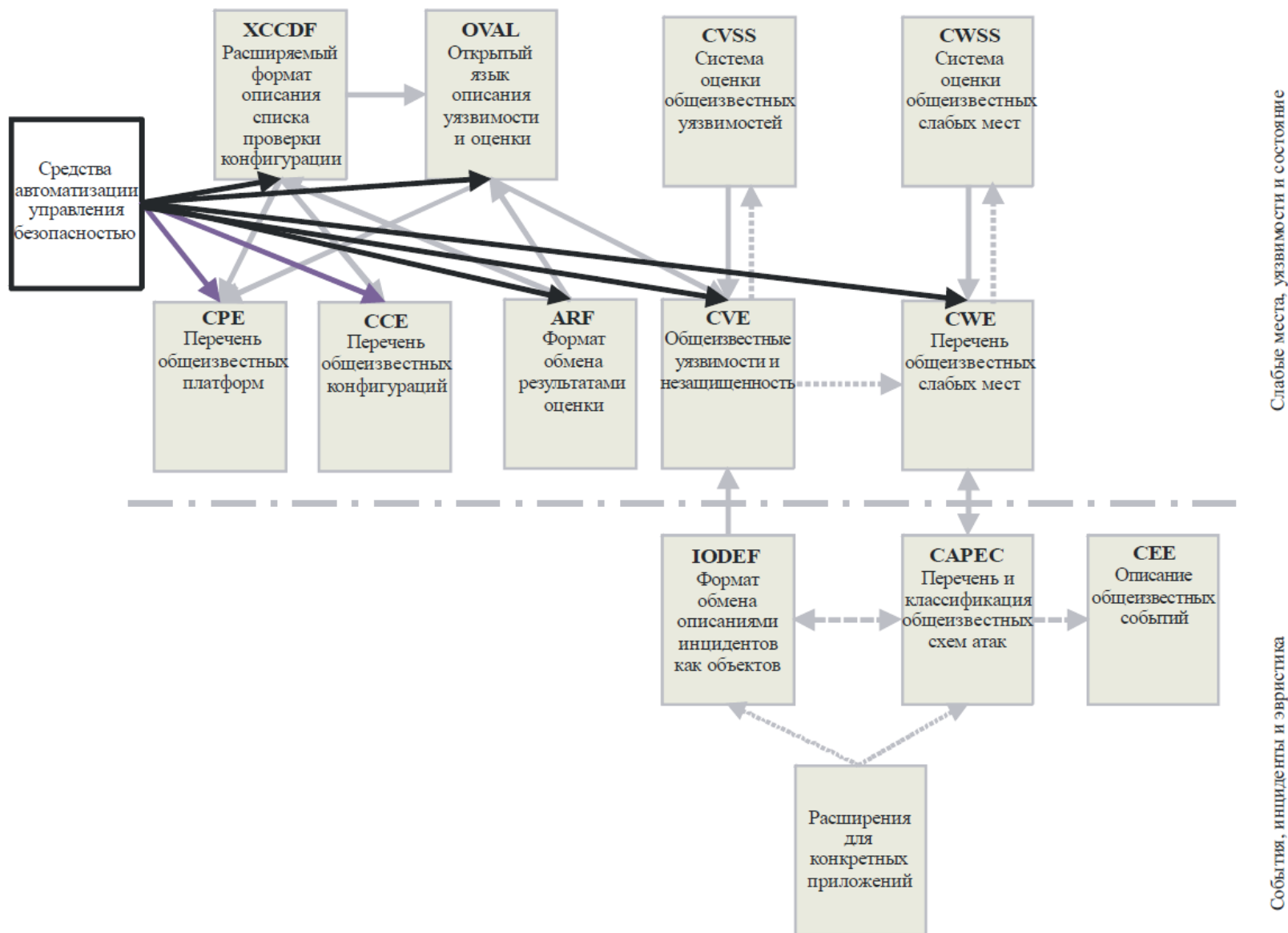
SCAP (Security Content Automation Protocol) включает в себя ряд открытых стандартов, поддерживаемых международным сообществом профессионалов в области информационной безопасности. Последняя версия SCAP состоит из одиннадцати компонентов протокола в пяти категориях.

<https://csrc.nist.gov/projects/security-content-automation-protocol/>

Компоненты SCAP (The Security Content Automation Protocol).

- 1. Языки.** Языки SCAP стандартизуют словари и выражения, описывающие политику безопасности, механизмы контроля и результаты оценки. SCAP включает в себя следующие компоненты:
Расширяемый формат описания контрольного списка конфигураций (XCCDF, Extensible configuration checklist description format);
Открытый язык описания уязвимостей и проведения оценок (OVAL, Open vulnerability and assessment language);
Открытый интерактивный язык описания контрольного списка (OCIL, Open checklist interactive language).
- 2. Формат отчетов.** Форматы отчета SCAP представляют необходимые конструкции, для выражения собранной информации в стандартизированных форматах:
Формат представления данных об активах ИБ (Asset Reporting Format);
Формат для уникальной идентификации активов ИБ на основе идентификаторов (Asset Identification [AI]).
- 3. Перечни.** Перечни SCAP определяют стандартизованные спецификации, официальные перечни (словари), выраженные с использованием этих спецификаций. SCAP включает в себя следующие перечни:
Общий перечень платформ (CPE , Common platform enumeration);
Общий перечень конфигураций (CCE , Common configuration enumeration);
Общий перечень уязвимостей и рисков (CVE, Common vulnerabilities and exposures).
- 4. Измерение и оценка систем.** В SCAP это выражается в оценке определенных особенностей уязвимости (например, слабых мест программного обеспечения и проблем конфигурации безопасности) и определении количественного значения влияния уязвимости (метрики). Метрики SCAP в терминах системных технических требований описываются:
Общей системой оценки уязвимости (CVSS, Common vulnerability scoring system) и
Общей системой оценки конфигурации (CCSS, Common configuration scoring system).
- 5. Целостность.** Спецификация целостности SCAP предназначена для обеспечения целостности информационного SCAP-контента и полученных с помощью него результатов. Модель доверия для данных об автоматизации безопасности (TMSAD, Trust Model for Security Automation Data) является спецификацией целостности SCAP.

6. Протокол автоматизации управления данными безопасности (SCAP).



7. Управление рисками информационной безопасности. Составляющие процесса управления рисками: процедуры своевременного выявления рисков (risk identification), их оценка (risk assessment) и последующая обработка (risk treatment).

Управление рисками информационной безопасности по своей сути является ядром системы менеджмента информационной безопасности (СМИБ) [ГОСТ Р ИСО/МЭК 27001 Информационные технологии - Методы обеспечения безопасности - Системы менеджмента информационной безопасности - Требования].

Составляющими процесса управления рисками являются процедуры своевременного выявления рисков (risk identification), их оценка (risk assessment) и последующая обработка (risk treatment).

Методология оценки рисков информационной безопасности предусматривает такие шаги, как:

- выявление уязвимостей (организационных и технических);
- выявление угроз, направленных на рассматриваемые активы;
- определение последствий от реализации угроз;
- выявление существующих контролей (контрмер);
- определение вероятности реализации угроз.

7. Управление рисками информационной безопасности. Формулы для расчета риска.

В качестве мер IT-риска выступают threat (угроза), vulnerability (уязвимость) и ценность актива:

$$Risk = Threat * Vulnerability * Asset$$

Другой вариант (с учетом контмер):

$$Risk = ((Vulnerability * Threat) / CounterMeasure) * Asset_VaR$$

Показатель *Value at Risk (VaR)* – стоимостная мера риска, выраженная в денежных единицах оценка величины, которую не превысят ожидаемые в течение данного периода времени потери с заданной вероятностью.

* VaR — это величина убытков, которая с вероятностью, равной уровню доверия (например, 99 %), не будет превышена. Следовательно, в 1 % случаев убыток составит величину, большую чем VaR.

<http://it-risk-management.com/>

ISO/IEC TR 27016:2014 Information security – Security techniques – Information security management – Organizational economics

<https://www.isaca.org/journal/archives/2010/volume-1/pages/performing-a-security-risk-assessment1.aspx>

7. Управление рисками информационной безопасности. Формулы для расчета риска. Показатели ALE, ARO, SLE.

Затраты на защиту информационного актива и прочие экономические факторы (объекты или информация) влияют на стоимость актива для организации и мер по его защите. Убыток, реализуемый посредством реализации риска и его негативных последствий, в стандарте расценивается как снижение стоимости актива.

VaR характеризует наибольший убыток за определённый период времени, который не будет превышен с ожидаемой вероятностью» *[ISO/IEC TR 27016:2014 Information security – Security techniques – Information security management – Organizational economics]*.

В свою очередь, размер убытков характеризуется показателем **ALE** (ожидаемый среднегодовой убыток).

$$\mathbf{ALE = ARO * SLE,}$$

где

ALE (annualized loss expectancy) - ожидаемые потери в год,

ARO (annual rate of occurrence) – частота возникновения инцидента течение года;

SLE (single loss expectancy) – размер потерь в случае одного инцидента.

8. Управление рисками информационной безопасности. Идентификация и оценка технических уязвимостей. Расчет рисков по методике CVSS 3.0 Common Vulnerability Score System. Общая система оценки уязвимостей.

Расчет рисков по методике CVSS 3.0 Common Vulnerability Score System. Общая система оценки уязвимостей.

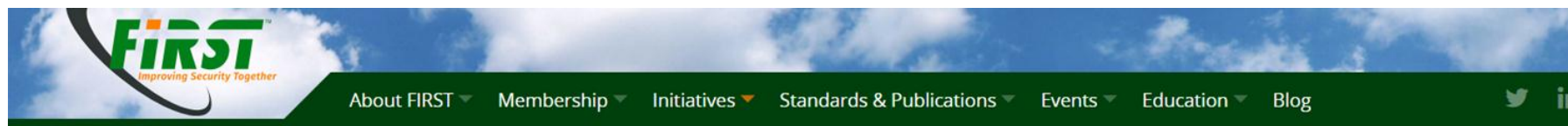
Метрики CVSS. Основные пользователи системы. Описание базовых, контекстных, временных метрик. Комментарии, примеры. Формулы расчета риска. Калькулятор. Примеры уязвимостей.

Документ МСЭ Т-РЕС-Х.1521-3.0 (03/2016) СЕРИЯ Х: СЕТИ ПЕРЕДАЧИ ДАННЫХ, ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ И БЕЗОПАСНОСТЬ. Обмен информацией, касающейся кибербезопасности – Обмен информацией об уязвимости/состоянии. Система оценки общеизвестных уязвимостей 3.0.

<https://www.first.org/cvss/calculator/3.0>

<https://nvd.nist.gov/vuln/detail/CVE-2022-22284>

8. Управление рисками информационной безопасности. Идентификация и оценка технических уязвимостей. Расчет рисков по методике CVSS 3.0 Common Vulnerability Score System. Общая система оценки уязвимостей.



