Столыпинский вестник

Научная статья

Original article

УДК 004

СИСТЕМАТИЧЕСКИЙ ОБЗОР: АНАЛИЗ УЯЗВИМОСТЕЙ КОДИРОВАНИЯ НА РАЗНЫХ ЯЗЫКАХ

SYSTEMATIC REVIEW: ANALYSIS OF CODING VULNERABILITIES IN DIFFERENT LANGUAGES

Голубятников Артем Олегович, Студент кафедры защищенных систем связи, Санкт-Петербургский государственный университет телекоммуникаций им. проф. М. А. Бонч-Бруевича, г. Санкт-Петербург artemgolubyatnikov@mail.ru

Golubyatnikov Artem Olegovich, Student of the Department of Secure Communication Systems, St. Petersburg State University of Telecommunications named after Prof. M. A. Bonch-Bruevich, St. Petersburg artemgolubyatnikov@mail.ru

Аннотация: Бум языков программирования в 1950-х годах произвел революцию в понимании и доступе к нашему цифровому миру. Изобретенные тогда языки, в том числе Fortrun , используются до сих пор благодаря своей универсальности и способности поддерживать подавляющее большинство старых частей нашего цифрового мира и приложений. Фортран, или перевод формул , был языком программирования, реализованным IBM, который сократил аппарат кодирования и эффективность синтаксиса языка. Фортран ознаменовал начало новой эры эффективного программирования, сократив в несколько раз количество операторов, необходимых для управления машиной . С

тех пор еще десятки языков вошли в регулярную практику и с годами становились все более разнообразными. Некоторые современные языки включают Python, Java, JavaScript, C, C++ и PHP. Эти языки значительно повысили эффективность, а также имеют широкий спектр применения. Python в основном используется для разработки веб-сайтов/программного обеспечения, анализа данных, автоматизации задач, обработки изображений и приложений графического дизайна. С другой стороны, Java в основном используется как язык программирования на стороне клиента.

Abstract: The boom of coding languages in the 1950s revolutionized how our digital world was construed and accessed. The languages invented then, including Fortran, are still in use today due to their versatility and ability to underpin a large majority of the older portions of our digital world and applications. Fortran, or Formula Translation, was a programming language implemented by IBM that shortened the apparatus of coding and the efficacy of the language syntax. Fortran marked the beginning of a new era of efficient programming by reducing the number of statements needed to operate a machine several-fold. Since then, dozens more languages have come into regular practice and have been increasingly diversified over the years. Some modern languages include Python, Java, JavaScript, C, C++, and PHP. These languages significantly improved efficiency and also have a broad range of uses. Python is mainly used for website/software development, data analysis, task automation, image processing, and graphic design applications. On the other hand, Java is primarily used as a client-side programming language. Expanding the coding languages allowed for increasing accessibility but also opened up applications to pertinent security issues. These security issues have varied by prevalence and language. Previous research has narrowed its focus on individual languages, failing to evaluate the security.

Ключевые слова: CWE (перечень общих уязвимостей) , безопасность данных , уязвимости кодирования

Keywords: CWE (Common Weakness Enumeration), Data Security, Coding Vulnerabilities

1. Введение

В современную эпоху языки кодирования используются для создания веб-сайтов и разработки приложений. Проблемы безопасности в этих приложениях могут вызвать серьезные проблемы во всем мире, в основном из-за организационного использования, от Alphabet до Amazon Web Services (AWS). Поскольку для создания таких веб-сайтов используются разные языки программирования, возникающие проблемы безопасности по своей сути различаются по типу риска и распространенности. Число случаев компрометации данных в США выросло со 157 в 2005 году до 1802 в 2022 году по мере роста популярности программирования [1].

Известный пример уязвимости в коде — ошибка ключа стоимостью 100 000 долларов. Грета Фосбакк пыталась перевести дочери 100 000 долларов через систему онлайн-банкинга. Она ввела номер банковского счета дочери и добавила одну дополнительную цифру в середину. Таким образом, значение было усечено неправильно, и деньги были отправлены не тому человеку. По словам Кая А. Олсена, для обнаружения таких ошибок существовал механизм контрольной суммы, проверяющий количество цифр на банковском счете, но он был эффективен только в 92% случаев [2]. Это привело к ошибке перевода денег в таких случаях, как дело Фосбакка и других.

