Классификация вредоносных программ:

- Жизненный цикл вредоносных программ.
- Основные каналы распространения вредоносных программ.
- Уязвимости программного обеспечения («переполнение буфера», «манипуляция указателя» и др.).
- База данных уязвимостей CVE.

Определение вируса согласно ГОСТ Р 51198-98

Вирус – программа, способная создавать свои копии (необязательно совпадающие с оригиналом) и внедрять их в файлы, системные области компьютера, компьютерных сетей, а также осуществлять иные деструктивные действия. При этом копии сохраняют способность дальнейшего распространения. Компьютерный вирус относится к вредоносным программам .

Проблема – под вирусом чаще всего понимается практически любая вредоносная программа. Это приводит к путанице в терминологии, осложненной еще и тем, что сегодня практически любой антивирус способен выявлять все типы вредоносных программ, таким образом ассоциация «вредоносная программа-вирус» становится все более устойчивой.

Вредоносная программа – компьютерная программа или переносной код, предназначенный для реализации угроз информации, хранящейся в компьютерной системе (КС), либо для скрытого нецелевого использования ресурсов КС, либо иного воздействия, препятствующего нормальному функционированию КС. К вредоносным программам относятся компьютерные вирусы, трояны, сетевые черви и др.

Классификация компьютерных вирусов:

- Загрузочные вирусы
- Файловые вирусы
- Макровирусы
- Скрипт-вирусы

Сетевой червь – тип вредоносных программ, распространяющихся по сетевым каналам, способных к автономному преодолению систем защиты автоматизированных и компьютерных систем, а также к созданию и дальнейшему распространению своих копий, не всегда совпадающих с оригиналом, осуществлению иного вредоносного воздействия.

Классификация сетевых червей:

В зависимости от путей проникновения в операционную систему черви делятся на:

- Почтовые черви(Mail-Worm)
- IM черви(IM-Worm)
- P2P черви(P2P-Worm)
- Сетевые черви(Net-Worm)

Троянские программы – тип вредоносных программ, основной целью которых является вредоносное воздействие по отношению к компьютерной системе. Трояны отличаются отсутствием механизма создания собственных копий.

Классификация троянских программ:

- Похитители паролей (Trojan-PSW)
- Клавиатурные шпионы (Trojan-SPY)
- Утилиты удаленного управления (Backdoor)
- Анонимные smtp-сервера и прокси (Trojan-Proxy)
- Модификаторы настроек браузера (Trojan-Cliker)
- Инсталляторы прочих вредоносных программ (Trojan-Dropper)
- Загрузчики вредоносных программ (Trojan Downloader)
- Уведомители об успешной атаке (Trojan-Notifier)
- Логические бомбы
- Утилиты дозвона

Жизненный цикл вредоносных программ

Жизненный цикл вирусов может быть условно разделён на следующие стадии:

1. Активация вируса
2. Поиск объектов для заражения
3. Подготовка вирусных копий
4. Внедрение вирусных копий

Жизненный цикл червей можно разделить на следующие стадии:

1. Проникновение в систему
2. Активация
3. Поиск «жертв»
4. Подготовка копий
5. Распространение копий

Жизненный цикл троянских программ можно разделить на следующие стадии:

1. Проникновение на компьютер
2. Активация
3. Выполнение заложенных функций

Основные каналы распространения вредоносных программ

Способы заражения вредоносным кодом

Файловые вирусы различными способами внедряются в исполнимые файлы (программы) и обычно активизируются при их запуске. После запуска зараженной программы вирус находится в оперативной памяти компьютера и является активным (то есть может заражать другие файлы) вплоть до момента выключения компьютера или перезагрузки операционной системы. При этом файловые вирусы не могут заразить файлы данных (например, файлы, содержащие изображение или звук). Часто файловые вирусы пытаются добавить себя в автозапуск или указать себя в качестве программы для открытия приложений.

Загрузочные вирусы записывают себя в загрузочный сектор диска. При загрузке операционной системы с зараженного диска вирусы внедряются в оперативную память компьютера. В дальнейшем загрузочный вирус ведет себя так же, как файловый, то есть может заражать файлы при обращении к ним компьютера.

Макровирусы заражают файлы документов Word и электронных таблиц Excel. Они являются фактически макрокомандами (макросами), которые встраиваются в документ. После загрузки зараженного документа в приложение макровирусы постоянно присутствуют в памяти компьютера и могут заражать другие документы. Угроза заражения прекращается только после закрытия приложения.

Профилактическая защита от макровирусов состоит в предотвращении запуска вируса. При открытии документа в приложениях Word и Excel сообщается о присутствии в них макросов (потенциальных вирусов) и предлагается запретить их загрузку.

Файловые вирусы и макровирусы могут распространятся через компьютерную сеть Это может происходить, например, при получении зараженных файлов с серверов файловых архивов или облачных или сетевых хранилищ.

Интернет-черви (worm) — это вирусы, которые распространяются в компьютерной сети во вложенных в почтовое сообщение файлах. Автоматическая активизация червя и заражение компьютера могут произойти при обычном просмотре сообщения. Интернет-черви часто выполняют роль «троянского коня», внедренного в операционную систему.

Лавинообразная цепная реакция распространения вредоноса базируется на том, что вирус после заражения компьютера начинает рассылать себя по всем адресам электронной почты, которые имеются в адресной книге пользователя. Кроме того, может происходить заражение и по локальной сети, так как червь перебирает все локальные диски и сетевые диски с правом доступа и копируется туда под случайным именем. Профилактическая защита от интернет-червей состоит в том, что не рекомендуется открывать вложенные в почтовые сообщения файлы, полученные из сомнительных источников.

Особой разновидностью вирусов являются активные элементы (программы) на языках JavaScript или VBScript, которые могут выполнять разрушительные действия, то есть являться вирусами (скрипт-вирусами). Такие программы передаются по Inernet в процессе загрузки Web-страниц с серверов Интернета в браузер локального компьютера. Профилактическая защита от скрипт-вирусов состоит в том, что в браузере можно запретить получение активных элементов на локальный компьютер.

