Федеральное государственное автономное
образовательное учреждение высшего образования
Национальный исследовательский университет
«Московский физико-технический институт»

Методические рекомендации
по решению конкурсных заданий практического этапа Московского конкурса межпредметных навыков и знаний «Интеллектуальный мегаполис. Потенциал» в номинации «ИТ-класс» по направлению «Информационная безопасность»

г.Москва
2022 г.

Оглавление

1. Тематическая направленность конкурсных материалов 3
2. Теоретическая справка 4
2.1 base64 4
2.2 bash 5
2.3 Утилита file 6
2.4 Утилита iconv 7
2.5 Python 7
2.6 Шифр Цезаря 7
2.7 Модель ISO/OSI 9
2.8 Стек TCP/IP 10
2.9 Протокол IP 11
2.10 Протокол TCP 11
2.11 Протокол UDP 12
2.12 Порт 12
2.13 Сканирование портов 12
2.14 HTTP 13
2.15 URL 14
2.16 HTML 14
2.17 Теги HTML 14
3. Организационные вопросы 16
3.1 Технические требования к аудитории. 16
4. Решение демонстрационного варианта 17
4.1 ЗАДАНИЕ № 1 17
4.2 ЗАДАНИЕ № 2 19
5. Инструкция по использованию автоматической проверяющей системы EJudge 24
6. Литература 29

1. Тематическая направленность конкурсных материалов

№ задания	Уровень сложности	Уникальные кодификаторы Конкурса	Контролируемые требования к проверяемым умениям	Балл
1.	Базовыми считаются №1, № 2 задания; остальные — повышенной сложности	11.1.3 - Аутентификация: определения, свойства, требования, 10.1.4.4 - Классическая криптография: шифр Цезаря, шифр Виженера, шифр Плейфера	Умение пользоваться браузером и знание основных Web-уязвимостей	30
2.	Базовыми считаются №1, №3, №5 задания; остальные — повышенной сложности	10.1.6.1 - Понятие технической защиты информации, 10.1.4.4 - Классическая криптография: шифр Цезаря, шифр Виженера, шифр Плейфера	Умение пользоваться командной строкой Linux	30
			Сумма баллов	**60**

2. Теоретическая справка

2.1 base64

Для передачи данных только печатными символами информация может быть закодирована в различных представлениях, например, в base64. По 3 байта представляются как 24 бита, затем разбиваются на 4 шестёрки бит. Каждая шестёрка кодирует число от 0 до 63, сами числа кодируются символами, как в Таблице 1 ниже.

Таблица 1.

Value	Char	Value	Char	Value	Char	Value	Char
0	A	16	Q	32	g	48	w
1	B	17	R	33	h	49	x
2	C	18	S	34	i	50	y
3	D	19	T	35	j	51	z
4	E	20	U	36	k	52	0
5	F	21	V	37	l	53	1
6	G	22	W	38	m	54	2
7	H	23	X	39	n	55	3
8	I	24	Y	40	o	56	4
9	J	25	Z	41	p	57	5
10	K	26	a	42	q	58	6
11	L	27	b	43	r	59	7
12	M	28	c	44	s	60	8
13	N	29	d	45	t	61	9
14	O	30	e	46	u	62	+
15	P	31	f	47	v	63	/

Пример (Таблица 2):

Таблица 2

Text content	M								a								n							
ASCII	77 (0x4d)								97 (0x61)								110 (0x6e)							
Bit pattern	0	1	0	0	1	1	0	1	0	1	1	0	0	0	0	1	0	1	1	0	1	1	1	0
Index	19								22								5							46
Base64-encoded	T								W								F							u

Для кодирования рекомендуется использовать утилиту base64, для декодирования — её же с опцией -d.

Исходный текст:

Winter is coming

Текст в base64:

V2ludGVyIGlzIGNvbWluZwo=

Кодирование в base64 легко узнать по знакам «=», которые добавляются, чтобы количество символов в итоге делилось на 4. В частности, файлы, которые прикрепляются к электронной почте, кодируются именно base64. Поэтому их размер при пересылке возрастает на треть.