Хотя вопросы денежных переводов и безопасности вызывают беспокойство, ставки зачастую даже выше, если принять во внимание оборудование, работающее с реактивными материалами, с системами управления, управляемыми алгоритмами языка кодирования. Одним из примеров был Therac-25, аппарат лучевой терапии, управляемый системой управления, используемый при лечении рака. Шесть серьезных аварий в системе из-за сбоев компьютерного языка в предыдущей версии системы и неправильного представления кодов ошибок привели к большой передозировке радиации, полученной пациентами. Более того, ошибки переполнения позволяли программному обеспечению время от времени обходить проверки безопасности, поскольку ошибка была числовой, а не фиксированным значением флага

[3]. Случаи медицинской халатности выявили алгоритмические ошибки, вызванные угрозами безопасности языка. Использование альтернативных языков позволило бы обойти эту угрозу безопасности. Решающий характер обеспечения языковой безопасности можно далее проиллюстрировать на примере множества других отраслей, примером которых является космическая отрасль. Ракета «Ариан-5» также вышла из строя из-за ошибок программного обеспечения и имела катастрофические последствия, взорвавшись в атмосфере через несколько секунд после запуска. И снова повторное использование старого кода, который нужно было оценить на предмет сбоев и угроз безопасности, оказалось проблематичным. Питер Ладкин сказал, что сбой «был вызван «ошибкой операнда» при преобразовании данных в подпрограмме из 64-битного числа с плавающей запятой в 16-битное целое число со знаком. Одно значение было слишком большим для преобразования, что привело к ошибке операнда» [4]. Ладкин продолжает винить в том, что в миссии не удалось использовать язык программирования Ada. Ada не учитывает необходимость явного жесткого кодирования базового преобразования данных низкого уровня в язык более высокого уровня. Использовать Аду было совершенно ненужно, но в результате ее использования вся миссия провалилась.

Разным языкам программирования свойственна разная восприимчивость к рискам безопасности кодирования в виде кибератак. Эти атаки обычно можно разделить на четыре основные категории: проблемы межсайтового скриптинга, SQL-инъекция, внедрение команд и криптографические проблемы. Одним из типов атак является XSS или межсайтовый скриптинг. В ходе этой атаки путем внедрения злоумышленники пытаются внедрить вредоносный код в заслуживающий доверия веб-сайт, чтобы атаковать пользователя. SQL-инъекции аналогичным образом внедряют вредоносный код для просмотра серверных данных, которые не должны быть доступны. Внедрение команд использует уязвимости операционной системы (ОС) путем выполнения произвольных команд для использования проблем проверки ввода. Криптографические проблемы являются наиболее разнообразными из

них. Криптографические атаки можно разделить на три основные цели: проверка личности, конфиденциальность и целостность. Каждый из них обеспечивает безопасность данных для определенных пользователей. В каждом языке существуют разные уязвимости, которые относятся к одной из этих четырех категорий кибератак. 86% приложений, использующих языки сценариев общего назначения, такие как PHP, содержат как минимум одну уязвимость межсайтового скриптинга (XSS) и уязвимы для внедрения команд [5]. Однако существуют различные риски безопасности, связанные с конкретными языками. Известно, что такие языки, как C и C++, имеют более надежную защиту от XSS, в то время как такие языки, как Classic ASP и ColdFusion, имеют более высокую распространенность проблем XSS. Наконец, криптографические проблемы более разнообразны и от них сложнее защититься. В целом, криптографические проблемы стали наиболее распространенным подтипом кибератак.

Таким образом, разные языки предлагают разные уровни защиты от атак. Однако крайне важно углубляться в особенности уязвимостей каждого языка, которые лежат в основе этих данных о кибератаках, и проверять данные о распространении этих кибератак, используя подход «снизу вверх». В этой исследовательской статье мы попытаемся определить следующие исследовательские вопросы: (1) Какие уязвимости/риски безопасности кодирования возникают при использовании разных языков? (2) Какие языки программирования наиболее уязвимы перед наиболее распространенными кибератаками?