Common Vulnerability Scoring System (CVSS-SIG)

■ Calculator

- Specification Document
- User Guide
- Examples
- CVSS v3.1 Documentation & Resources
- CVSS v3.0 Archive
- CVSS v2 Archive
- CVSS v1 Archive
- JSON & XML Data Representations
- CVSS On-Line Training Course
- Identity & logo usage



Common Vulnerability Scoring System Version 3.1 Calculator

Hover over metric group names, metric names and metric values for a summary of the information in the official CVSS v3.1 Specification Document available in the list of links on the left, along with a User Guide providing additional scoring guidance, an Examples document of scored vulnerability calculator (including its design and an XML representation for CVSS v3.1).

Base Score

Attack Vector (AV)

Network (N) Adjacent (A) Local (L) Physical (P)

Attack Complexity (AC)

Low (L) High (H)

Privileges Required (PR)

None (N) Low (L) High (H)

User Interaction (UI)

None (N) Required (R)

Scope (S)

Unchanged (U) Changed (C)

Confidentiality (C)

None (N) Low (L) High (H)

Integrity (I)

None (N) Low (L) High (H)

Availability (A)

None (N) Low (L) High (H)

8. Управление рисками информационной безопасности. Идентификация и оценка технических уязвимостей. Расчет рисков по методике CVSS 3.0 Common Vulnerability Score System. Общая система оценки уязвимостей.

Общая система оценки уязвимостей [*Common Vulnerability Scoring System, CVSS*], версия 2.0 (3.1-текущая) – открытая схема для обмена и оценки уязвимостей ИТ. В этой системе используются группы метрик, а также дается описание базовых метрик [*base metrics*], вектора уязвимости [*vector*] и оценок уязвимости.

Группы метрик CVSS 3.0:

- I. **Базовые метрики**: используются для описания основополагающих сведений об уязвимости — возможности эксплуатации уязвимости и воздействии уязвимости на систему, не изменяются со временем и не зависят от среды.
- II. **Временные метрики** [*temporal*]: при оценке метрики учитывается время, например, опасность уязвимости [*severity of the vulnerability*] снижается с выходом официального обновления безопасности [*official patch*].
- III. **Контекстные метрики** [*environmental*]: вопросы контекста, среды принимаются во внимание при оценке опасности уязвимости. Например, чем больше систем подвержены [*affected*] уязвимости, тем выше ее опасность.

Подробнее на русском языке: Документ МСЭ Т-РЕС-Х.1521-3.0 (03/2016) СЕРИЯ Х: СЕТИ ПЕРЕДАЧИ ДАННЫХ, ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ И БЕЗОПАСНОСТЬ. Обмен информацией, касающейся кибербезопасности – Обмен информацией об уязвимости/состоянии. Система оценки общеизвестных уязвимостей 3.0.

8. Управление рисками информационной безопасности. Идентификация и оценка технических уязвимостей. Расчет рисков по методике CVSS 3.0 Common Vulnerability Score System. Общая система оценки уязвимостей.

Компоненты системы, для которых рассчитываются метрики:

уязвимый компонент (vulnerable component) — тот компонент информационной системы, который содержит уязвимость и подвержен эксплуатации;

атакуемый компонент (impacted component) — тот, конфиденциальность, целостность и доступность которого могут пострадать при успешной реализации атаки.

В большинстве случаев **уязвимый** и **атакуемый** компоненты совпадают, но есть целые классы уязвимостей, для которых это не так, например:

- выход за пределы песочницы приложения;
- получение доступа к пользовательским данным, сохраненным в браузере, через уязвимость в веб-приложении (XSS);
- выход за пределы гостевой виртуальной машины.

CVSS v 3: **Метрики эксплуатируемости** для **уязвимого компонента**
Метрики воздействия для **уязвимого** и **атакуемого компонента**

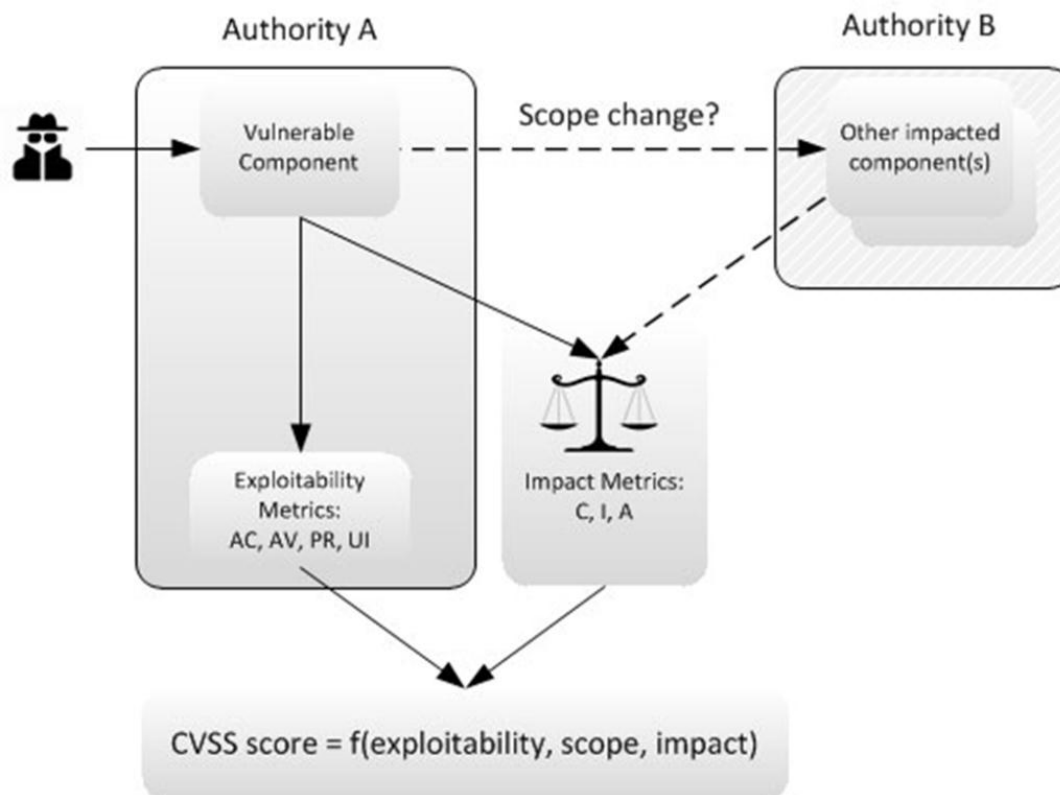
Иллюстрация на следующем слайде.

8. Управление рисками информационной безопасности. Идентификация и оценка технических уязвимостей. Расчет рисков по методике CVSS 3.0 Common Vulnerability Score System. Общая система оценки уязвимостей.

Компоненты системы, для которых рассчитываются метрики:

уязвимый компонент (vulnerable component) — тот компонент информационной системы, который содержит уязвимость и подвержен эксплуатации;

атакуемый компонент (impacted component) — тот, конфиденциальность, целостность и доступность которого могут пострадать при успешной реализации атаки.



8. Управление рисками информационной безопасности. Идентификация и оценка технических уязвимостей. Расчет рисков по методике CVSS 3.0 Common Vulnerability Score System. Практический пример.

Для каждого вопроса по системе оценки уязвимостей CVSS студенту предлагается проанализировать свой вариант уязвимости.

Приведено описание уязвимости, оценка и базовый вектор в формате CVSS 3.0

Задание: расшифруйте вектор, кратко поясните значения метрик.

Ответьте на вопрос: какими метриками из группы временных и контекстных вы можете дополнить этот вектор?

CVE-2022-22284 - Improper authentication vulnerability in Samsung Internet prior to 16.0.2.19 allows attackers to bypass secret mode password authentication.

Base Score: 5.5 MEDIUM

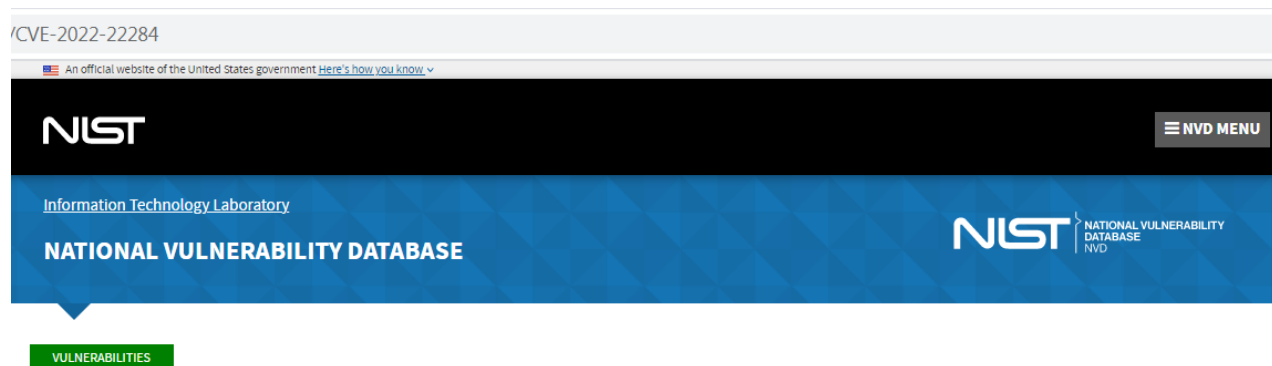
*Vector: **CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N***

(Рассматриваем только последнюю версию калькулятора CVSS 3.x)

Для того, чтобы выполнить анализ представленной уязвимости необходимо обратиться к базе данных уязвимостей NVD:

<https://nvd.nist.gov/vuln/detail/CVE-2022-22284>

8. Управление рисками информационной безопасности. Идентификация и оценка технических уязвимостей. Расчет рисков по методике CVSS 3.0 Common Vulnerability Score System. Практический пример.



Base Score: 5.5 MEDIUM
Vector:
CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N
(Рассматриваем только последнюю версию калькулятора CVSS 3.x)

🚩 CVE-2022-22284 Detail

Description

Improper authentication vulnerability in Samsung Internet prior to 16.0.2.19 allows attackers to bypass secret mode password authentication

Severity CVSS Version 3.x CVSS Version 2.0

CVSS 3.x Severity and Metrics:

Source	Base Score	Vector
NIST: NVD	5.5 MEDIUM	CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:N/A:N
CNA: Samsung Mobile	5.7 MEDIUM	CVSS:3.1/AV:L/AC:H/PR:N/UI:N/S:U/C:L/I:H/A:N

NVD Analysts use publicly available information to associate vector strings and CVSS scores. We also display any CVSS information provided within the CVE List from the CNA.

Note: It is possible that the NVD CVSS may not match that of the CNA. The most common reason for this is that publicly available information does not provide sufficient detail or that information simply was not available at the time the CVSS vector string was assigned.

QUICK INFO

CVE Dictionary Entry:

CVE-2022-22284

NVD Published Date:

01/10/2022

NVD Last Modified:

01/18/2022

Source:

Samsung Mobile

Путем нажатия непосредственно на оценку 5.5 попадаем в калькулятор для оценки метрик.