Троянские программы (программные закладки) – могут распространятся самостоятельно, как вышеперечисленные вредоносы, так и путём использования уязвимостей:

- активация вредоносного кода при попытке открытия специально подготовленных файлов или обработки специально подготовленных данных программами и компонентами ОС (вредоносный код в изображениях, документах, флэш-анимации, html-страницах и др.).
- рассылка на атакуемые узлы в сети специально сформированных сетевых пакетов, содержащих: компонент, вызывающий ошибку (при попытке обработать сетевой пакет или данные, содержащиеся в нём) и «полезную нагрузку» (код троянской программы, командную оболочку и др.). Кроме этого, «полезная нагрузка» может маскироваться (шифрованием) и могут добавляться компоненты для скрытого удалённого управления (VPN-соединения) и маскирования вредоноса в заражённой системе (rootkit).

Для совершения атак на узлы применяют эксплоиты (exploits) – специальные программы (обычно в виде скриптов) формирующие вредоносные сетевые пакеты или блоки данных.

MetasploitFramework – известная среда для разработки и применения эксплоитов для решения задач информационной безопасности (pentesting – тестирование на проникновение).

Троянские программы различаются между собой по тем действиям, которые они производят на зараженном компьютере следующим образом:

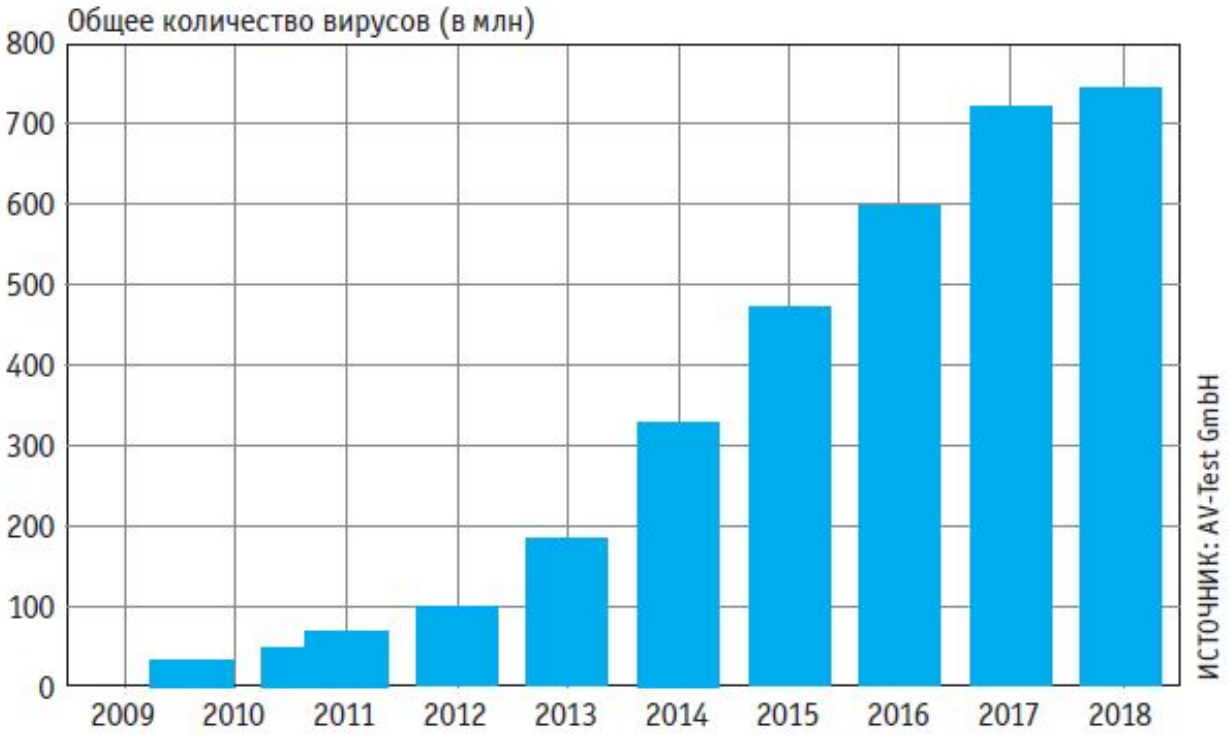
- Backdoor – троянские утилиты удаленного администрирования;
- Trojan-PSW – воровство паролей;
- Trojan-Clicker – интернет-кликеры;
- Trojan-Downloader – доставка прочих вредоносных программ;
- Trojan-Dropper – инсталляторы прочих вредоносных программ;
- Trojan-Proxy – троянские прокси-сервера;
- Trojan-Spy – шпионские программы;
- Trojan – прочие троянские программы;
- Rootkit – сокрытие присутствия в операционной системе;
- ArcBomb – «бомбы» в архивах;
- Trojan-Notifier – оповещение об успешной атаке.

К хакерским утилитам и прочим вредоносным программам можно отнести:

- утилиты автоматизации создания вирусов, червей и троянских программ (конструкторы);
- программные библиотеки, разработанные для создания вредоносного ПО;
- хакерские утилиты скрытия кода зараженных файлов от антивирусной проверки (шифровальщики файлов);
- «злые шутки», затрудняющие работу с компьютером;
- программы, сообщающие пользователю заведомо ложную информацию о своих действиях в системе;
- прочие программы, тем или иным способом намеренно наносящие прямой или косвенный ущерб данному или удаленным компьютерам.

Представителями таковых являются:

- DoS, DDoS – сетевые атаки;
- Exploit, HackTool – взломщики удаленных компьютеров;
- Flooder – «замусоривание» сети;
- Constructor – конструкторы вирусов и троянских программ;
- Nuker – фатальные сетевые атаки;
- Bad-Joke, Hoax – злые шутки, введение пользователя в заблуждение;
- FileCryptor, PolyCryptor – скрытие от антивирусных программ;
- PolyEngine – полиморфные генераторы;
- VirTool – утилиты, предназначенные для облегчения написания компьютерных вирусов.



Распространение вирусов за последние 10 лет. На конец 2017 года количество известных вирусов превысило 700 млн (https://ichip.ru/sovety/sravnivaem-15-antivirusov-s-zashhitnikom-windows-271392)

Сигнатура вируса – в широком смысле, информация, позволяющая однозначно определить наличие данного вируса в файле или ином коде.

Пример сигнатуры – уникальная последовательность байт, присутствующая в данном вирусе и не встречающаяся в других программах; контрольная сумма такой последовательности.