2. Методология

Чтобы ответить на цели исследования, в этой статье будет использоваться приведенная ниже методология. Был проведен систематический обзор литературы последних исследований, чтобы обеспечить четкий и всесторонний обзор данных и интерпретаций предыдущей литературы. Была использована одна база данных, Google Scholar. Используемые ключевые слова включали

«риски безопасности», «уязвимости кодирования», «кибератаки», «Python», «JavaScript», «Java», «C», «C++», «PHP» и «риски, присущие языкам кодирования». Каждый документ был проанализирован на предмет рисков безопасности и уязвимостей, связанных с каждым языком программирования.Кроме того, были определены области использования и применения каждого языка программирования.Сравнивая статистику серьезности и частоты уязвимостей кодирования на разных языках, можно определить риск использования каждого языка. могут быть оценены. Использовалась модель PRISMA 2020. Были включены систематические обзоры литературы, тематические исследования и метаанализы. Как показано на рисунке 1 , 37 исследований соответствовали критериям включения. Из них два были Тридцать пять исследований оставалось для проведения нашего обзора литературы.

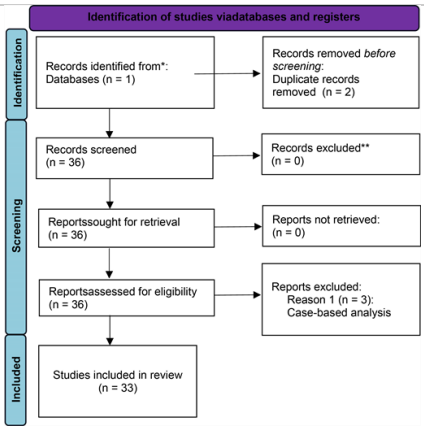


Рисунок 1 . ПРИЗМА.

3. Результаты

3.1. Питон

В последние годы Python пережил взрывной рост популярности. Согласно индексу TIOBE, по состоянию на август 2022 года Python обогнал C и стал самым популярным языком программирования с рейтингом 15,42% и, вероятно, продолжит расти. [2] Его популярность в основном объясняется его простым и легким для чтения синтаксисом.

Поскольку Python легко кодируется и удобен для пользователя, на уровне отдельного аналитика качества гораздо проще обнаружить ошибки и ошибки в программах, которые могут вызвать проблемы с безопасностью. По данным mend.io, только 6% всех зарегистрированных уязвимостей с открытым исходным кодом связаны с веб-сайтами, написанными на Python. Кроме того, в среднем только 15% серьезных уязвимостей Python за последние пять лет считаются высоким риском, что относительно низко по сравнению с другими основными языками программирования [4].

Однако у Python также есть некоторые уязвимости безопасности, на которые стоит обратить внимание. Согласно списку CWE (Common Weakness Enumeration) Python, существует четыре основных уязвимости безопасности. В порядке убывания частоты: проверка ввода (CWE-20), разрешения, привилегии и контроль доступа (CWE-264), межсайтовый скриптинг (XSS) (CWE-79) и утечка/раскрытие информации (CWE-79). 200) влияют на сценарии на основе языка Python [4].

При дальнейшем анализе риски безопасности Python в первую очередь являются результатом недостаточной проверки ввода данных пользователем [3]. Проверка ввода — это то, как программа может гарантировать, что данные, вводимые пользователем, не являются вредоносными, неправильно отформатированы или неправильными другими способами. Программа может избежать утечки важной информации, выполняя очистку ввода пользователя.

Кроме того, Python является типобезопасным языком. Это означает, что он проверяет наличие ошибок типа при компиляции программы, что называется временем компиляции. С другой стороны, языки, которые не являются типобезопасными, проверяют ошибки типов во время выполнения. Это выгодно, поскольку делает Python менее подверженным ошибкам. Языки, которые не являются типобезопасными, такие как C и C++, подвержены ошибкам из-за отсутствия типобезопасности.