References to Advisories, Solutions, and Tools

By selecting these links, you will be leaving NIST webspace. We have provided these links to other web sites because they may have information that would be of interest to you. No inferences should be drawn on account of other sites being referenced, or not, from this page. There may be other web sites that are more appropriate for your purpose. NIST does not necessarily endorse the views expressed,

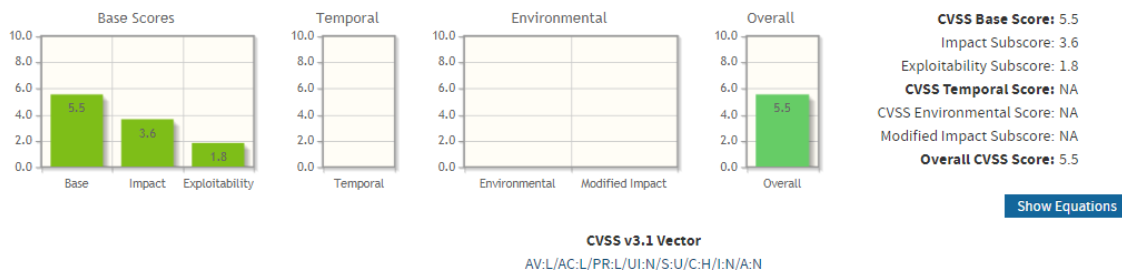
В разделе «References to Advisories, Solutions, and Tools» находится информация о наличии обновлений и рекомендаций производителя или третьих лиц.

8. Управление рисками информационной безопасности. Идентификация и оценка технических уязвимостей. Расчет рисков по методике CVSS 3.0 Common Vulnerability Score System. Практический пример.

Common Vulnerability Scoring System Calculator CVE-2022-22284

Source: NIST

This page shows the components of the CVSS score for example and allows you to refine the CVSS base score. Please read the CVSS standards guide to fully understand how to score CVSS vulnerabilities and to interpret CVSS scores. The scores are computed in sequence such that the Base Score is used to calculate the Temporal Score and the Temporal Score is used to calculate the Environmental Score.



Base Score Metrics

Exploitability Metrics

Attack Vector (AV)*

Network (AV:N) | Adjacent Network (AV:A) | **Local (AV:L)** | Physical (AV:P)

Attack Complexity (AC)*

Low (AC:L) | High (AC:H)

Privileges Required (PR)*

None (PR:N) | **Low (PR:L)** | High (PR:H)

User Interaction (UI)*

None (UI:N) | Required (UI:R)

Scope (S)*

Unchanged (S:U) | Changed (S:C)

Impact Metrics

Confidentiality Impact (C)*

None (C:N) | Low (C:L) | **High (C:H)**

Integrity Impact (I)*

None (I:N) | Low (I:L) | High (I:H)

Availability Impact (A)*

None (A:N) | Low (A:L) | High (A:H)

* - All base metrics are required to generate a base score.

Temporal Score Metrics

Exploit Code Maturity (E)

Not Defined (E:X) | Unproven that exploit exists (E:U) | Proof of concept code (E:P) | Functional exploit exists (E:F) | High (E:H)

Remediation Level (RL)

Not Defined (RL:X) | Official fix (RL:O) | Temporary fix (RL:T) | Workaround (RL:W) | Unavailable (RL:U)

Report Confidence (RC)

Базовые метрики

формируют вектор:

CVSS:3.1/AV:L/AC:L/PR:L/
UI:N/S:U/C:H/I:N/A:N

Студенту их необходимо прокомментировать.

Временные метрики

студенту предлагается заполнить

самостоятельно исходя из информации в разделе «References to Advisories, Solutions, and Tools», тем самым уточнить общую итоговую оценку уязвимости. Их можно дополнить исходя из информации о наличии обновлений и рекомендаций производителя или третьих лиц.

8. Управление рисками информационной безопасности. Идентификация и оценка технических уязвимостей. Расчет рисков по методике CVSS 3.0 Common Vulnerability Score System. Список уязвимостей для билетов.

CVE-2022-0129 - Uncontrolled search path element vulnerability in McAfee TechCheck prior to 4.0.0.2 allows a local administrator to load their own Dynamic Link Library (DLL) gaining elevation of privileges to system user. This was achieved through placing the malicious DLL in the same directory that the process was run from.

CVE-2022-21833 - Virtual Machine IDE Drive Elevation of Privilege Vulnerability.

CVE-2022-21839 - Windows Event Tracing Discretionary Access Control List Denial of Service Vulnerability.

CVE-2022-21847 - Windows Hyper-V Denial of Service Vulnerability.

CVE-2022-21855 - Microsoft Exchange Server Remote Code Execution Vulnerability.

CVE-2022-21862 - Windows Application Model Core API Elevation of Privilege Vulnerability.

CVE-2022-21868 - Windows Devices Human Interface Elevation of Privilege Vulnerability.

CVE-2022-21880 - Windows GDI+ Information Disclosure Vulnerability.

CVE-2022-21893 - Remote Desktop Protocol Remote Code Execution Vulnerability.

CVE-2022-21899 - Windows Extensible Firmware Interface Security Feature Bypass Vulnerability.

CVE-2022-21925 - Windows BackupKey Remote Protocol Security Feature Bypass Vulnerability.

CVE-2022-21907 - HTTP Protocol Stack Remote Code Execution Vulnerability.

CVE-2022-0087 - Keystone is vulnerable to Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting').

CVE-2022-22990 - A limited authentication bypass vulnerability was discovered that could allow an attacker to achieve remote code execution and escalate privileges on the My Cloud devices. Addressed this vulnerability by changing access token validation logic and rewriting rule logic on PHP scripts.

CVE-2022-0609 - Use after free in Animation in Google Chrome prior to 98.0.4758.102 allowed a remote attacker to potentially exploit heap corruption via a crafted HTML page.

CVE-2022-22664 - An out-of-bounds read was addressed with improved bounds checking. This issue is fixed in Logic Pro 10.7.3, GarageBand 10.4.6, macOS Monterey 12.3. Opening a maliciously crafted file may lead to unexpected application termination or arbitrary code execution.

CVE-2019-3751 - Dell EMC Enterprise Copy Data Management (eCDM) versions 1.0, 1.1, 2.0, 2.1, and 3.0 contain a certificate validation vulnerability. An unauthenticated remote attacker may potentially exploit this vulnerability to carry out a man-in-the-middle attack by supplying a crafted certificate and intercepting the victim's traffic to view or modify a victim's data in transit.

9. Управление рисками информационной безопасности. База Common Weakness Enumeration. Классификация общеизвестных слабых мест CWE.

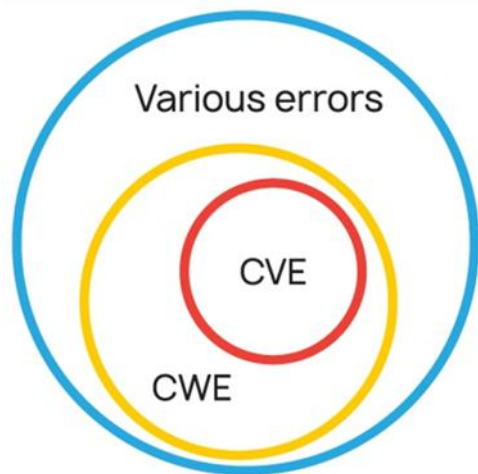
<https://cwe.mitre.org/data/index.html>

Документ МСЭ-Т Х.1524 (03/2012) СЕРИЯ Х: СЕТИ ПЕРЕДАЧИ ДАННЫХ, ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ И БЕЗОПАСНОСТЬ. Обмен информацией, касающейся кибербезопасности – Обмен информацией об уязвимости/состоянии. Перечень общеизвестных слабых мест CWE.

Цель CWE состоит в том, чтобы обеспечить более эффективное обсуждение, описание, отбор и использование инструментальных средств и услуг по защите программного обеспечения, которые могут обнаруживать эти слабые места в кодах источников и операционных системах, а также улучшить понимание слабых мест программного обеспечения, связанных с его архитектурой и проектированием, и управление этими слабыми местами.

Перечень CWE предназначен для того, чтобы охватить причины всех общеизвестных видов уязвимости и незащищенности, связанных со слабыми местами в архитектуре, проектировании, кодировании или развертывании программного обеспечения.

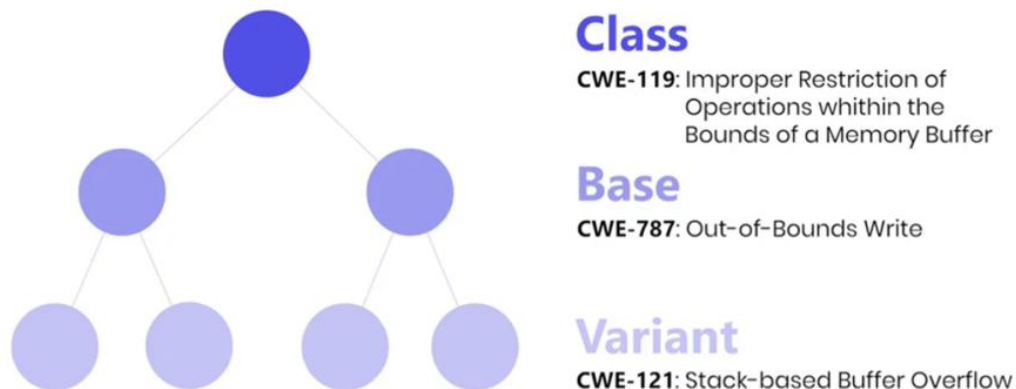
Слабое место (weakness) - дефект или изъян в коде, проектировании, архитектуре или развертывании программного обеспечения, способный в определенный момент стать уязвимостью или приводить к возникновению других уязвимостей.



Ошибки в программном обеспечении, которые могут быть непосредственно использованы злоумышленником для реализации угроз безопасности называются **уязвимостями**. Ошибки, которые могут привести к возникновению уязвимостей – **недостатками безопасности**.

Для классификации недостатков используется многоуровневая структура, которая описывает древовидное устройство CWE: конечные недостатки объединяются в типы, типы – в категории, категории – в представления. Каждое представление – особый способ классификации записей CWE, предназначенный для упрощения решения конкретной задачи.

9. Управление рисками информационной безопасности. База Common Weakness Enumeration. Классификация общеизвестных слабых мест CWE.



Тип (тип) – это абстракция (метка) слабого места. Тип обозначается одним из символов:

V (variant) - Вариант - слабость, которая связана с определенным типом продукта, обычно с использованием определенного языка или технологии. Более конкретная, чем базовая слабость. Слабые стороны уровня варианта обычно описывают проблемы с точки зрения 3-5 следующих параметров: поведение, свойство, технология, язык и ресурс;

C (class) - Класс - это слабость, которая описывается очень абстрактно, как правило, независимо от какого-либо конкретного языка или технологии. Более конкретный, чем Pillar, но более общий, чем База. Слабые стороны уровня класса обычно описывают проблемы в терминах 1 или 2 следующих измерений: поведение, свойство и ресурс;

B (Base) - База - слабость, которая по-прежнему в основном не зависит от ресурса или технологии, но обладает достаточными деталями, чтобы предоставить конкретные методы обнаружения и предотвращения. Слабые стороны базового уровня обычно описывают проблемы в терминах 2 или 3 следующих параметров: поведение, свойство, технология, язык и ресурс;

Chain – Цепочка - сложный элемент, представляющий собой последовательность двух или более отдельных слабых мест, которые могут быть тесно связаны между собой в рамках программного обеспечения. Одна слабость, X, может непосредственно создать условия, необходимые для того, чтобы другая слабость, Y, вошла в уязвимое состояние. Когда это происходит, CWE обращается к X как к "первичному" Y, а Y является "результатирующим" от X. цепочки могут включать более двух слабых мест, и в некоторых случаях они могут иметь древовидную структуру.