Современные вредоносные программы используют сложные технологии маскировки и защиты своих копий, которые обуславливают необходимость применение специальных средств для их анализа.

Технологии создании копий для маскировки:

- Шифрование
- Метаморфизм

Сочетание этих двух технологий приводит к появлению следующих типов вирусов.

- Шифрованный вирус
- Метаморфный вирус
- Полиморфный вирус

Программные уязвимости

Согласно терминологии MITRE CVE:

Уязвимость—это состояние вычислительной системы (или нескольких систем), которое позволяет:

- исполнять команды от имени другого пользователя;
- получать доступ к информации, закрытой от доступа для данного пользователя;
- показывать себя как иного пользователя или ресурс;
- производить атаку типа «отказ в обслуживании».

В MITRE считают, что атака, производимая вследствие слабой или неверно настроенной политики безопасности, лучше описывается термином «открытость» (exposure).

Открытость — это состояние вычислительной системы (или нескольких систем), которое не является уязвимостью, но:

- позволяет атакующему производить сбор защищенной информации;
- позволяет атакующему скрывать свою деятельность;
- содержит возможности, которые работают корректно, но могут быть легко использованы в неблаговидных целях;
- является первичной точкой входа в систему, которую атакующий может использовать для получения доступа или информации.

Уязвимости программного обеспечения («переполнение буфера», «манипуляция указателя» и др.).

База данных уязвимостей CVE и ФСТЭК.

Литература:

1. https://studopedia.ru/9_116035_setevie-chervi.html

2. https://www.opennet.ru

Специализированные средства и методы выявления вредоносных программ:

- Сигнатурное и эвристическое сканирование, контроль целостности, мониторинг информационных потоков, изолированная программная среда, программные ловушки.
- Антивирусные программы и комплексы.

Сигнатурное и эвристическое сканирование, контроль целостности, мониторинг информационных потоков, изолированная программная среда, программные ловушки.

Традиционный подход при обнаружении вредоносных программ (ВП) заключается в анализе образцов кода, которые являются уникальными для определенного вредоносного приложения. Исходный образец служит для того, чтобы создать по нему, так называемую, сигнатуру, и состоит обычно из нескольких участков кода программы. Например, сигнатура должна быть идентична трем частям кода по 50 байт, расположенных друг от друга на определенном расстоянии по коду в файле. Комбинация частей переделенного кода и их взаимного расположения и составляют обычную сигнатуру, по которой можно однозначно определить, к какому приложению относится запускаемый файл.

Основные методы сигнатурного обнаружения вредоносного кода:

- Метод прямого поиска («грубой силы»).
- Алгоритм Рабина.
- Алгоритм Бойера-Мура.
- Алгоритм Ахо-Корасика.
- Метод дерева.
- Нейросетевые классификаторы.

Антивирусные программы и комплексы (средства обнаружения вредоносного кода).

Чаще всего серьезные программные средства обнаружения вредоносного кода носят название «антивирус». Их принято разделять на:

- Сканеры (устаревший вариант «полифаги»);
- Ревизоры;
- Сторожам (мониторы);
- Вакцины.

Сканеры определяют наличие вируса по базе данных, хранящей сигнатуры (или их контрольные суммы) вирусов. Их эффективность определяется акутальностью вирусной базы и наличием эвристического анализатора.

Ревизоры запоминают состояние файловой системы, что делает в дальнейшем возможным анализ изменений. (Класс близкий к IDS).

Сторожа отслеживают потенциально опасные операции, выдавая пользователю соответствующий запрос на разрешение или запрещение операции.

Вакцины изменяют прививаемый файл таким образом, чтобы вирус, против которого делается прививка, уже считал файл зараженным. В современных условиях, когда количество возможных вирусов измеряется десятками тысяч, этот подход неприменим.

Популярные антивирусные продукты:

- Avira AntiVir
- Avas Antivirus
- AVG Anti-virus
- ClamAV
- Dr.Web
- Eset NOD32
- Kaspersky Anti-Virus
- McAfee Antivirus
- Microsoft Security Essentials.
- Symantec Norton AntiVirus.
- Panda Antivirus.

Сравнительный тест антивирусов (https://ichip.ru/sovety/sravnivaem-15-antivirusov-s-zashhitnikom-windows-271392):

СРАВНИТЕЛЬНЫЙ ТЕСТ ПАКЕТОВ БЕЗОПАСНОСТИ

Несмотря на то, что Kaspersky Internet Security не занял первое место, лицензия на этот очень хороший антивирус (1 год/3 устройства) доступна всего за 1990 рублей		Общая оценка	Цена (прибл.) в руб.	Эффективность защиты (50%)	Ложные срабатывания (25%)	Производительность (25%)	Распознавание угроз нулевого дня (в %)	Распознавание известных вирусов (в %)	Ложные срабатывания при работе с веб-страницами/программами	Замедление на стандартных ПК (в %)	Замедление на мощных ПК (в %)
1	Bitdefender Internet Security	97,5	2200	100	90,6	99,5	100	100	○/●	7,79	10,41
2	Symantec Norton Security	97,1	1799	100	95,8	92,8	100	100	○/●	11,78	13,34
3	Kaspersky Internet Security	96	1990	92	100	100	84,2	99,8	○/●	7,51	10,23
4	Trend Micro Internet Security	94,7	2600	92,6	97,5	96	85,2	100	○/●	11,29	10,51
5	AVG Internet Security	89,6	4600	90,9	89,9	86,8	92,1	89,8	○/●	14,89	16,37
6	McAfee Internet Security	89,3	1300	83,3	98	92,7	69,1	97,6	○/●	13,76	11,47
7	F-Secure Safe	88,9	3100	93,5	81,2	87,6	92,1	94,9	○/●	13,49	16,95
8	Avast Free Antivirus	88,7	беспл.	89	89,9	86,7	87,7	90,4	○/●	14,82	16,53
9	Avira Antivirus Pro	85,7	2100	75,2	98	94,6	66,8	83,6	○/●	11,28	12,01
10	BullGuard Internet Security	78,4	1200	67	90,1	89,4	39,2	94,7	○/●	13,03	15,53
11	MicroWorld eScan Internet Security Suite	73,8	4400	52,6	90,6	99,4	9,6	95,7	○/●	7,95	10,43
12	G Data Internet Security	73,5	2600	62,8	88,3	80,2	30	95,6	○/●	16,72	21,24
13	Comodo Internet Security Premium	69,3	беспл.	64,9	90,4	56,9	100	29,8	○/●	22,85	38,91
14	ESET Internet Security	68,6	2200	56,1	96,5	65,6	33,3	78,8	○/●	31,39	21,50
15	Microsoft Windows Defender	60,6	беспл.	48,6	59,7	85,4	25,6	71,6	○/●	14,88	17,81
16	K7 Computing Total Security	58,9	3000	32,7	89,4	80,6	30,7	34,6	○/●	12,97	24,57