3.2. JavaScript

JavaScript является популярным языком, и согласно индексу TIOBE, JavaScript занимает седьмое место в списке самых популярных языков программирования в мире с рейтингом 2,33% по состоянию на август 2022 года [1]. Однако популярный веб-сайт по программированию Stack Overflow в 2021 году опубликовал опрос разработчиков, в котором утверждалось, что JavaScript является самым популярным языком. 64,96% респондентов и 68,62% профессионалов выбрали JavaScript в качестве наиболее любимого и используемого языка разработки [2]. JavaScript лежит в основе почти 95+% веб-сайтов, доступных сегодня, подчеркивая широко распространенный характер и разрушительные последствия кибератак, использующих одну-единственную уязвимость [1].

Несмотря на такую популярность, JavaScript имеет больше уязвимостей, чем Python. По данным mend.io, 11% всех зарегистрированных уязвимостей с открытым исходным кодом связаны с веб-сайтами, использующими JavaScript. Mend.io также утверждает, что в среднем 31% уязвимостей, обнаруженных на веб-сайтах, использующих JavaScript, содержали уязвимости высокого риска за последние пять лет, что более чем вдвое больше, чем у Python.

По данным mend.io, JavaScript уязвим к трем основным CWE: проблемам криптографии, обходу пути и межсайтовому скриптингу (XSS), первые два относительно редки среди других ведущих языков программирования. Кроме того, Майкл Холландер утверждает, что внедрение SQL и раскрытие конфиденциальных файлов cookie являются одними из главных уязвимостей JavaScript [1].

3.3. Джава

Java также чрезвычайно популярен. Согласно индексу TIOBE, который является показателем относительной популярности языков программирования, Java занимает третье место по популярности с 12,40% [4]. Кроме того, Java является типобезопасным языком, что полезно для минимизации ошибок типов. Более того, технология Java используется шестью миллионами

разработчиков на миллиардах устройств [2]. В результате любая значительная уязвимость, обнаруженная в Java, потенциально может привести к тому, что миллиарды устройств подвергнутся риску взлома. Поэтому крайне важно проверить, безопасен ли Java.

По данным mend.io, Java составляет 11% всех уязвимостей безопасности с открытым исходным кодом, что ставит ее в тройку лучших в этой категории. Однако количество веб-сайтов, написанных на Java и содержащих серьезные уязвимости безопасности, составляет в среднем примерно 19%, но с 2015 года оно снижается. Кроме того, Java разделяет четыре лучших CWE Python: утечка/раскрытие информации, проверка входных данных, межсайтовый скриптинг и разрешения. , привилегии и контроль доступа [4] . Java и Python имеют несколько сходств, что объясняет, почему их основные уязвимости безопасности одинаковы.

Однако попадание в число лучших CWE не указывает на критическую уязвимость. Статическое тестирование безопасности приложений (SAST) помогает выявить уязвимости кода. Наиболее распространенной уязвимостью кода, выявленной в ходе этого процесса, была уязвимость непропатченных библиотек [2]. Кроме того, С. Рахаман и др . пришли к выводу, что по совокупной оценке приложений Android эти приложения черпают свои уязвимости в основном из упакованных библиотек кода [3]. Неисправленные библиотеки возникают, когда версии языков программирования обновляются для исправления уязвимостей, обнаруженных в библиотеках. Списки этих уязвимостей можно найти в таких базах данных, как Национальная база данных уязвимостей или база данных CVE. Языки программирования могут быть уязвимы для злоумышленников, которым известны эти уязвимости, если они не обновлены.

3.4. C

C — относительно старый язык, который до сих пор используется. Согласно индексу TIOBE, C занимает второе место по популярности после Python и занимал первое место до августа 2022 года. Однако,

согласно опросу Stack Overflow, только 21,01% респондентов провели обширную работу по разработке с использованием C [6].

Несмотря на широкую применимость, существует три важных компонента необходимости более глубокого знакомства с C. 1) C не очень удобен для пользователя, и его сложно освоить из-за требований к синтаксису и практики жесткого кодирования. 2) C содержит недостатки безопасности и слабые места инфраструктуры, которые легко использовать [14]. 3) C не является типобезопасным, что делает C все более склонным к ошибкам типов. Mend.io утверждает, что C имеет наибольшую уязвимость из вышеперечисленных языков: на него приходится 50% всех зарегистрированных уязвимостей за десятилетие. Однако этот последний аргумент о безопасности типов может быть ненадежным, поскольку стоит упомянуть, что длительность существования C и масса кода, основанного на C, могут неправильно исказить эти данные [4].