Composite – Композит - сложный элемент, состоящий из двух или более различных слабых мест, в котором все слабые места должны присутствовать одновременно, чтобы возникла потенциальная уязвимость. Устранение любого из недостатков устраняет или резко снижает риск. Одна слабость, X, может быть "разбита" на составляющие слабости Y и Z. бывают случаи, когда одна слабость может не быть существенной для композита, но изменяет природу композита, когда он становится уязвимым.

CWE List: структура и классификатор

View by Software Development

View by Hardware Design

View by Research Concepts

external groupings such as a Top-N list, as well as to express

CWE Top 25 (2020)

OWASP Top Ten (2017)

Seven Pernicious Kingdoms

Software Fault Pattern Clusters

SEI CERT Oracle Coding Standard for Java

SEI CERT C Coding Standard

SEI CERT Perl Coding Standard

CISQ Quality Measures (2020)

Architectural Concepts

Introduced During Design

Introduced During Implementation

Quality Weaknesses with Indirect Security Impacts

Software Written in C

Software Written in C++

Software Written in Java

Software Written in PHP

Weaknesses in Mobile Applications

CWE Composites

CWE Named Chains

CWE Cross-Section

CWE Simplified Mapping

CWE Deprecated Entries

CWE Comprehensive View

Weaknesses without Software Fault Patterns

Weakness Base Elements

CWE List: Software Development

ID:699

[Home](#)[About](#)[CWE List](#)[Scoring](#)[Community](#)[News](#)[Search](#)

CWE VIEW: Software Development

View ID: 699

Type: Graph

Status: Draft

Downloads: [Booklet](#) | [CSV](#) | [XML](#)

▼ Objective

This view organizes weaknesses around concepts that are frequently used or encountered in software development. This includes all aspects of the software development lifecycle including both architecture and implementation. Accordingly, this view can align closely with the perspectives of architects, developers, educators, and assessment vendors. It provides a variety of categories that are intended to simplify navigation, browsing, and mapping.

▼ Audience



Stakeholder	Description
Software Developers	Software developers (including architects, designers, coders, and testers) use this view to better understand potential mistakes that can be made in specific areas of their software application. The use of concepts that developers are familiar with makes it easier to navigate this view, and filtering by Modes of Introduction can enable focus on a specific phase of the development lifecycle.
Educators	Educators use this view to teach future developers about the types of mistakes that are commonly made within specific parts of a codebase.

▼ Relationships

The following graph shows the tree-like relationships between weaknesses that exist at different levels of abstraction. At the highest level, categories and pillars exist to group weaknesses. Categories (which are not technically weaknesses) are special CWE entries used to group weaknesses that share a common characteristic. Pillars are weaknesses that are described in the most abstract fashion. Below these top-level entries are weaknesses at varying levels of abstraction. Classes are still very abstract, typically independent of any specific language or technology. Base level weaknesses are used to present a more specific type of weakness. A variant is a weakness that is described at a very low level of detail, typically limited to a specific language or technology. A chain is a set of weaknesses that must be reachable consecutively in order to produce an exploitable vulnerability. While a composite is a set of weaknesses that must all be present simultaneously in order to produce an exploitable vulnerability.

[Show Details:](#) ☐[Expand All](#) | [Collapse All](#) | [Filter View](#)

699 - Software Development

-  API / Function Errors - (1228)
-  Audit / Logging Errors - (1210)

CWE List: Software Development

ID:699

699 - Software Development

- ☐ **C** API / Function Errors - (1228)
 - ☐ **B** Use of Inherently Dangerous Function - (242)
 - ☐ **B** Use of Function with Inconsistent Implementations - (474)
 - ☐ **B** Undefined Behavior for Input to API - (475)
 - ☐ **B** Use of Obsolete Function - (477)
 - ☐ **B** Use of Potentially Dangerous Function - (676)
 - ☐ **B** Use of Low-Level Functionality - (695)
 - ☐ **B** Exposed Dangerous Method or Function - (749)
- ☐ **C** Audit / Logging Errors - (1210)
- ☐ **C** Authentication Errors - (1211)
- ☐ **C** Authorization Errors - (1212)
- ☐ **C** Bad Coding Practices - (1006)
- ☐ **C** Behavioral Problems - (438)
- ☐ **C** Business Logic Errors - (840)
- ☐ **C** Communication Channel Errors - (417)
- ☐ **C** Complexity Issues - (1226)
- ☐ **C** Concurrency Issues - (557)
- ☐ **C** Credentials Management Errors - (255)
- ☐ **C** Cryptographic Issues - (310)
- ☐ **C** Key Management Errors - (320)
- ☐ **C** Data Integrity Issues - (1214)
- ☐ **C** Data Processing Errors - (19)

CWE-611 (Improper Restriction of XML External Entity - Неверное ограничение XML-ссылок на внешние объекты)

CWE-611: Improper Restriction of XML External Entity Reference

Weakness ID: 611

Abstraction: Base
Structure: Simple

View customized information:

Conceptual

Operational

Mapping-Friendly

Complete

Description

The software processes an XML document that can contain XML entities with URIs that resolve to documents outside of the intended sphere of control, causing the product to embed incorrect documents into its output.

Extended Description

XML documents optionally contain a Document Type Definition (DTD), which, among other features, enables the definition of XML entities. It is possible to define an entity by providing a substitution string in the form of a URI. The XML parser can access the contents of this URI and embed these contents back into the XML document for further processing.

By submitting an XML file that defines an external entity with a file:// URI, an attacker can cause the processing application to read the contents of a local file. For example, a URI such as "file:///c:/winnt/win.ini" designates (in Windows) the file C:\Winnt\win.ini, or file:///etc/passwd designates the password file in Unix-based systems. Using URIs with other schemes such as http://, the attacker can force the application to make outgoing requests to servers that the attacker cannot reach directly, which can be used to bypass firewall restrictions or hide the source of attacks such as port scanning.

Once the content of the URI is read, it is fed back into the application that is processing the XML. This application may echo back the data (e.g. in an error message), thereby exposing the file contents.

Alternate Terms

XXE: XXE is an acronym used for the term "XML eXternal Entities"

Relationships

Relevant to the view "Research Concepts" (CWE-1000)

Nature	Type	ID	Name
ChildOf	🟢	610	Externally Controlled Reference to a Resource in Another Sphere
PeerOf	🟢	441	Unintended Proxy or Intermediary (Confused Deputy)

Relevant to the view "Software Development" (CWE-699)

Nature	Type	ID	Name
MemberOf	🔴	19	Data Processing Errors

Relevant to the view "Weaknesses for Simplified Mapping of Published Vulnerabilities" (CWE-1003)

Relevant to the view "Architectural Concepts" (CWE-1008)

Nature	Type	ID	Name
MemberOf	🔴	1015	Limit Access

Modes Of Introduction

Phase	Note
Implementation	REALIZATION: This weakness is caused during implementation of an architectural security tactic.

Applicable Platforms

Languages

XML (Undetermined Prevalence)

CWE VIEW:
Research Concept
(Концепции исследования)

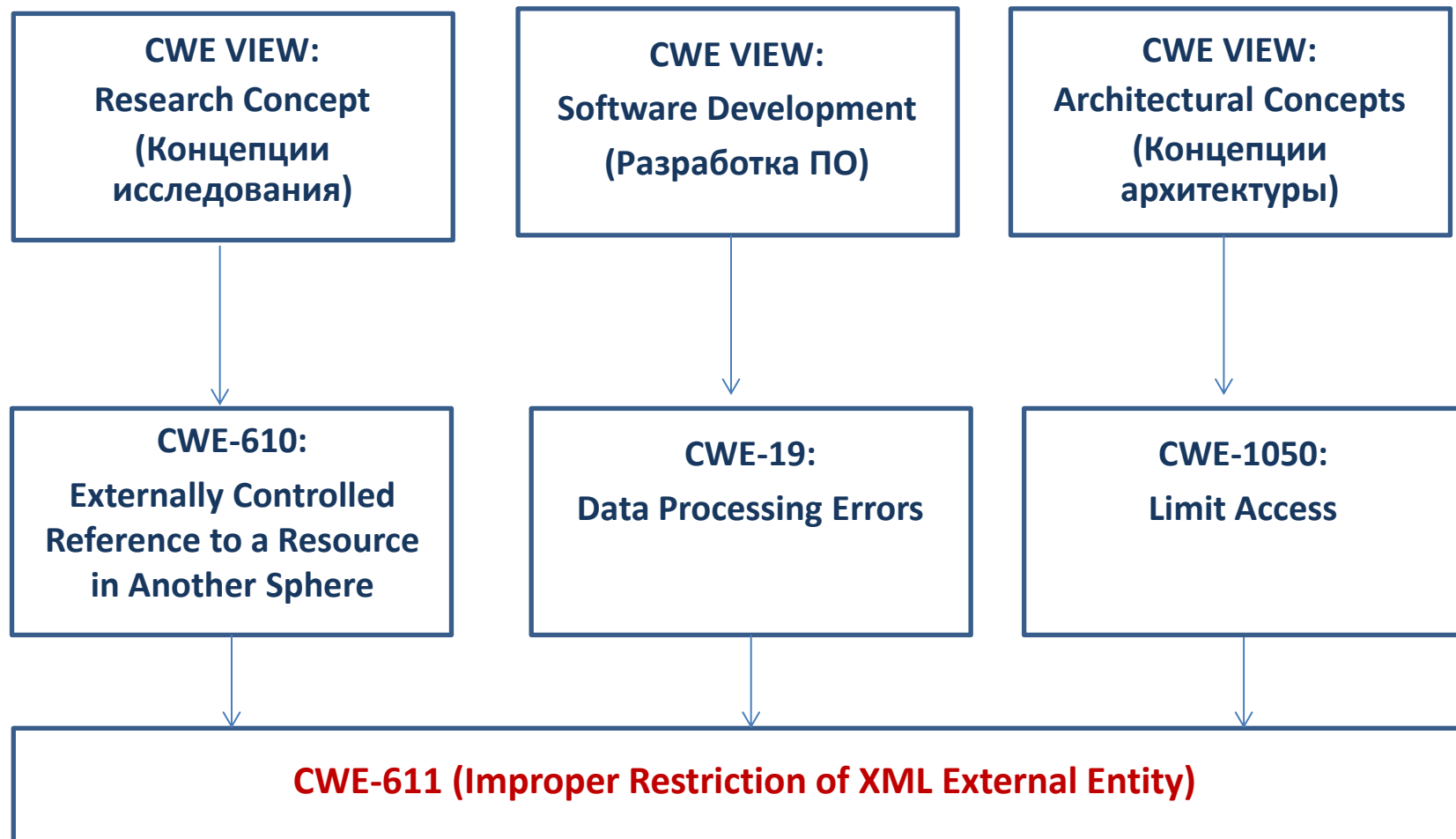
CWE VIEW:
Software Development
(Разработка ПО)

CWE VIEW:
Architectural Concepts
(Концепции архитектуры)

Пример таксономии CWE

Уязвимость – **CVE-2010-2245** (внедрение внешних XML-сущностей в веб-сервисе Apache версии ниже 1.1.1) и недостаток безопасности **CWE-611** (неверное ограничение XML-ссылок на внешние объекты), послуживший причиной этой уязвимости.