Высший класс (100–90) Высокий класс (89–75) Средний класс (74–60) Начальный класс (59–45) Не рекомендуется (44–0)
Все оценки в баллах (максимум — 100); ●Да ○Нет

Антивирус ClamAV (www.clamav.com). Данный антивирус имеет статус opensource проекта, распространяется по лицензии GPLv2. В настоящее время проект куплен компанией Cisco Systems для интеграции в сетевое оборудование. Проект начал функционировать в 2002 году. Данный антивирус является фактически единственным эффективным антивирусом с открытым исходным кодом, составляющий конкуренцию коммерческим антивирусам.

Особенности ClamAV. Основное назначение ClamAV – сканирование файлов на почтовых серверах и т.д. Содержит сотни тысяч сигнатур вредоносного кода. Не позволяет лечить файлы – заражённые файлы либо удаляются, либо помещаются в карантин. Использует в основном сигнатурный метод поиска компьютерных вирусов. Упрощённа структура ClamAV изображена на рисунке 1.

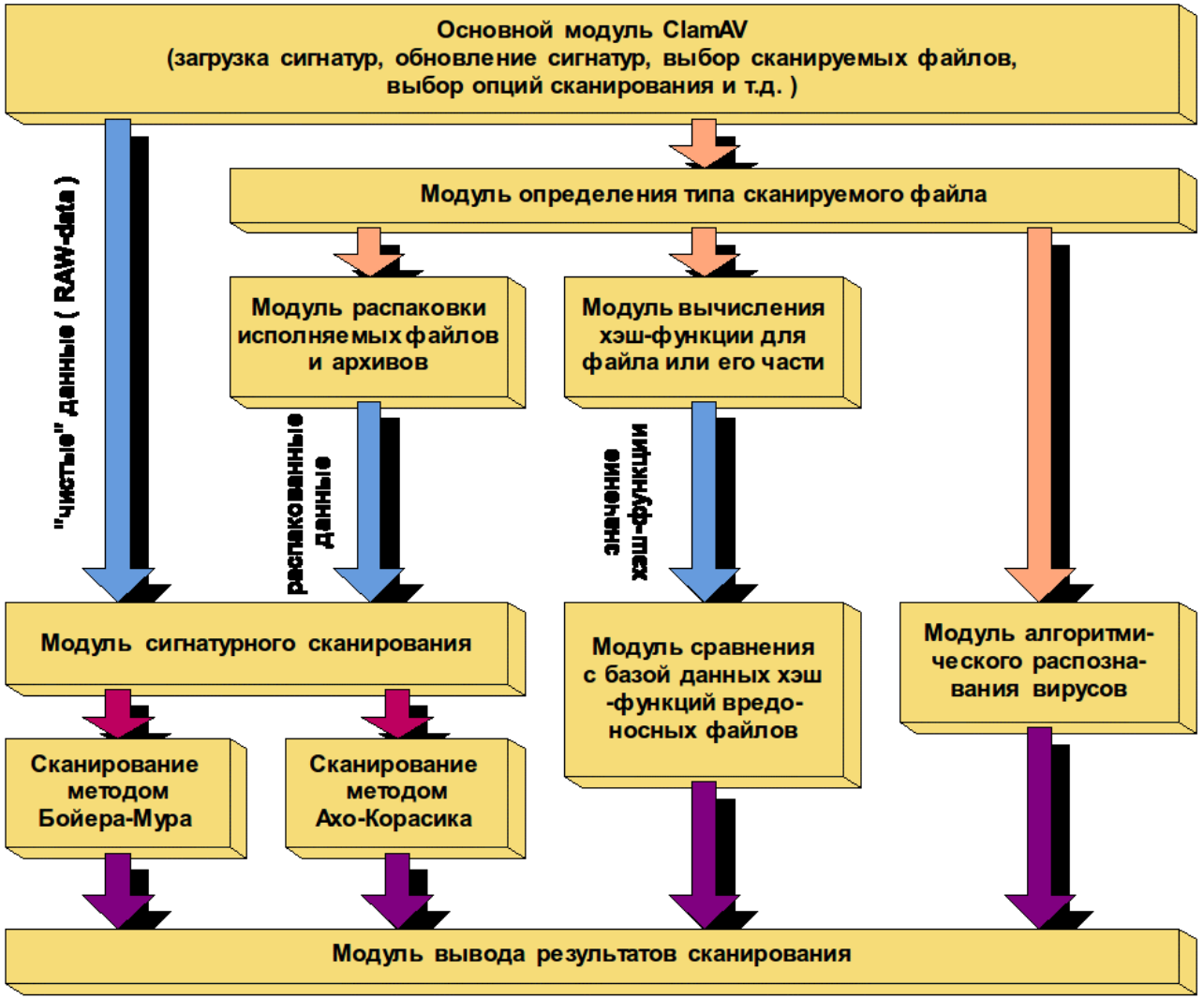


Рис. 1. Упрощённая структура антивируса ClamAV

ClamAV использует несколько вариантов описания сигнатур вирусов:

Описание сигнатуры в виде последовательности шестнадцатеричных цифр. Данный тип сигнатур хранится в файле с расширением *.db. Пример сигнатуры: «Phantom.4=0190e800005e56ba4c0881ea000183ee». Допускается использование масок, т.е. замена шестнадцатеричных значений символами «*», «?» и т.д.