Верхним CWE C является Неправильное Ограничение Операций в пределах буфера памяти, набравшее 75,56 баллов из 100. Чтение за пределами границ является вторым по распространенности CWE со счетом 26,5, гораздо менее распространенным, чем предыдущий [5] .

3.5. C++

C++ занимает четвертое место в индексе TIOBE с рейтингом 10,17% [1]. Однако, согласно опросу разработчиков Stack Overflow, 24,31% респондентов провели обширную разработку с использованием C++, что занимает 10-е место в этом опросе. Это указывает на улучшение доступности для пользователей и простоты использования.

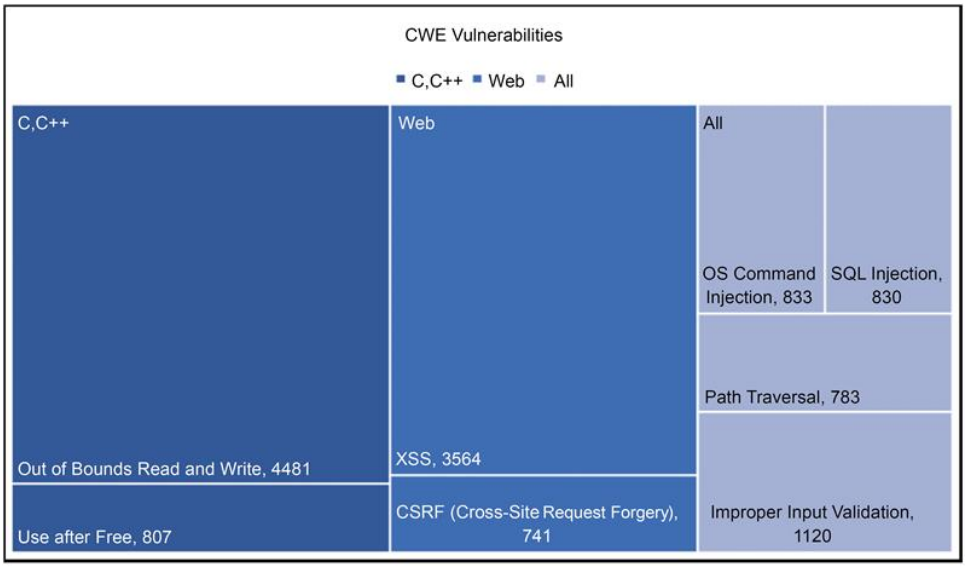
Как и вышеописанное, с точки зрения безопасности C++ не является типобезопасным, поэтому он уязвим для ошибок типов. Примечательно, что C++ имеет два существенных уникальных риска безопасности. Во-первых, C++ имеет самый высокий процент серьезных уязвимостей безопасности в этой группе — 36% [4]. Во-вторых, главным CWE C++ являются ошибки буфера, на которые приходится более 50% всех CWE. По данным mend.io, эти ошибки буфера, называемые кодом CWE 19, невероятно распространены в C. Как объясняется

ошибкой времени выполнения C, теперь становится очевидным, что они не уникальны для C. C++ имеет наблюдается резкий всплеск этих ошибок с 2017 года [4] . Более того, как показано на рисунке 2 , C и C++ несут единственную ответственность за значительную часть уязвимостей CWE [5].

3.6. PHP

Согласно индексу TIOBE, PHP занимает десятое место с рейтингом популярности 1,39% [1]. 21,98% респондентов опроса Stack Overflow 2021 заявили, что они провели обширную разработку с использованием PHP [3]. PHP особенно популярен в разработке веб-сайтов, поскольку его возможности постоянно растут и применяются [2]. В первую очередь это связано с использованием веб-разработчиками PHP для создания динамического контента и использованием надежных систем поддержки баз данных. В целом, это может обеспечить удобство использования [1].