<https://cwe.mitre.org/data/definitions/611.html>



9. Управление рисками информационной безопасности. База Common Weakness Enumeration. Практический пример.

Дано слабое место, недостаток безопасности в формате CWE. Определите и постройте для него таксономию. Приведите его описание и один демонстративный пример (если доступен) из базы CWE.

CWE-242: Use of Inherently Dangerous Function.

<https://cwe.mitre.org/data/definitions/242.html>

9. Управление рисками информационной безопасности. База Common Weakness Enumeration. Классификация общеизвестных слабых мест CWE.

Разбор практического примера CWE-242: Use of Inherently Dangerous Function.

Описание слабого места:

Программа вызывает функцию, безопасная работа которой не может быть гарантирована.

Расширенное описание:

Некоторые функции ведут себя опасным образом независимо от того, как они используются. Функции этой категории часто реализовывались без учета соображений безопасности. Функция `gets()` небезопасна, потому что она не выполняет проверку границ размера своих входных данных. Злоумышленник может легко отправить ввод произвольного размера в `gets()` и переполнить буфер назначения.

Классификационные взаимосвязи: (см. пример на предыдущем слайде)

Способы реализации (различные способы реализации или внедрения ошибки предоставляют информацию о том, как и когда может быть введена эта слабость. Этап определяет точку жизненного цикла, в которой может произойти внедрение, а в примечании приводится типичный сценарий, связанный с внедрением на данном этапе).

Применимые платформы (в этом списке показаны возможные области, в которых может проявиться данная слабость. Это могут быть определенные названные языки, операционные системы, архитектуры, парадигмы, технологии или класс таких платформ. Платформа указана вместе с тем, как часто данная слабость появляется для этого экземпляра.): языки программирования C и C++.

Общие последствия: техническое воздействие

Вероятность эксплуатации: высокая

Демонстративные примеры:

Пример 1: В приведенном ниже коде вызывается функция `gets()` для чтения информации в буфер.

Example Language: C

```
char buf[BUFSIZE];  
gets(buf);
```

Пример 2: В приведенном ниже коде вызывается функция `gets()` для считывания данных из командной строки.

Example Language: C

```
char buff[24];  
printf("Please enter your name and press <Enter>\n");  
gets(buff);  
...  
}
```

Способы устранения ошибки:

1) Программист должен знать максимум числа символов, которые должны быть считаны `gets`, чтобы удостовериться, что выделяется буфер достаточного размера. Подобное невозможно без информации о данных. Эта проблема может приводить к созданию ошибок и открывает простор для нарушений компьютерной безопасности при помощи переполнения буфера. Многие источники советуют программистам никогда не использовать `gets` в новых программах. Используйте их безопасный эквивалент.

Например, вместо `gets` могут быть использованы другие функции строкового ввода, что позволит избежать ошибок, связанных с переполнением буфера. Простейшим вариантом будет `fgets(buffer, sizeof(buffer), stdin)`.

2) Используйте инструменты `gcr` или статического анализа, чтобы обнаружить использование опасных функций.

10. Управление рисками информационной безопасности. Common Weakness Scoring System – CWSS - Общая метрическая система оценки «слабых мест».

<https://cwe.mitre.org/data/index.html>

https://cwe.mitre.org/cwss/cwss_v1.0.1.html

Документ МСЭ-Т X.1525 СЕРИЯ X: СЕТИ ПЕРЕДАЧИ ДАННЫХ, ВЗАИМОСВЯЗЬ ОТКРЫТЫХ СИСТЕМ И БЕЗОПАСНОСТЬ. Обмен информацией, касающейся кибербезопасности – Обмен информацией об уязвимости/состоянии. Система оценки общеизвестных слабых мест (CWSS). Открытая структура представления информации о характеристиках и воздействиях слабых мест информационно-коммуникационных технологий (ИКТ) в ходе разработки возможностей программного обеспечения.

Каждая группа содержит несколько метрик- так же обозначаемых как факторы (factors) - которые применяются для вычисления оценки CWSS score (CWSS score, between 0 and 100) для каждого слабого места.

- **Группа показателей базовых:** охватывает внутренние риски, присущие слабому месту, доверие к точности результатов поиска, а также действенность средств контроля.
- **Группа показателей области атаки:** барьеры, которые должен преодолеть злоумышленник, для того чтобы эксплуатировать слабое место.
- **Группа показателей среды:** характеристики слабого места, присущие конкретной среде или операционному контексту.

Подробное описание метрик и примеры есть в документе МСЭ-Т X.1525.



11. CWE/SANS Top 25 Most Dangerous Software Errors. Наиболее Опасные Ошибки ПО. Примеры. Классификация общеизвестных слабых мест CWE.

<https://cwe.mitre.org/top25/index.html>

Rank	ID	Name	Score	KEV Count (CVEs)	Rank Change vs. 2021
1	CWE-787	Out-of-bounds Write	64.20	62	0
2	CWE-79	Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')	45.97	2	0
3	CWE-89	Improper Neutralization of Special Elements used in an SQL Command ('SQL Injection')	22.11	7	+3 ▲
4	CWE-20	Improper Input Validation	20.63	20	0
5	CWE-125	Out-of-bounds Read	17.67	1	-2 ▼
6	CWE-78	Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')	17.53	32	-1 ▼
7	CWE-416	Use After Free	15.50	28	0
8	CWE-22	Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')	14.08	19	0
9	CWE-352	Cross-Site Request Forgery (CSRF)	11.53	1	0
10	CWE-434	Unrestricted Upload of File with Dangerous Type	9.56	6	0
11	CWE-476	NULL Pointer Dereference	7.15	0	+4 ▲
12	CWE-502	Deserialization of Untrusted Data	6.68	7	+1 ▲
13	CWE-190	Integer Overflow or Wraparound	6.53	2	-1 ▼
14	CWE-287	Improper Authentication	6.35	4	0
15	CWE-798	Use of Hard-coded Credentials	5.66	0	+1 ▲
16	CWE-862	Missing Authorization	5.53	1	+2 ▲
17	CWE-77	Improper Neutralization of Special Elements used in a Command ('Command Injection')	5.42	5	+8 ▲
18	CWE-306	Missing Authentication for Critical Function	5.15	6	-7 ▼
19	CWE-119	Improper Restriction of Operations within the Bounds of a Memory Buffer	4.85	6	-2 ▼
20	CWE-276	Incorrect Default Permissions	4.84	0	-1 ▼
21	CWE-918	Server-Side Request Forgery (SSRF)	4.27	8	+3 ▲
22	CWE-362	Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')	3.57	6	+11 ▲
23	CWE-400	Uncontrolled Resource Consumption	3.56	2	+4 ▲
24	CWE-611	Improper Restriction of XML External Entity Reference	3.38	0	-1 ▼
25	CWE-94	Improper Control of Generation of Code ('Code Injection')	3.32	4	+3 ▲

11. CWE/SANS Top 25 Most Dangerous Software Errors. Наиболее Опасные Ошибки ПО. Примеры. Классификация общеизвестных слабых мест CWE.

Алгоритм составления и ранжирования списка CWE Top 25.

Основными источниками информации для исследования являются:

- национальная база данных уязвимостей США (U.D. National Vulnerability Database (NVD)) за 2020–2021 годы;
- каталог эксплуатируемых уязвимостей (KEV) агентства по кибербезопасности и защите инфраструктуры США (Cybersecurity and Infrastructure Security Agency (CISA)), составленный в ноябре 2021 года.

Далее команда исследователей использует на полученных данных собственную формулу для расчёта порядка ранжирования, учитывающую частоту, с которой какой-либо недостаток (CWE) является основной причиной уязвимости, и потенциальную опасность его эксплуатации. Частота и прогнозируемая серьёзность нормализованы относительно своих минимальных и максимальных значений.

Для вычисления частоты упоминания в формуле подсчитывается, сколько раз CVE ссылались на CWE в базе данных NVD. В расчёте используются только те CVE, которые имеют ссылку на CWE, поскольку использование полного набора данных привело бы к очень низким показателям частоты и незначительной разнице между различными типами дефектов.

$$Freq = \{count(CWE_X' \in NVD) \text{ for each } CWE_X' \text{ in } NVD\}$$
$$Fr(CWE_X) = (count(CWE_X \in NVD) - min(Freq)) / (max(Freq) - min(Freq))$$

Другим важным компонентом формулы ранжирования является расчёт серьёзности недостатка, который вычисляется по формуле:

$$Sv(CWE_X) = (average_CVSS_for_CWE_X - min(CVSS)) / (max(CVSS) - min(CVSS))$$

В конце вычисляется итоговая оценка путём перемножения оценки частоты упоминания на оценку серьёзности.

$$Score(CWE_X) = Fr(CWE_X) * Sv(CWE_X) * 100$$

Задание:

Дано слабое место, недостаток безопасности в формате CWE. Определите и постройте для него таксономию. Приведите его описание и один демонстративный пример (если доступен) из базы Top 25 CWE-2022.

CWE-787: Out-of-bounds Write

<https://cwe.mitre.org/data/definitions/787.html>

12. Управление инцидентами информационной безопасности.

SOC: создание ситуационного центра ИБ. Центр оперативного управления информационной безопасностью. Цели и задачи группы реагирования (incident response), управление событиями ИБ.

SIM (Security information management) — управление информационной безопасностью, и SEM (Security event management) — управление событиями безопасности.

SIEM (Security information and event management) системы – задачи и функционал.

13. Управление инцидентами информационной безопасности.

Системы поведенческого анализа - User and Entity Behavioral Analytics (UBA).

User [and Entity] Behavioral Analytics (UBA), как класс систем, позволяющих на основе массивов данных о пользователях и ИТ-сущностях (конечных станциях, серверах, коммутаторах и т. д.) с помощью алгоритмов машинного обучения и статистического анализа строить модели поведения пользователей и определять отклонения от этих моделей, как в режиме реального времени. Цели и задачи таких систем. Основные возможности.