Сигнатуры описываются так же, как и в предыдущем случае. Дополнительно указывается номер и тип секции исполняемого файла, в котором может находиться вредоносный код. Дополнительные сведения служат для увеличения скорости работы антивируса. Данный тип сигнатур хранится в файле с расширением *.ndb. Пример сигнатуры: «W32.MyLife.E:1:*:7a6172793230*40656d61696c2e636f6d»

Сигнатура представляется в виде результата вычисления хэш-функции от файла, содержащего вредоносный код. Данный вариант применим в случае, когда вирус размещается в отдельном файле. Неприменим для полиморфных и самошифрующихся вирусов, а так же для вирусов, заражающих другие файлы. Достоинство – простота получения сигнатуры для вируса. Для ускорения работы антивируса дополнительно указывается размер файла, содержащий вредоносный код. Это позволяет исключить вычисление хэш-функций для файлов, размер которых не соответствует размерам, приведённых в базе сигнатур. Данный тип сигнатур хранится в файлах с расширением *.mdb и *.hdb. Пример сигнатуры:

«36864:d1a320843e3a92fdbb7d49137f9328a0:Trojan.Agent-1701»

Быстродействие ClamAV. В ClamAV применяется три основных метода поиска сигнатур в файлах: метод Бойра-Мура, метод Ахо-Корасика и сравнение результатов хэш-функции. Следует отметить, что со времени запуска проекта ClamAV (2002 год) программные реализации методов Бойра-Мура и Ахо-Корасика постоянно совершенствовались, особенно в плане быстродействия.

Основы безопасности операционных систем:

- Классификация угроз безопасности ОС, типичные атаки на ОС.
- Понятие защищённой ОС: основные определения, подходы к построению защищённых ОС, административные меры защиты, адекватная политика безопасности.
- Основные виды уязвимостей в ОС.
- Основные задачи аппаратного обеспечения защиты информации: управление оперативной памятью, планирование задач, синхронизация параллельных задач, обеспечение корректности совместного доступа к объектам, предотвращение тупиковых ситуаций.
- Аппаратная защита в процессорах с архитектурами x86 и x86-64: адресация оперативной памяти, уровни привилегированности, защита сегментов оперативной памяти.
- Расширения команд процессоров AES-NI и NX-Bit (ND-Bit). Модуль доверенной платформы Trusted Platform Module (TPM).

Классификация угроз безопасности ОС, типичные атаки на ОС

Классификация угроз по цели:

- несанкционированное чтение информации;
- несанкционированное изменение информации;
- несанкционированное уничтожение информации;
- полное или частичное разрушение операционной системы.

Под разрушением операционной системы (ОС) понимается целый комплекс разрушающих воздействий от кратковременного вывода из строя отдельных программных модулей системы до физического стирания системных файлов с дисков.

Классификация угроз по принципу воздействия на ОС:

- использование известных каналов получения информации (например, угроза несанкционированного чтения файла, доступ к которому определен некорректно);
- использование скрытых каналов получения информации (например, угроза использования злоумышленником недокументированных возможностей ОС);
- создание новых каналов получения информации с помощью программных закладок.

Классификация угроз по характеру воздействия на ОС:

- активное воздействие – несанкционированные действия злоумышленника в системе;
- пассивное – несанкционированное наблюдение злоумышленника за процессами, происходящими в системе.

Классификация угроз по используемых злоумышленником слабостей защиты:

- неадекватная политика безопасности, в том числе и ошибки администратора системы;
- ошибки и недокументированные возможности программного обеспечения ОС, в том числе «люки» – случайные или преднамеренно встроенные в систему «служебные входы», позволяющие входить в систему защиты. Обычно «люки» создаются разработчиками программного обеспечения (ПО) для тестирования и отладки, и иногда разработчики забывают их удалить или оставляют специально.

Классификация угроз по способу воздействия на объект атаки:

- непосредственное воздействие;
- превышение пользователем своих полномочий;

- работа от имени другого пользователя;
- использование результатов работы другого пользователя (например, несанкционированный перехват информационных потоков, инициированных другим пользователем).

Классификация угроз по способу действия злоумышленника:

- в интерактивном режиме (вручную);
- в пакетном режиме (с помощью специально написанной программы, которая выполняет негативные воздействия на ОС без непосредственного участия нарушителя).

Классификация угроз по объекту атаки:

- операционная система в целом;
- объекты операционной системы (файлы, устройства и т.д.);
- субъекты операционной системы (пользователи, системные процессы и т.д.);
- каналы передачи данных.

Классификация угроз по используемым средствам атаки:

- штатные средства ОС без использования дополнительного ПО;
- ПО «третьих» фирм (к этому классу ПО относятся компьютерные вирусы и другие вредоносные программы, которые легко можно найти в Internet, а также ПО, изначально разработанное для других целей – отладчики, сетевые мониторы, и т.д.).

Классификация угроз по состоянию атакуемого объекта операционной системы на момент атаки:

- хранение;
- передача;
- обработка.

Типичные атаки на операционную систему

1. Сканирование файловой системы. Данная атака является одной из наиболее тривиальных, но в то же время одной из наиболее опасных. Суть атаки заключается в том, что злоумышленник путем просмотра файловой системы пытается получить информацию. Данная атака эффективна при некорректных настройках прав доступа к файлам системы.

2. Кража ключевой информации.

3. Подбор пароля.

4. Сборка «мусора». Во многих операционных системах информация, уничтоженная пользователем, для увеличения производительности системы не уничтожается физически, а только помечается как уничтоженная. Суть данной атаки заключается в восстановлении и анализе такой информации (так называемого мусора) специальными программными средствами.

5. Превышение полномочий. При реализации данной угрозы злоумышленник, используя ошибки в программном обеспечении и политике безопасности, получает более высокие полномочия, чем предоставленные в соответствии с политикой безопасности.

6. Программные закладки.

7. «Жадные» программы. «Жадными» называются программы, преднамеренно захватывающие

значительную часть ресурсов системы, в результате чего другие программы не могут выполняться или выполняются крайне медленно и неэффективно.

Понятие защищённой ОС: основные определения, подходы к построению защищённых ОС, административные меры защиты, адекватная политика безопасности.

Защищенная операционная система – это ОС, предусматривающая средства защиты от основных классов угроз безопасности.

Защищенная ОС обязательно должна содержать средства разграничения доступа к своим ресурсам, а также средства проверки подлинности пользователя, начинающего работу с ОС. Кроме того, защищенная ОС должна содержать средства противодействия преднамеренному выводу ОС из строя.