Что касается безопасности, по данным mend.io, за последние пять лет 16% веб-сайтов, написанных на PHP, содержали серьезные уязвимости. В PHP есть один явный главный CWE: межсайтовый скриптинг или XSS. Несмотря на то, что мы подробно изучили распространенность XSS CWE, PHP является единственным языком с заметными уязвимостями внедрения SQL (CWE-89) [4].



Фигура 2 . Количество CWE-уязвимостей за 2021 год по языкам и типам.

4. Дискуссия

Чтобы правильно интерпретировать это исследование, также важно понимать, что каждый из языков, обсуждавшихся ранее, имеет разные цели. Несмотря на свои уязвимости, языки по-прежнему могут использоваться исключительно для выполнения конкретной задачи, которую ни один другой язык не может выполнить лучше.

4.1. Приложения Python

Python имеет относительно небольшое количество уязвимостей, что является положительной особенностью, поскольку это один из наиболее используемых языков программирования. Большинство его приложений связаны с обработкой изображений и графическим дизайном [1]. Кроме того, Python можно использовать при разработке веб-сайтов/программного обеспечения, анализе данных и автоматизации задач. Поскольку Python очень удобен для пользователя, прост в освоении и имеет простой синтаксис, многие бухгалтеры и ученые также могут использовать его в своих рабочих целях [2]. Его часто называют языком, удобным для начинающих веб-разработчиков и студентов, изучающих информатику.

4.2. Приложения JavaScript

JavaScript — это клиентский язык программирования, который в основном используется в веб-разработке, в частности для придания веб-сайтам динамичности и интерактивности [1]. 98,0% всех веб-сайтов используют JavaScript в качестве языка программирования на стороне клиента [2]. Уязвимости JavaScript могут иметь высокие ставки, поскольку один из наиболее популярных языков лежит в основе некоторых веб-страниц с наибольшим трафиком без серьезных сбоев или угроз безопасности. Примеры включают Google, YouTube, Facebook, Wikipedia и Amazon. В результате любая уязвимость, обнаруженная в JavaScript, может быть перекрестно использована на нескольких веб-сайтах, что не только важно, но и может иметь финансовые катастрофические последствия.

4.3. Приложения Java

Java можно использовать для создания приложений на нескольких платформах, особенно в связи с недавним бумом видеоигр [3]. Однако Java не ограничивается мобильными и компьютерными приложениями. Одна из наиболее новых угроз безопасности, возникших в результате использования Java, исходит от Национального управления по аэронавтике и исследованию космического пространства (НАСА). Одним из недавних проектов НАСА стало WorldWind, общедоступное приложение, позволяющее осуществлять мониторинг камер высокого разрешения со спутников практически в любом месте на Земле [2]. В результате уязвимость безопасности в Java может привести к тому, что приложения станут объектом нападения злоумышленников и могут привести к сбою.

4.4. Приложения C

Как установлено, относительно большее количество уязвимостей, специфичных для C, и их распространенность по сравнению с другими языками означает, что любое приложение, основанное на C, имеет повышенный риск безопасности. Приложения языка C столь же фундаментальны для наших компьютеров и лежат в основе большинства операционных систем (ОС). Многие стандартные ОС на базе C включают, помимо прочего, Windows, Linux (почти полностью написанный на C), Google Chrome OS, RIM Blackberry OS 4.x, Symbian OS, Apple Mac OS X, iPad OS, Apple и Cisco IOS. [5] . Рекомендации по безопасности указывают на минимизацию использования C, особенно при разработке веб-сайтов, где риски безопасности постепенно возрастают. Несмотря на свои недостатки, C по-прежнему является распространенным языком программирования.

4.5. Приложения C++

C++ — универсальный язык программирования, используемый во многих проектах, таких как приложения, игры, разработка браузеров и операционных систем [5]. Поскольку C++ используется в таком большом спектре программ, крайне важно гарантировать, что уязвимости кода C++ не вызывают серьезных

проблем с безопасностью. В противном случае одни и те же основные уязвимости могут быть использованы во множестве программ, использующих C++.

4.6. Приложения PHP

PHP — это язык сценариев, который в основном используется для веб-разработки, но его также можно использовать для создания таких проектов, как графические интерфейсы пользователя или графические интерфейсы пользователя [3]. Многие популярные веб-интерфейсы используют PHP, включая Facebook, Wikipedia и Etsy, а также многие другие [5].

