Прзентация на тему

Методы защиты от атаки типа переполнение буфера

Замула Е. С.

04 апреля 2024 года

Российский университет дружбы народов, Москва, Россия

Информация

Докладчик

:::::::::::::: {.columns align=center} ::: {.column width="70%"}

- Замула Егор Сергеевич
- Стдуент 1-ого курса
- Российский университет дружбы народов
- [113220796@pfur.ru]
- https://github.com/egorzam21/study_2023-2024_os-intro

::: ::: {.column width="30%"}

Введение

Введение

Переполнение буфера является одной из наиболее распространенных уязвимостей в программном обеспечении, которая может быть использована злоумышленниками для выполнения вредоносного кода или получения несанкционированного доступа к системе. В данном плане лекции мы рассмотрим суть переполнения буфера, его причины и последствия, а также методы защиты от этой уязвимости.

Что такое переполнение буфера

Переполнение буфера

Переполнение буфера – это тип атаки на программное обеспечение, при котором данные, превышающие размер выделенного буфера, записываются в память, что может привести к нарушению работы программы или даже к возможности выполнения вредоносного кода.

Буфер

Буфер – это область памяти, используемая для временного хранения данных. Он имеет фиксированный размер, который определяется программой. Когда данные записываются в буфер, они должны быть не больше его размера. Однако, если программист не предусмотрел проверку размера данных, то при записи в буфер может произойти переполнение.

Для чего?

Переполнение буфера может быть использовано злоумышленниками для выполнения различных атак, таких как внедрение вредоносного кода, перезапись важных данных или даже получение удаленного доступа к системе.

Причины переполнения буфера

Причины переполнения буфера

Переполнение буфера может произойти по нескольким причинам:

Неправильная проверка размера данных

Одной из основных причин переполнения буфера является неправильная проверка размера данных, которые записываются в буфер. Если программист не учитывает размер данных и не проверяет, что они не превышают размер буфера, то при записи в буфер может произойти переполнение.

Некорректная обработка пользовательского ввода

Еще одной причиной переполнения буфера является некорректная обработка пользовательского ввода. Если программа не проверяет входные данные на соответствие ожидаемому формату или не ограничивает их размер, то злоумышленник может ввести данные большего размера, чем ожидается, и вызвать переполнение буфера.

Ошибки в алгоритмах обработки данных

Некорректные алгоритмы обработки данных могут также привести к переполнению буфера. Например, если программа неправильно вычисляет размер данных или неправильно управляет указателями на буфер, то может произойти переполнение.

Последствия переполнения буфера

Последствия переполнения буфера

Переполнение буфера является серьезной уязвимостью, которая может иметь различные последствия для безопасности системы. Вот некоторые из них:

Выполнение вредоносного кода

Одним из наиболее опасных последствий переполнения буфера является возможность выполнения вредоносного кода. Если злоумышленник может переполнить буфер и записать в него свой вредоносный код, то он может получить полный контроль над системой. Это может привести к краже данных, удалению файлов, установке вредоносного ПО и другим негативным последствиям.

Крах программы

Переполнение буфера может привести к краху программы. Если буфер переполняется, то данные могут быть записаны в область памяти, которая не предназначена для хранения этих данных. Это может привести к некорректной работе программы и ее аварийному завершению. Крах программы может привести к потере данных и недоступности сервисов.

Утечка конфиденциальных данных

Переполнение буфера может привести к утечке конфиденциальных данных. Если буфер переполняется и записывает данные за его пределами, то эти данные могут быть доступны злоумышленнику. Например, если переполнение происходит в буфере, который содержит пароль или другую конфиденциальную информацию, то злоумышленник может получить доступ к этим данным и использовать их в своих целях.

Методы защиты от переполнения буфера

Методы защиты от переполнения буфера

Переполнение буфера является одной из наиболее распространенных уязвимостей в программном обеспечении. Однако, существуют различные методы и техники, которые могут помочь защитить программы от этой уязвимости. Рассмотрим некоторые из них:

ASLR

В Linux есть некоторые встроенные механизмы, которые предотвращают выполнение потенциально вредоносного кода в случае переполнения буфера в программе. ASLR — рандомизация расположения адресного пространства — это рандомизирует адреса памяти, используемые программой каждый раз, когда она запускается, это становится препятствием на пути злоумышленника, поскольку это мешает им легко сделать рабочий эксплойт. Например, если каждый раз меняется обратный адрес, на который вы хотите перенаправить свою программу, она не может быть жестко закодирована в эксплойте. Это особенность самой ОС. Чтобы отключить ASLR: Это должно быть запущено снова после перезагрузки, если оно не включено в сценарий запуска, который автоматически выполняется при каждой загрузке.

DEP

DEP — предотвращение выполнения данных. Это помечает некоторые уязвимые части в памяти как неисполняемые. Это особенность компилятора. Некоторые директивы можно использовать во время компиляции, чтобы отключить такие функции защиты, поскольку они включены по умолчанию в gcc. Параметр -fno-stack-protector отключает стековую часть (канарейки), а параметр -z execstack делает исполняемыми и кучу, и стек.

Инструменты во время разработки

Во время разработки существуют инструменты, которые могут анализировать исходный код и работающие программы, чтобы попытаться обнаружить опасные конструкции или ошибки переполнения до того, как эти ошибки попадут в готовое программное обеспечение. Например, AddressSantizer (это инструмент программирования с открытым исходным кодом, который обнаруживает ошибки повреждения памяти, такие как переполнение буфера или доступ к висящему указателю), и более старые, такие как Valgrind (это программный инструмент для отладки памяти, обнаружения утечек памяти и профилирования), предлагают такие возможности. Однако эти инструменты требуют активного участия разработчика.

Неисполняемый стек

Неисполняемый стек предотвращает выполнения кода в области памяти стека (или динамической памяти), а также запись исполняемого кода, что может в некоторых случаях предотвратить эксплуатацию переполнения буфера. Таким образом, произвести переполнение буфера путем инъекции в стек исполняемого кода представляется невозможным, если используется неисполняемый стек.

Вывод

Вывод

Это лишь некоторые из методов защиты от переполнения буфера. Важно понимать, что защита от этой уязвимости требует комплексного подхода и комбинации различных методов и техник. Кроме того, необходимо учитывать особенности конкретной системы и программного обеспечения, чтобы выбрать наиболее эффективные меры защиты.