Частично защищенная операционная система – это ОС, предусматривающая защиту не от всех основных классов угроз безопасности.

Политика безопасности – это набор норм, правил и практических приемов, регулирующих порядок хранения и обработки ценной информации.

Адекватная политика безопасности – политика безопасности, обеспечивающая достаточный уровень защищенности ОС.

Следует особо отметить, что адекватная политика безопасности – это не обязательно та политика безопасности, при которой достигается максимально возможная защищенность ОС. Адекватность политики безопасности определяется не только архитектурой ОС, но и ее конфигурацией, установленными программами и т.д.

Подходы к построению защищенных операционных систем

Существует два основных подхода к созданию защищенных ОС: фрагментарный и комплексный.

Фрагментарный подход – это подход, предполагающий организацию защиты вначале от одной угрозы, затем от другой и т.д. Примером фрагментарного подхода является установка на незащищенную ОС антивирусного пакета, системы шифрования и т.д.

Недостатки фрагментарного подхода:

1) разрозненность программных продуктов подсистемы защиты ОС, что значительно затрудняет осуществление их тесного взаимодействия;
2) возможность некорректной работы отдельных элементов подсистемы защиты вследствие их совместного функционирования. Это может привести к значительному снижению надежности ОС;
3) возможность отключения злоумышленником отдельных функций защиты.

Комплексный подход – это подход, предполагающий внесение защитных функций в ОС на этапе проектирования ее архитектуры. При этом защитные функции являются неотъемлемой частью ОС.

Административные меры защиты

Организация эффективной и надежной защиты ОС невозможна при помощи только программно-аппаратных средств. Без постоянной квалифицированной поддержки со стороны администратора

неэффективна даже самая надежная программная или аппаратная защита.

Основные административные меры защиты:

1. Постоянный контроль корректности функционирования ОС, особенно ее подсистемы защиты.
2. Организация и поддержание адекватной политики безопасности.
3. Инструктирование пользователей ОС о необходимости соблюдения мер безопасности при работе с ОС и контроль за соблюдением этих мер (например, около 80% НСД происходит по вине пользователей).
4. Регулярное создание и обновление резервных копий, программ и данных.
5. Постоянный контроль изменений в конфигурационных данных и политике безопасности ОС.

Адекватная политика безопасности

Верно следующее утверждение: чем лучше защищена операционная система, тем труднее с ней работать пользователям и администраторам. Это связано с отсутствием интеллекта у системы защиты и обусловлено следующими факторами:

1. Система защиты, не обладающая интеллектом, не всегда способна определить, является ли некоторое действие пользователя злонамеренным. Чем выше защищенность системы, тем шире класс технически легальных действий пользователя, который рассматривается как несанкционированный. Например, если пользователю запрещено создавать файлы, то этот

пользователь не сможет запустить ни одну программу, которой для нормального функционирования необходимо создавать временные файлы.

2. Чем больше в ОС защитных функций, тем больше времени и средств нужно тратить на поддержание защиты.
3. Чем сложнее устроены защитные функции, тем больше аппаратных ресурсов компьютера затрачивается на поддержание функционирования системы защиты и тем меньше ресурсов остается на долю прикладных программ.

Этапы определения и поддержания адекватной политики безопасности:

1. **Анализ угроз**. Рассматриваются возможные угрозы безопасности и выделяются наиболее опасные.
2. **Формирование требований к политике безопасности**. Определяется, какие средства и методы будут применяться для защиты от тех или иных угроз.
3. **Формальное определение политики безопасности**. Результатом данного этапа является развернутый перечень настроек конфигурации ОС и дополнительных пакетов защиты с указанием того, в каких ситуациях какие настройки должны быть выставлены.
4. **Претворение в жизнь политики безопасности**. Задачей данного этапа является приведение конфигурации ОС и дополнительных пакетов защиты в соответствии с определенной политикой безопасности.
5. **Поддержание и коррекция политики безопасности**.

Основные виды уязвимостей в ОС.

https://geektimes.ru/company/ua-hosting/blog/295125/

Рассмотрим первые 5 участников нашего рейтинга, базируясь на данных за 2017 год.

Название ОС	Производитель	Общее число уязвимостей за 2017 год	Общее число уязвимостей за 2016 год	Общее число уязвимостей за все время ведения статистики
Android	Google	**666**	**523**	1357
Linux Kernel	Linux	381	217	**1921**
Iphone Os	Apple	293	161	1277
Windows 10	Microsoft	226	172	451
Windows Server 2016	Microsoft	212	39	251
Windows Server 2008	Microsoft	212	133	981
Mac Os X	Apple	210	215	1888
Windows Server 2012	Microsoft	201	156	606
Windows 7	Microsoft	197	134	838
Windows 8.1	Microsoft	192	154	542
Windows Rt 8.1	Microsoft	124	139	438
Debian Linux	Debian	95	327	1029
Fedora	Fedora project	84	120	441
Ubuntu Linux	Canonical	66	279	867

Watchos	Apple	65	77	231
Windows Vista	Microsoft	64	125	814
Opensuse	Opensuse Project	58	5	119
Leap	Opensuse Project	57	2	60
Leap	Novell	48	260	349
XEN	XEN	44	28	228

Основные виды уязвимостей:

- DoS (Denial of Service / отказ в обслуживании) (эксплойт уязвимости приводит к DoS устройства);
- Обход чего-либо (например, пароля для входа в систему);
- Исполнение кода (возможность злоумышленником выполнить какую-то команду на устройстве жертвы);
- Повреждение памяти;
- Доступ к информации (имеется в виду секретная информация, полученная за счет уязвимости);
- Увеличение привилегий (в частности для вредоносного ПО);
- Переполнение (буфера);

Основные задачи аппаратного обеспечения защиты информации: управление оперативной памятью, планирование задач, синхронизация параллельных задач, обеспечение корректности совместного доступа к объектам, предотвращение тупиковых ситуаций.

Под аппаратным обеспечением средств защиты операционной системы (ОС) традиционно понимается совокупность средств и методов, используемых для решения следующих задач:

1) управление оперативной и виртуальной памятью компьютера;
2) распределение процессорного времени между задачами в многозадачной ОС;
3) обеспечение корректности совместного доступа задач к ресурсам ОС;
4) исключение тупиковых ситуаций в процессе совместного доступа задач к ресурсам ОС.