4.7. Происхождение уязвимостей безопасности

Многие уязвимости безопасности возникают на ранних стадиях, когда новичков учат программированию, а не тому, как безопасно писать код. Тейлор и др . продемонстрировали, что значительная работа над уязвимостями SQL-инъекций демонстрирует удивительное отсутствие беспокойства или обсуждения превентивных мер безопасности [2]. Отсутствие образования в области профилактических мер приводит к минимизации проблем безопасности и увеличению числа аналитиков, создающих доступные продукты; веб-сайты, приложения и другие программы неизбежно содержат уязвимости. Хакеры часто могут легко воспользоваться этими уязвимостями, что приводит к утечке данных и серьезным финансовым последствиям в отрасли.

Более того, в последние годы, начиная с 2017 года, мы наблюдаем резкий рост числа уязвимостей, возможно, из-за резкого роста проблем безопасности и применимости C++ [3]. Это подчеркивает важность методов безопасного кодирования, поскольку большее количество уязвимостей означает больше точек доступа для атак злоумышленников.

4.8. Последствия исследования

Изучая риск использования популярных языков программирования в повседневных компьютерных программах и на веб-сайтах, в этом исследовании описываются риски, связанные с использованием каждого языка. Кроме того, определяя различные варианты использования каждого языка, этот документ

может помочь программистам взвесить преимущества и недостатки использования конкретных языков для определенных задач.

5. Кибератаки по типам уязвимостей

В этом разделе анализируются данные и тенденции, чтобы определить, насколько безопасны различные языки программирования, исследуя, насколько хорошо каждый язык реагирует на основные типы кибератак.

5.1. Межсайтовый скриптинг (XSS)

Чтобы отыграться, межсайтовый скриптинг, или XSS, — это тип кибератаки, которая происходит, когда злоумышленники внедряют вредоносный код в заслуживающий доверия веб-сайт, чтобы атаковать ничего не подозревающего пользователя. Уязвимости, подверженные XSS, встречаются очень часто; По данным исследования Virginia Journal of Science, более 60% веб-приложений уязвимы для XSS-атак [1].

XSS обычно входит в число наиболее уязвимых мест ведущих языков программирования. Согласно отчету Mend.io WhiteSource, PHP, Javascript, Java и Python содержат XSS как одну из трех наиболее распространенных уязвимостей безопасности [4]. Эта тенденция иллюстрирует угрозу XSS для нескольких веб-сайтов, поскольку PHP и Javascript повсеместно используются при разработке веб-сайтов.

5.2. Проверка ввода

Проверка ввода — это процесс, который включает в себя проверку ввода пользователя, чтобы убедиться, что он соответствует набору стандартов. Например, поле ввода, запрашивающее адрес электронной почты от пользователя, может проверить ввод пользователя, проверив наличие символа «@» и других функций. Однако иногда этот процесс можно обойти. Злоумышленники также могут получить доступ к частной информации. Один из сценариев, в котором это может произойти, — это когда человек заказывает товар с веб-сайта, но вводит команду SQL, которая извлекает номера кредитных карт разных людей, которые ранее делали заказ.

Проверка ввода широко распространена среди ведущих языков программирования. Согласно отчету Mend’s WhiteSource, CWE для проверки ввода занимает второе место по распространенности в C, Java и C++ и наиболее распространено в Python (по сравнению с другими CWE) [4].

5.3. SQL-инъекция

SQL-инъекция — это стратегия, которую злоумышленники используют для борьбы со слабой проверкой входных данных. Злоумышленники могут ввести на веб-сайт вредоносную строку кода SQL, что позволит им получить доступ к ограниченной информации из баз данных. В простом примере злоумышленники могут ввести «SELECT * FROM Users» в текстовом поле, чтобы получить элементы в таблице «Пользователи».

В отчете WhiteSource компании Mend SQL-инъекция названа второй по распространенности CWE-уязвимостью SQL [4]. Более того, согласно исследованию интернет-компании Akamai, на SQL-инъекции приходится более 65,1% всех атак на веб-приложения [3].