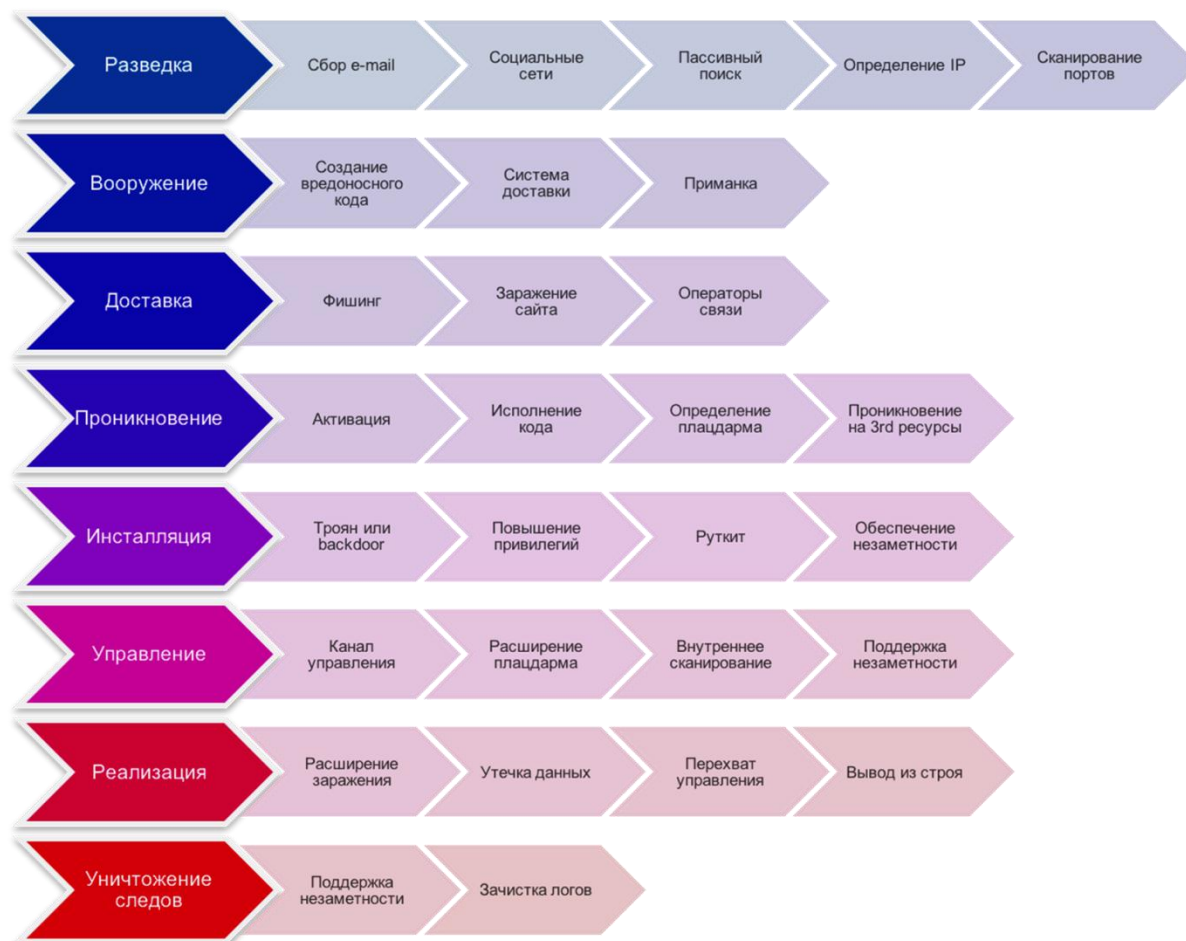
14. Управление инцидентами информационной безопасности.

APT (англ. advanced persistent threat — «развитая устойчивая угроза»; целевая кибератака).

Цели и характерные особенности. Типовой сценарий APT-атаки и этапы.

The Cyber Kill Chain. Киллчейн, как последовательные стадии, которые проходит злоумышленник для успешной реализации атаки. Основные этапы и их характеристики. Примеры.

14. Управление инцидентами информационной безопасности. The Cyber Kill Chain. Киллчейн, как последовательные стадии, которые проходит злоумышленник для успешной реализации атаки. Основные этапы и их характеристики. Примеры.



15. Управление инцидентами информационной безопасности.

IoC (Indicator of Compromise) - Показатели компрометации.

Характерные признаки (observables) и индикаторы компрометации (indicators of compromise, IOC) - формат, описанный в статье MITRE от 2012 года, A structured language for cyber threat intelligence, STIX.

Что представляют собой показатели компрометации IoC, и показатели атаки IoA? Приведите примеры типичных IoC и IoA. Прокомментируйте рисунок ниже.



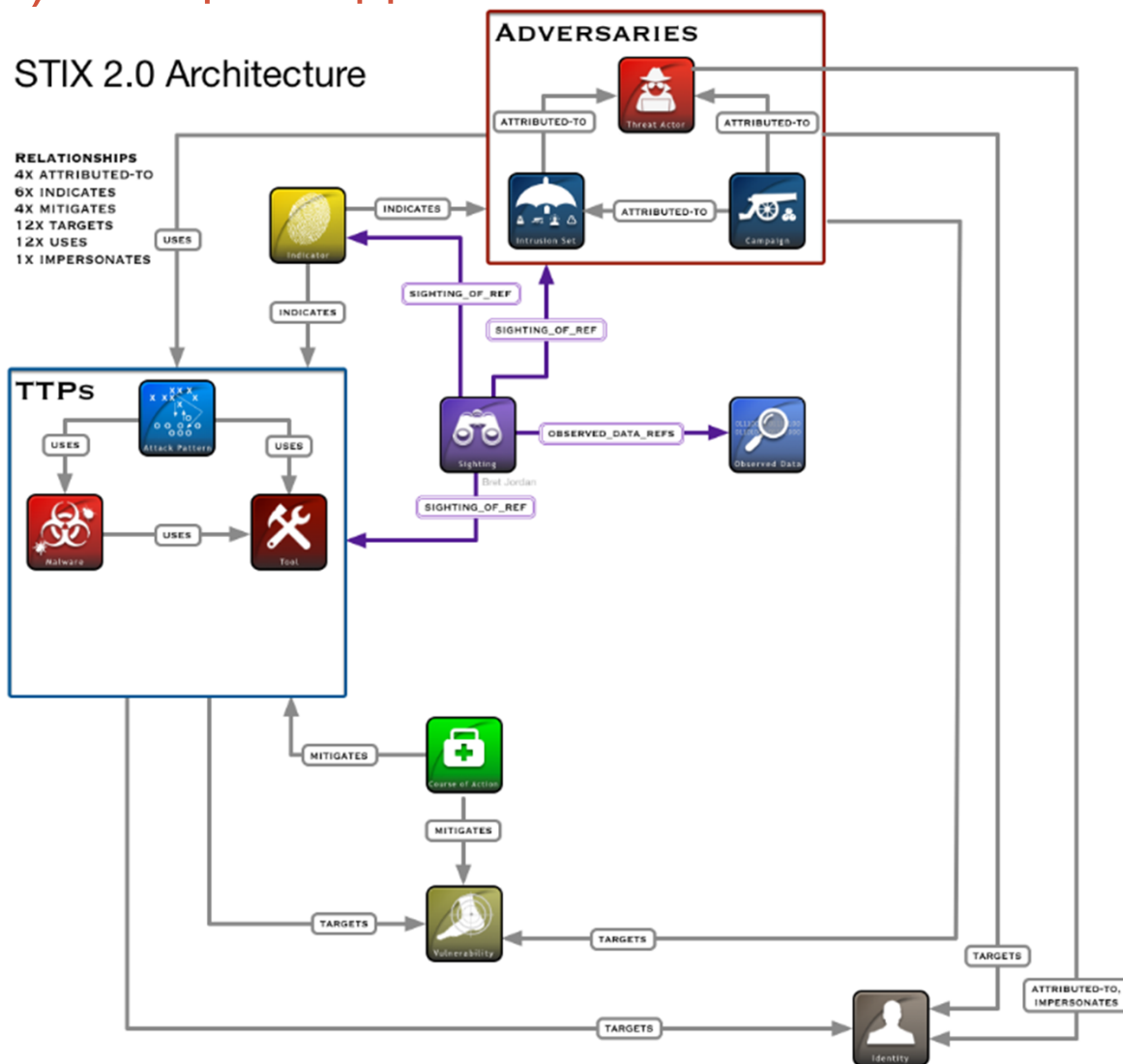
16. Управление инцидентами информационной безопасности.

STIX (Structured Threat Information eXpression).

*Стандарт, используемый для предоставления унифицированной информации о киберугрозах (CTI).
Позволяет совместно использовать описание различных угроз и связанных с ними параметров в различных областях. Формат JSON для описания угроз, визуализация представления в виде графа.
Приведите пример в формате JSON, визуализируйте свой пример.*

Архитектура STIX 2.1.

16. Управление инцидентами информационной безопасности. STIX (Structured Threat Information eXpression). Общий вид.



16. Управление инцидентами информационной безопасности. STIX (Structured Threat Information eXpression). Компоненты.

STIX описывает данные об угрозах как связный граф, где узлами являются **SDO (STIX Domain Objects)**, а ребрами — **SRO (STIX Relationship objects)**.

В качестве **SDO STIX** определяет следующие сущности:

Схема атаки (Attack pattern) — описывает подход (TTP), который использовал злоумышленник для взлома своей цели. Эта сущность используется для классификации атак, обобщения конкретных атак в соответствии со схемами, которым они следуют, и предоставления подробной информации о том, как атаки выполняются.

Вредоносная кампания (Campaign) — описывает последовательность вредоносных поведенческих признаков, которые возникают на протяжении определенных промежутков времени.

План действий (Course of action) — описывает меры, которые нужно принять, чтобы избежать или противостоять атаке.

Личность (Identity) — описывает персоны, организации либо их группы.

Индикатор (Indicator) — описывает технические вредоносные артефакты, которые могут быть использованы для обнаружения вредоносной активности (например, IP-адреса, домены, хеши, ключи реестра).

Intrusion set — описывает набор поведенческих признаков и ресурсов с общими свойствами, которые, вероятней всего, подконтрольны одной организации. Ключевое отличие от Campaign в том, что последняя обычно представляет собой вредоносную активность, направленную на конкретную цель и длится в течение ограниченного промежутка времени, тогда как Intrusion set длится продолжительное время, может участвовать в нескольких Campaigns и иметь несколько целей.

16. Управление инцидентами информационной безопасности. STIX (Structured Threat Information eXpression). Компоненты.

STIX описывает данные об угрозах как связный граф, где узлами являются **SDO (STIX Domain Objects)**, а ребрами — **SRO (STIX Relationship objects)**.

В качестве **SDO STIX** определяет следующие сущности:

Вредоносное ПО (Malware) — описывает экземпляры вредоносного ПО.

Объект наблюдения (Observed data) — описывает не вредоносные технические артефакты.

Отчет (Report) — описывает в понятном виде какую-либо угрозу, вредоносную группировку, их ТТР, жертв. Своего рода аналитическая сводка, позволяющая понять суть угрозы, ее опасность, вредоносное ПО, используемые техники, тактики и процедуры, применяемые атакующей стороной.

Злоумышленник (Threat actor) — описывает персон, группы или организации, которые действуют со злым умыслом. Если коротко, злоумышленники и хакеры. Именно злой умысел в мотивации этой сущности отличает ее от Identity.

Инструмент (Tool) — описывает легитимное ПО, которое может быть использовано для осуществления атак. Отличие этой сущности от Malware именно в том, что это легитимный софт, например, nmap или RDP, VNC.