Управление оперативной памятью

Основная угроза, защита от которой реализуется с помощью средств управления оперативной памятью, заключается в получении одним из процессов несанкционированного доступа к оперативной памяти другого процесса, выполняющегося параллельно.

Существуют два основных подхода к обеспечению защиты оперативной памяти процесса от несанкционированного доступа со стороны других процессов:

Первый подход заключается в том, что при каждом обращении процесса к оперативной памяти

осуществляется проверка корректности доступа. То есть, каждому процессу выделяется отдельная область памяти и блокируются все обращения за ее пределы. Теоретически это подход позволяет создать абсолютно надёжную защиту от несанкционированного доступа к «чужой» памяти. Однако при это становится невозможным взаимодействие процессов.

Второй подход заключается в выделении каждому процессу индивидуального адресного пространства, аппаратно изолированного от других процессов. Для практической реализации этого подхода необходимо, чтобы центральный процессор компьютера поддерживал логическую адресацию оперативной памяти (то есть, автоматически преобразовывал виртуальные адреса в физические незаметно для выполняющегося процесса).

Описанные подходы не являются взаимоисключающими и могут применяться в совокупности.

При любом подходе к защите оперативной памяти ОС должна предусматривать средства отладки программ. А отладка программ невозможна без доступа процесса-отладчика к оперативной памяти отлаживаемого процесса. Использование отладчиков для несанкционированного доступа представляет собой серьезную угрозу безопасности ОС.

Планирование задач

Планирование задач в многозадачной ОС заключается в распределении ОС времени процессора между параллельно выполняющимися задачами.

Планирование задач может быть реализовано без вытеснения прерванных задач или с вытеснением прерванных задач.

При планировании задач без вытеснения выполнение задач может быть прервано только по инициативе самой задачи. Если задача зациклилась в процессе обработки сообщения, то другие задачи никогда не смогут получить управление и ОС «зависает». Такое планирование задач не применимо для защищенных ОС.

При планировании задач с вытеснением выполнение любой задачи может быть прервано в любой момент времени.

Основная угроза подсистеме планирования задач заключается в том, что злоумышленник сможет приостановить выполнение задач, критических для обеспечения безопасности ОС. Для нейтрализации этой угрозы ОС должна обладать следующими свойствами:

1) поддерживается вытеснение задач;
2) создание высокоприоритетных задач доступно только привилегированным пользователям;
3) задачи, критичные для обеспечения безопасности ОС, защищены от несанкционированного вмешательства в ход их выполнения;
4) фатальный сбой в процессе функционирования одной из критичных задач должен вызывать крах ОС.

Предотвращение тупиковых ситуаций

Тупиковая ситуация может возникнуть, когда несколько программ одновременно пытаются открыть несколько одних и тех же объектов в режиме монопольного доступа. Если одна программа открыла одну часть множества объектов, а другая программа – другую часть, то ни одна из программ не сможет открыть остальные объекты до тех пор, пока другая программа их не закроет.

Если функции закрытия объектов в подобной ситуации не предусмотрены ни в одной из программ, ситуация становиться тупиковой. Одним из самых простых методов борьбы с такими ситуациями является проверка программой возможности открытия объектов в режиме монопольного доступа. В случае невозможности доступа к объектам программа должна закрыть все уже открытые объекты, подождать некоторое время и повторить всю операцию с начала.

Аппаратная защита в процессорах с архитектурами x86 и x86-64: адресация оперативной памяти, уровни привилегированности, защита сегментов оперативной памяти.

Процессоры модели i386 и более поздних моделей могут работать в одном из трех режимов – реальном режиме, защищенном режиме и режиме эмуляции виртуального 8086 (виртуальный режим). При старте процессора он начинает работу в реальном режиме, в котором защитные функции не поддерживаются. Виртуальный режим процессора предназначен для выполнения в защищенном режиме программ, предназначенных для работы в реальном режиме.

Адресация оперативной памяти

Программа, выполняемая на процессоре i386, обращаясь к фрагменту оперативной памяти, должна указать адрес этого фрагмента. Этот адрес складывается из двух составляющих – *селектора* и *смещения*.

Селектор представляет собой идентификатор сегмента, в котором располагается требуемый фрагмент памяти, а смещение определяет порядковый номер первого байта фрагмента в этом сегменте.

При обращении к оперативной памяти необходимый селектор должен быть заранее загружен в один из *сегментных регистров* процессора – **cs**, **ds**, **es**, **fs**, **gs** или **ss**. Для адресации данным может использоваться любой сегментный регистр, для адресации кода – только регистр **cs**, для адресации стека – только **ss**. Каждый сегментный регистр имеет размер 16 бит.

С каждым сегментным регистром связан один дескрипторный регистр.

При загрузке селектора в сегментный регистр в соответствующий дескрипторный регистр автоматически загружается *дескриптор сегмента*.

Дескриптор сегмента содержит:

- линейный адрес начала сегмента в оперативной памяти (при отключенной страничной адресации этот адрес совпадает с линейным);
- длину сегмента;
- ряд атрибутов сегмента.

Сегменты оперативной памяти, которым соответствуют различные дескрипторы, могут пересекаться и даже совпадать. Дескриптор сегмента может быть *глобальным* или *локальным*.

Глобальные дескрипторы хранятся в специальном сегменте называемом *таблицей глобальных дескрипторов*.

Локальный дескриптор (в отличие от глобального) доступен только одной задаче. Каждая задача выполняемая в системе имеет *таблицу локальных дескрипторов*.

Уровень привилегированности

Защита оперативной памяти в процессоре i386 основана на понятии уровня привилегированности.

Уровень привилегированности (privilege level, PL) – это числовой идентификатор, принимающий

значения от 0 до 3, который определяет возможности задачи выполнять команды процессора, модифицировать регистры и области оперативной памяти.

Чем меньше числовое значение уровня привилегированности, тем выше этот уровень и тем более полный доступ имеет задача к аппаратным возможностям процессора.

Кольцо защиты – множество задач, обладающих некоторым конкретным уровнем привилегированности. Например, если код программы имеет уровень привилегированности, равный нулю, то говорят, что программа выполняется в нулевом кольце защиты (или просто в нулевом кольце).