5.4. Запись за пределами границ

Запись за пределами установленного буфера происходит, когда программа записывает данные за пределы установленного буфера, что приводит к сбою или сбою запуска программы. Другое название этого явления — переполнение буфера. Злоумышленники могут воспользоваться переполнением буфера, чтобы намеренно завершить работу программы или заставить ее делать то, что хочет злоумышленник.

По данным веб-сайта CWE, Out-of-bounds Write занимает первое место в списке 25 самых опасных уязвимостей программного обеспечения за 2022 год [3].

6. Выводы

Целью этой статьи было ответить на исследовательские вопросы: (1) Какие уязвимости/риски безопасности кодирования возникают при использовании разных языков? и (2) Какие языки программирования наиболее уязвимы перед наиболее распространенными кибератаками? Чтобы ответить на эти вопросы,

ведущие CWE и другие тенденции для каждого языка проанализировали различные уязвимости, влияющие на популярные языки программирования. C и C++ наиболее уязвимы и наиболее уязвимы для кибератак. Хотя Python, Java, JavaScript и PHP не являются полностью безопасными, они все же относительно безопаснее, чем C и C++.

Чтобы снизить частоту кибератак, будущие пользователи языков программирования должны учитывать общие уязвимости для каждого языка и стараться не повторять такие угрозы безопасности. Это особенно важно для пользователей C и C++, которые могут радикально минимизировать количество уязвимостей в своем коде, добавив меры безопасности, не позволяющие хакерам использовать такие уязвимости.

Литература

1. Красов А. и соавт. Использование методов математического прогнозирования для оценки нагрузки на вычислительные мощности сети IoT // 4-я Международная конференция по будущим сетям и распределенным системам (ICFNDS). – 2020. – С. 1-6.
2. Гельфанд А. М. и др. ИНТЕРНЕТ ВЕЩЕЙ (IoT): УГРОЗЫ БЕЗОПАСНОСТИ И КОНФИДЕНЦИАЛЬНОСТИ //Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2021). – 2021. – С. 215-220.
3. Гельфанд А. М. и др. Исследование распределенного механизма безопасности для устройств интернета вещей с ограниченными ресурсами //Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2020). – 2020. – С. 321-326.
4. Косов Н. А. и др. АНАЛИЗ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ДЕТЕКТИРОВАНИЯ АНОМАЛИЙ В СЕТЕВОМ ТРАФИКЕ //Цифровизация образования: теоретические и прикладные исследования современной науки. – 2021. – С. 33-37.
5. Косов Н. А., Тимофеев Р. С. СРАВНЕНИЕ МЕТОДОВ ОБУЧЕНИЯ СВЁРТОЧНЫХ НЕЙРОННЫХ СЕТЕЙ //Актуальные проблемы

инфотелекоммуникаций в науке и образовании (АПИНО 2021). – 2021. – С. 526-530.

6. Literature

1. Krasov A. et al. Using mathematical forecasting methods to estimate the load on computing power of an IoT network // 4th International Conference on Future Networks and Distributed Systems (ICFNDS). – 2020. – P. 1-6.
2. Gelfand A. M. et al. INTERNET OF THINGS (IoT): THREATS TO SECURITY AND PRIVACY // Current problems of information telecommunications in science and education (APINO 2021). – 2021. – P. 215-220.
3. Gelfand A. M. et al. Study of a distributed security mechanism for Internet of Things devices with limited resources // Current problems of information telecommunications in science and education (APINO 2020). – 2020. – P. 321-326.
4. Kosov N. A. et al. ANALYSIS OF MACHINE LEARNING METHODS FOR DETECTING ANOMALIES IN NETWORK TRAFFIC // Digitalization of education: theoretical and applied research of modern science. – 2021. – pp. 33-37.
5. Kosov N. A., Timofeev R. S. COMPARISON OF TRAINING METHODS OF CONVOLUTIONAL NEURAL NETWORKS // Current problems of information telecommunications in science and education (APINO 2021). – 2021. – P. 526-530.

Для цитирования: Голубятников А.О. Систематический обзор: анализ уязвимостей кодирования на разных языках // Научный сетевой журнал «Столыпинский вестник» №1\2024