Уязвимость (Vulnerability) — описывает недостатки/дырки в требованиях, логике, дизайне, реализации ПО или железа, которые могут быть проэксплуатированы и негативно повлиять на конфиденциальность, целостность или доступность системы.

16. Управление инцидентами информационной безопасности. STIX (Structured Threat Information eXpression). JSON (JavaScript Object Notation) синтаксис.

В качестве значений в JSON могут быть использованы:

- **запись** — это неупорядоченное множество пар **ключ: значение**, заключённое в фигурные скобки «{ }». Ключ описывается **строкой**, между ним и значением стоит символ «:». Пары *ключ-значение* отделяются друг от друга запятыми.
- **массив** (одномерный) — это упорядоченное множество **значений**. Массив заключается в квадратные скобки «[]». Значения разделяются запятыми. Массив может быть пустым, то есть не содержать ни одного значения. Значения в пределах одного массива могут иметь разный тип.
- **строка** — это упорядоченное множество из нуля или более символов, заключённое в двойные кавычки.

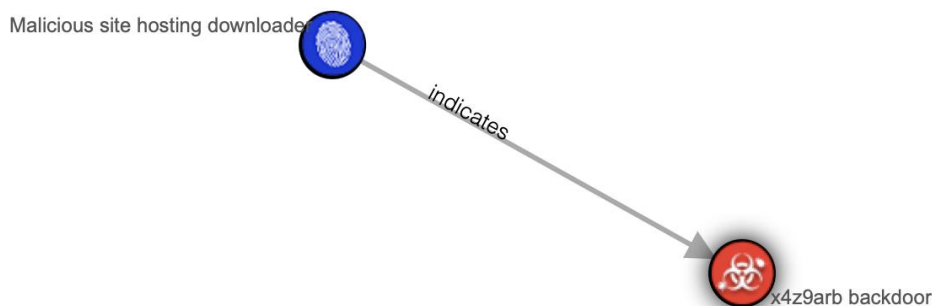
```
{  
  "type": "malware",  
  "is_family": true,  
  "spec_version": "2.1",  
  "id": "malware--591f0cb7-d66f-4e14-a8e6-5927b597f920",  
  "name": "Poison Ivy",  
  "description": "Poison Ivy is a remote access tool, first released in 2005 but unchanged since 2008. It  
includes features common to most Windows-based RATs, including key logging, screen capturing, video  
capturing, file transfers, system administration, password theft, and traffic relaying.",  
  "malware_types": [  
    "remote-access-trojan"  
  ]  
},
```

16. STIX(Structured Threat Information eXpression) - Примеры

Пример 1. Описывается индикатор компрометации <http://x4z9arb.cn/4712/> типа URL и его связь (атрибуция) с вредоносным ПО x4z9arb backdoor.

При этом явно видно, что индикатор — это сайт, на котором располагается вредоносный загрузчик (downloader), в данном случае вредонос x4z9arb backdoor.

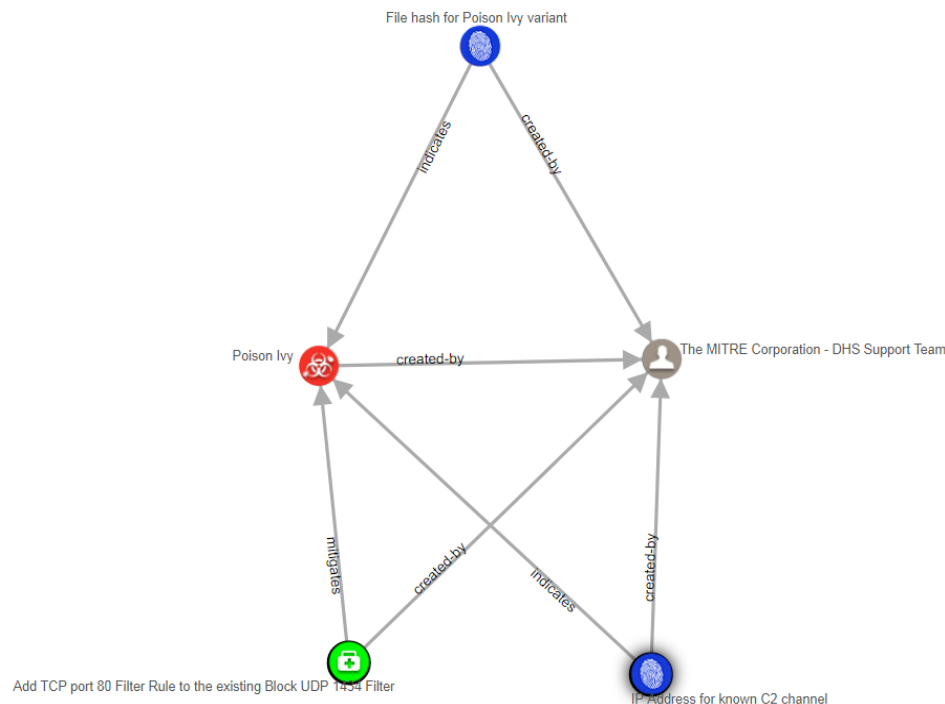
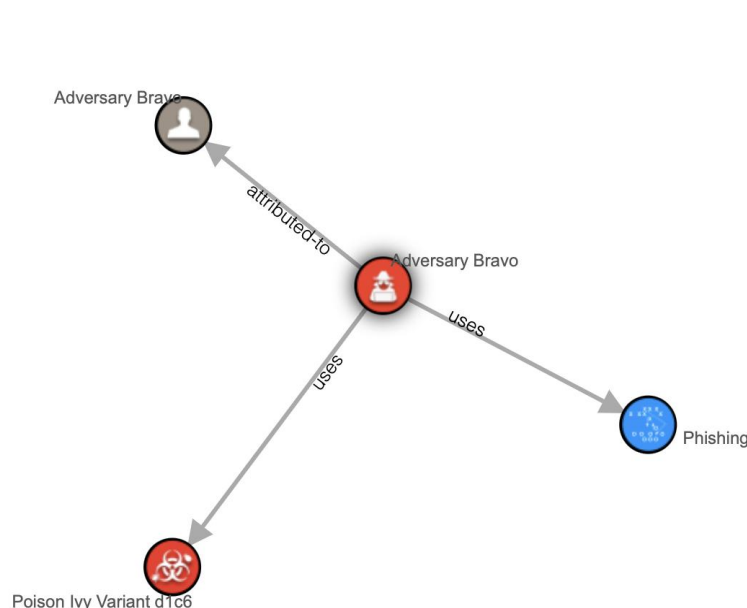
Что это значит для аналитика? Если мы обнаружим следы присутствия (индикатор компрометации <http://x4z9arb.cn/4712/>) в инфраструктуре компании, то можем сделать вывод, что имеем дело с вредоносным ПО x4z9arb backdoor. Дальнейшие шаги обычно зависят от вредоносности ПО, попавшего внутрь инфраструктуры. Для его анализа существует множество баз, например, Malpedia (<https://malpedia.caad.fkie.fraunhofer.de/>)



16. STIX(Structured Threat Information eXpression) - Примеры

Пример 2. Описаны связи между злоумышленником Adversary Bravo, используемой им техникой атаки — фишингом — и вредоносным ПО Poison Ivy Variant d1c6.

В этом случае при обнаружении в инфраструктуре вредоносного ПО PoisonIvy вариант d1c6 такая структура фида с взаимосвязями поможет понять, что подобным вредоносным ПО пользуется злоумышленник.



16. STIX(Structured Threat Information eXpression) - Примеры

Пользуясь доступными ресурсами, студенту предлагается привести свой небольшой пример построения индикатора в формате JSON с его последующей визуализацией.

Необходимые ресурсы:

- <https://oasis-open.github.io/cti-documentation/stix/examples>
- <https://oasis-open.github.io/cti-stix-visualization/>
- <https://oasis-open.github.io/cti-documentation/resources#stix-2.1-specification>
- <https://github.com/oasis-open/cti-stix2-json-schemas/tree/master/examples>

17. Источники поступления информации об угрозах. Threat Intelligence: цели, задачи, этапы. Платформы Threat Intelligence.

Главным элементом платформы Threat Intelligence являются потоки данных, представленные в виде индикаторов компрометации (Indicators of Compromise, IoC).

Они представляют собой признаки, по которым можно обнаружить угрозу безопасности: IP- и URL-адреса, связанные с вредоносной активностью, хеш-суммы вредоносных файлов и т. д.

Последовательность индикаторов компрометации из одного источника принято называть «фидом» (от англ. feed — «подача материала, питание»).

Threat Intelligence можно классифицировать следующим образом: по способу получения (например, из открытых источников или путём обмена информацией между специалистами), по типу данных (фиды, представленные в виде хешей вредоносных файлов, DDoS-фиды, C&C-фиды, отчёты о деятельности APT-группировок и др.), а также по влиянию на уровень принятия решений (тактические, технические и стратегические).

Далее представить и описать **пирамиду индикаторов компрометации** в зависимости от сложности получения данных:



18. Анализ данных из открытых источников OSINT. Цели и задачи. OSINT в сфере информационной безопасности.

Пассивные и активные методы OSINT.

Автоматизированные инструменты: Shodan, Metagoofil, Maltego, WHOIS.

Набор запросов для выявления ошибок в безопасности ресурсов Google Dorks.

Примеры. Операторы Google.

<https://www.exploit-db.com/google-hacking-database>

18. Анализ данных из открытых источников OSINT. Цели и задачи. OSINT в сфере информационной безопасности.

Open source intelligence подразумевает получение данных из источников в общественном достоянии и/или таких, доступ к которым возможен по запросу. К ним относятся:

- информационные материалы (статьи, новости, заметки) в СМИ; научные исследования, опубликованные в специализированных изданиях; книги, посты и комментарии в социальных сетях;
- форумы, блоги, сайты обмена видео, такие как YouTube.com, вики, Записи Whois о зарегистрированных доменных именах, метаданных и цифровых файлов, даркнет-ресурсы, данные геолокации, IP-адреса;
- информация из переписки;
- документы из открытых государственных и негосударственных архивов;
- публичные коммерческие данные (доход, прибыль, убыток, рост, стоимость акций и т.д.);
- результаты публичных опросов;
- данные со спутников дистанционного зондирования Земли и самолетов аэрофотосъемки;
- полицейские и судебные документы и другие источники.

** Сбор и анализ информации, находящейся в общественном достоянии, не противоречат нормам международного законодательства, а также законам большинства государств, хотя некоторые источники и способы их исследования могут находиться на грани законности.*

18. Анализ данных из открытых источников OSINT. Цели и задачи. OSINT в сфере информационной безопасности.

С развитием интернета фокус внимания аналитиков сместился в киберпространство как один из главных источников информации. Здесь полезными данными могут являться:

- регистрационные сведения о сертификате или домене сайта;
- открытые персональные данные пользователей (username, адреса электронной почты, номера телефонов);
- пользовательская активность в социальных сетях (посты, комментарии и т.д.);
- пользовательские запросы в поисковых системах;
- HTML-код сайта;
- публичные текстовые, графические, аудио-, видеофайлы и их метаданные (например, дата, время и место создания, использованное устройство);
- геолокационные данные и другие виды информации.