Обычно прикладные программы выполняются в третьем кольце защиты, а код ОС – в нулевом кольце защиты.

Уровень привилегированности задачи называют текущим уровнем привилегированности.

Каждый дескриптор и каждый селектор также имеют свои уровни привилегированности.

Защита сегментов оперативной памяти

Защита сегментов ОП в процессоре i386 основана на сравнении уровня привилегированности задачи, уровня привилегированности дескриптора и уровня привилегированности селектора в момент загрузки селектора в сегментный регистр.

При загрузке селектора в сегментный регистр выполняются следующие проверки:

1. Проверяется корректность селектора.

2. Проверяется совместимость типа загружаемого селектора с типом сегментного регистра.

3. Проверяется достаточность текущего уровня привилегированности задачи для загрузки сегмента.

Таким образом, низкопривилегированная задача не может обращаться к высокопривилегированным сегментам ни при каких обстоятельствах. Высокопривилегированная задача может понизить свой уровень привилегированности в отношении конкретного сегмента памяти, загрузив в сегментный регистр низкопривилегированный селектор.

Расширения команд процессоров AES-NI и NX-Bit (ND-Bit). Модуль доверенной платформы Trusted Platform Module (TPM).

https://ru.wikipedia.org/wiki/

Расширение системы команд AES (Advanced Encryption Standard) — расширение системы команд x86 для микропроцессоров, предложенное компанией Intel в марте 2008[1]. Целью данного расширения является ускорение приложений, использующих шифрование по алгоритму AES. Сходное расширение PadLock engine существует в микропроцессорах от VIA Technologies.

Инструкция	Описание
AES Encrypt Round (AESENC)	Выполнить один раунд шифрования AES
AES Encrypt Last Round (AESENCLAST)	Выполнить последний раунд шифрования AES
AES Decrypt Round	Выполнить один раунд расшифрования AES

(AESDEC)	
AES Decrypt Last Round (AESDECLAST)	Выполнить последний раунд расшифрования AES
AES Key Generation Assist (AESKEYGENASSIST)	Поспособствовать генерации раундового ключа AES
AES Inverse Mix Columns (AESIMC)	Inverse Mix Columns

Intel SHA (*Secure Hash Algorithm extensions*) - набор инструкций процессора, разработанных компанией Intel для ускорения работы приложений, используемых алгоритмы шифрования SHA. Включает 7 инструкций, 4 из которых ускоряют работу SHA-1, остальные 3 - SHA-256. Ускорение может составлять 150-200 % и более (в зависимости конкретного приложения).

2017:

Улучшения безопасности. Intel представила расширения Intel Memory Protection Extensions (MPE), призванные повысить уровень защиты программного обеспечения от вредоносных атак с использованием переполнения буфера. Среди других улучшений безопасности в архитектуре набора команд x86 можно выделить расширение Intel Safer Mode Extensions (SMX), которое создано с целью обеспечить максимально полный контроль над запуском системного программного обеспечения, которое создает защищенное окружение для самого себя и для любого дополнительного ПО, которое может быть запущено в этом окружении. Новейшее достижение в области безопасности, расширение Intel Software Guard Extensions (SGX) представляет собой инструмент для защиты конкретного кода и данных от раскрытия или модификации за счет использования защищенных областей памяти для их выполнения.

NX-Bit

Атрибут (бит) NX-Bit (англ. no execute bit в терминологии фирмы AMD) или XD-Bit (англ. execute disable bit в терминологии фирмы Intel) — бит запрета исполнения, добавленный в страницы (см. таблицы страниц (англ.)) для реализации возможности предотвращения выполнения данных как кода. Используется для предотвращения уязвимости типа «переполнение буфера», позволяющей выполнять произвольный код на атакуемой системе локально или удалённо. Технология требует программной поддержки (см. DEP) со стороны ядра операционной системы.

Технология NX-bit может работать только при соблюдении следующих условий:

- наличие поддержки NX-bit со стороны процессора. NX-bit поддерживают процессоры фирмы Intel, начиная с Pentium 4 серии 6xx, и процессоры фирмы AMD, начиная с Athlon 64;
- наличие поддержки NX-bit со стороны операционной системы. NX-bit поддерживают ОС Linux, начиная с ядра версии 2.3.23, и ОС Windows, начиная с Windows XP SP2;
- использование PAE для процессоров архитектуры x86 или использование процессоров архитектуры x86-64 (бит запрета исполнения доступен в таблице страниц).

Некоторое ПО несовместимо с технологией NX-bit, поэтому BIOS предоставляет возможность отключения технологии.

Trusted Platform Module

https://docs.microsoft.com/ru-ru/windows/device-security/tpm/trusted-platform-module-overview

Технология доверенных платформенных модулей (TPM) предназначена для предоставления аппаратных функций, связанных с безопасностью. Микросхема TPM — это надежный криптографический процессор, спроектированный для выполнения операций шифрования. Микросхема содержит несколько механизмов физической защиты, чтобы предотвратить взлом, и вредоносные программы не могут обойти функции безопасности TPM. Некоторые из основных преимуществ использования технологии TPM:

- создание, сохранение и ограничение использования криптографических ключей;
- технологию TPM можно использовать для проверки подлинности устройства с помощью уникального RSA-ключа TPM, записанного в модуль;
- обеспечение целостности платформы за счет хранения измерений безопасности.

Самые распространенные функции TPM используются для оценки целостности системы, а также для создания и применения ключей. Во время загрузки системы загружаемый загрузочный код (в том числе встроенное ПО и компоненты операционной системы) можно проверить и записать в модуль TPM. Оценку целостности можно использовать для проверки запуска системы и подтверждения того, что ключ на основе TPM использовался, только когда система была загружена с правильным программным обеспечением.

Ключи на основе TPM можно настраивать различными способами. Например, можно сделать ключ на основе TPM недоступным за пределами модуля. Это удобно для защиты от фишинга, так как в этом случае ключ не может быть скопирован и использован без TPM. Ключи на основе TPM также можно настроить для ввода значения авторизации. Если число неудачных попыток авторизации слишком велико, TPM активирует свою логику защиты от атак перебором по словарю и предотвращает дальнейшие попытки подбора.