** Ко многим данным можно получить доступ через открытый интернет с помощью ресурсов, индексируемых поисковыми системами. Однако и источники из «глубинной Сети», к которым у обычных пользователей нет доступа из-за необходимости платить за них, тоже попадают под определение open source. Иными словами, OSINT работает со всеми данными, которые не являются конфиденциальными, не составляют коммерческую или государственную тайну.*

18. Анализ данных из открытых источников OSINT. Цели и задачи. Методы OSINT.

Все методы и инструменты, используемые для анализа данных из открытых источников, можно разделить на две категории.

Пассивные

Позволяют получать общую информацию об объекте. Она собирается вручную или с помощью специальных сервисов и инструментов, упрощающих сбор, систематизацию и анализ данных. Например, программ для парсинга сайтов.

К пассивным методам можно отнести:

- сбор информации (в том числе по фотографиям) из открытых поисковых систем;
- анализ пользовательской активности в социальных сетях и блогах, на форумах, иных виртуальных платформах;
- поиск открытых данных пользователей в социальных сетях, мессенджерах;
- просмотр сохраненных копий сайтов в поисковых системах, интернет-архиве;
- получение геолокационных данных с помощью общедоступных ресурсов вроде Google Maps или Яндекс.Карты.

18. Анализ данных из открытых источников OSINT. Цели и задачи. Методы OSINT.

Активные

Такие методы подразумевают непосредственное влияние аналитика на исследуемый объект, использование специализированных средств получения данных или совершение действий, требующих определенных усилий, например:

- сбор данных на закрытых ресурсах, доступ к которым возможен только по подписке;
- применение специализированных сервисов и программ, которые активно воздействуют на исследуемый объект — например, автоматически регистрируются на сайте;
- использование сервисов, сканирующих приложения, файлы или сайты на наличие вредоносного кода;
- создание поддельных веб-ресурсов, каналов в мессенджерах, собирающих данные пользователей, конфиденциальные или секретные сведения.

В логике OSINT пассивные методы, направленные на сбор общей информации из легкодоступных источников, предваряют применение активных способов, предназначенных для сбора конкретных данных об объекте.

18. Автоматизированные инструменты OSINT: Shodan, Metagoofil, Maltego, WHOIS.

Shodan

Это поисковая система, предназначенная для нахождения подключенных к интернету устройств по IPv4-адресам (роутеры, камеры видеонаблюдения, датчики безопасности и т.д.).

Сама система не наносит вреда, но с ее помощью любой желающий при должном старании может найти незащищенное или плохо защищенное устройство.

<https://www.shodan.io/>

Metagoofil

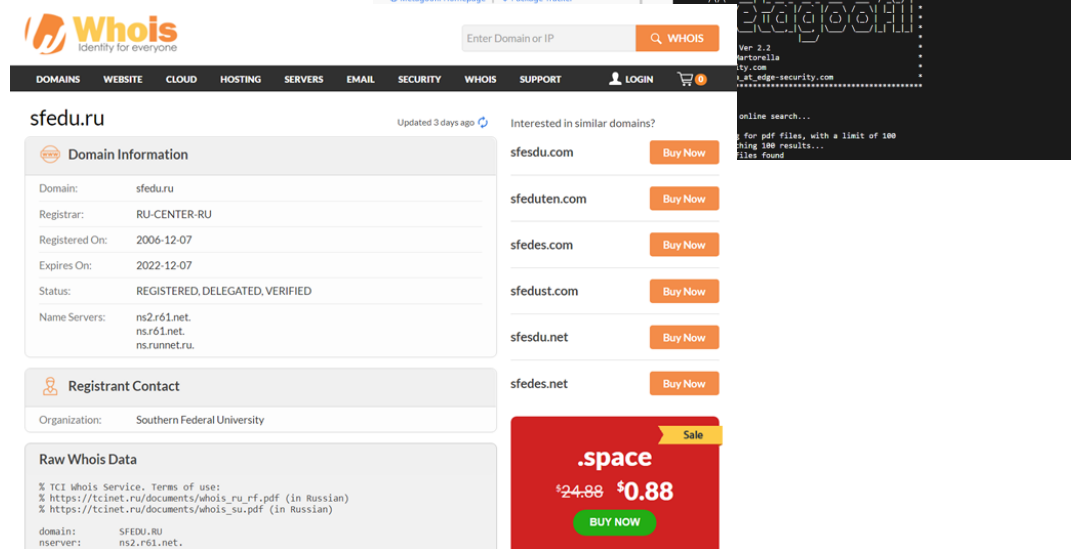
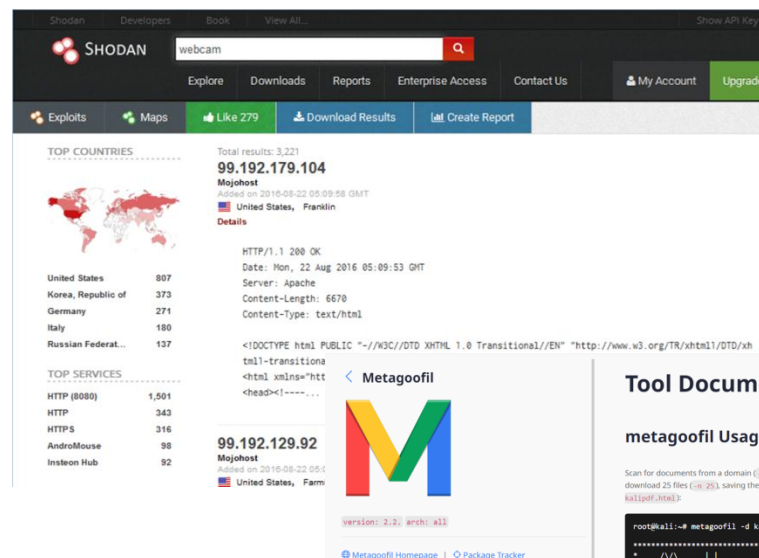
Это метапоисковая система, которая использует другие поисковики для нахождения и извлечения находящихся в открытом доступе файлов PDF, Word, Powerpoint и Excel. С ее помощью можно парсить техническую документацию, клиентские базы данных, справочники, каталоги и прочие полезные источники.

<https://www.kali.org/tools/metagoofil/>

Maltego — это программа, которая может быть использована для выявления отношений и реальных связей между:

- Людьми, Группами людей (социальные сети), Компаниями, Организациями, Веб-сайтами
- Интернет инфраструктурами, такими как: *Доменами, DNS именами, Сетевыми блоками, IP адресами.*

WHOIS (от англ. who is — «кто это?») — сетевой протокол прикладного уровня, базирующийся на протоколе TCP (порт 43). Основное применение — получение регистрационных данных о владельцах доменных имён, IP-адресов и автономных систем.



18. Автоматизированные инструменты OSINT:

Google Dorks.

Google Dorks или Google Hacking — техника, используемая СМИ, следственными органами, инженерами по безопасности и любыми пользователями для создания запросов в поисковых системах для обнаружения скрытой информации и уязвимостях, которые можно обнаружить на общедоступных серверах. Это метод, в котором обычные запросы на поиск веб-сайтов используются в полную меру для определения информации, скрытой на поверхности.

** Google Dork или Google Dork Queries (GDQ) — это набор запросов для выявления грубейших дыр в безопасности ресурсов.*

Операторы Google:

site — искать по конкретному сайту;

inurl — указать на то, что искомые слова должны быть частью адреса страницы / сайта;

intitle — оператор поиска в заголовке самой страниц;

ext или filetype — поиск файлов конкретного типа по расширению.

| — оператор OR он же вертикальный слеш (логическое или) указывает, что нужно отобразить результаты, содержащие хотя бы одно из слов, перечисленных в запросе.

«» — оператор кавычки указывает на поиск точного соответствия.

— — оператор минус используется для исключения из выдачи результатов с указанными после минуса словами.

* — оператор звездочка, или астериск используют в качестве маски и означает «что угодно».

18. Автоматизированные инструменты OSINT: Google Dorks.

Онлайн-сервис Exploit-DB — это некоммерческий проект Offensive Security, данная компания занимается обучением в области информационной безопасности, а также предоставляет услуги пентеста.

База данных Exploit-DB насчитывает огромное количество дорков и уязвимостей.

**Для поиска дорков зайдите на сайт [exploit-db.com](https://www.exploit-db.com) и перейдите на вкладку «Google Hacking Database».*

<https://www.exploit-db.com/google-hacking-database>

Dork	Category	Author
<code>intext:"index of" ".sql"</code>	Files Containing Juicy Info	Gopalsamy Rajendran
<code>intitle:"index of" inurl:superadmin</code>	Files Containing Juicy Info	Mahedi Hassan
<code>intitle:"WAMPSEVER Homepage"</code>	Files Containing Juicy Info	HackerFrenzy
<code>inurl: json beautifier online</code>	Files Containing Juicy Info	Nyein Chan Aung
<code>intitle:"IIS Windows Server"</code>	Files Containing Juicy Info	HackerFrenzy
<code>intitle:"index of" inurl:SUID</code>	Files Containing Juicy Info	Mahedi Hassan

19. Компьютерная криминалистика (Computer Forensics). Определение, цели и задачи. Специальные методы, применяемые для раскрытия и расследования компьютерных преступлений.

Приведите этапы и примеры инструментов сбора, анализа, выявления доказательств.

Например, сбор цифровых доказательств в соответствии с Федеральным законом от 31.05.2001 г. № 73-ФЗ "О государственной судебно-экспертной деятельности в Российской Федерации".

Сохранность информации и данных.

Источники криминалистически значимой компьютерной информации. Документация накопителей и аппаратной конфигурации систем, Защита от записи, Логирование действий.

Приведите примеры программных блокираторов записи, аппаратных криминалистических копировальщиков с блокировкой записи.

20. Компьютерная криминалистика (Computer Forensics). Волатильные данные и их сбор – Live Forensics. Какую информацию можно с помощью них найти и проанализировать?

Прокомментируйте следующие виды данных:

- данные в состоянии изменения;
- данные, содержащиеся в активной физической памяти;
- данные, которые существуют в транзите;
- системные данные, которые теряются при потере мощности, отключении ПК, холодной перезагрузке.

Live Forensics лучшими практиками являются:

- Использование криминалистического ПО, способного производить побитовое чтение данных через локальную сеть и сохранять данные уже на защищенной системе (например, ПО, которое использует протокол iSCSI, обеспечивающий доступ на уровне блоков к устройствам хранения путем переноса команд SCSI через сеть TCP/IP).
- Использование аппаратного обеспечения для создания чистого дампа памяти через прямой доступ к памяти DMA в сочетании с программным подходом (может быть использовано для сравнительного анализа).

Правило волатильности: «RFC 3227: Рекомендации по сбору и архивированию доказательств»

Порядок для типичной системы (RFC 3227):

- регистры
- оперативная память, кэши, таблица маршрутизации, таблица процессов, статистика ядра
- временные файловые системы
- диск
- физическая конфигурация, сетевая топология
- архивы и файлы

Общее правило: при сборе доказательств вы должны перейти от более к менее волатильным.