Пример запуска утилиты base64 с использованием перенаправления потоков ввода и вывода через символ «|»:

echo ‘String to encode’ | base64

Результат:

U3RyaW5nIHRvIGVuY29kZQo=

Декодирование:

echo U3RyaW5nIHRvIGVuY29kZQo= | base64 -d

Результат:

String to encode

2.2 bash

Наиболее часто используемый в Linux командный интерпретатор — bash. С его помощью можно сделать много различных операций. В нём часто используется перенаправление потока ввода-вывода через символ «>», который означает, что поток вывода программы нужно направить в файл,

перезаписав его, например, вывод команды ls направим в файл list_of_directory:

```
ls  > list_of_directory
```

Символы «>>» перенаправляют с **до**писыванием файла. Например, нижеприведённая команда при каждом запуске будет **до**писывать текущую дату и время в файл log, не удаляя существующие записи.

```
date >>log
```

Вертикальная черта «|» перенаправляет вывод одной команды через «трубу» (pipe) другой. Например, вывод команды ls -alt, которая выведет все файлы в текущем каталоге с их атрибутами, отсортировав по дате модификации, перенаправим команде grep, которая найдёт все строки, в которых есть подстрока «odt»:

```
ls -alt | grep odt
```

2.3 Утилита file

В частности, можно выяснить, какого типа данные лежат в файле с помощью утилиты file, которая смотрит на содержимое и содержащиеся в нём сигнатуры, а не на имя и его части, которые могут быть легко заменены при переименовании.

Например, переименуем текстовый файл с суффиксом .exe

```
mv file.txt file.exe
```

и посмотрим, какого типа там на самом деле содержимое:

```
file --mime file.exe
```

получим результат такого типа:

```
file.exe: ASCII text
```

Видим, что на самом деле файл текстовый, а не машинный код.

2.4 Утилита iconv

Для превращения текста в одной кодировке в текст в другой используется утилита iconv. Список поддерживаемых кодировок можно узнать с помощью команды

```
iconv -l
```

Перекодировка может осуществляться, например, такой командой:

```
iconv -f cp1251 -t utf8 old.txt > new.txt
```

2.5 Python

Как правило, на Linux-системах установлен интерпретатор языка python. С его помощью, а также благодаря развитой системе модулей можно делать множество различных операций. Например, выполним кодирование и декодирование в base64 с помощью python.

```
>>> import base64
>>> base64.b64encode('String to encode')
```

Вывод:

```
'U3RyaW5nIHRvIGVuY29kZQ=='
>>> base64.b64decode('U3RyaW5nIHRvIGVuY29kZQ==')
```

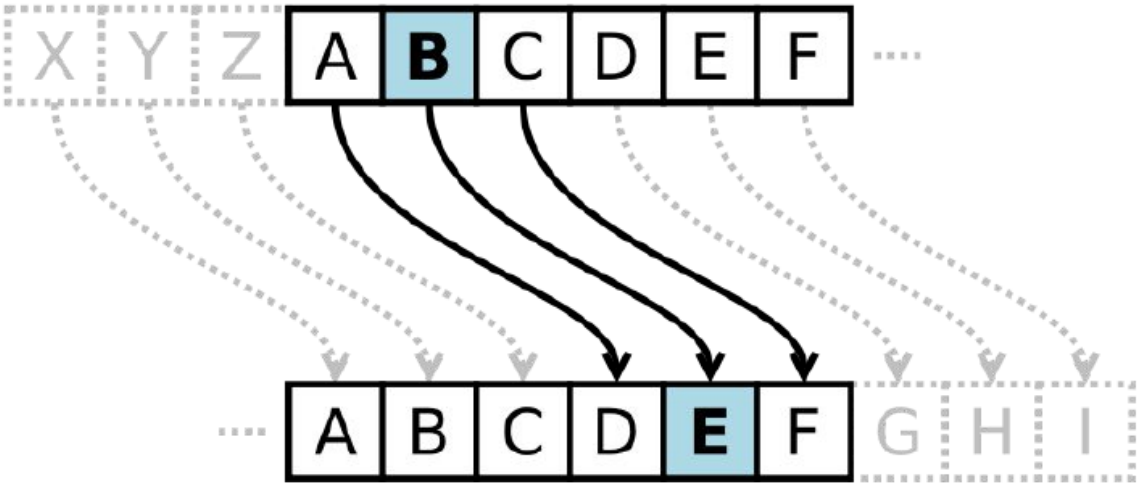
Вывод:

```
'String to encode'
```

2.6 Шифр Цезаря

Один из простейших шифров, известный со времён древнего Рима — шифр Цезаря. В начале определяется алфавит, в котором происходит написание как исходного текста, так и зашифрованного, а затем каждая буква циклически сдвигается на определённое количество символов (см.Рисунок 1).

Рисунок 1.



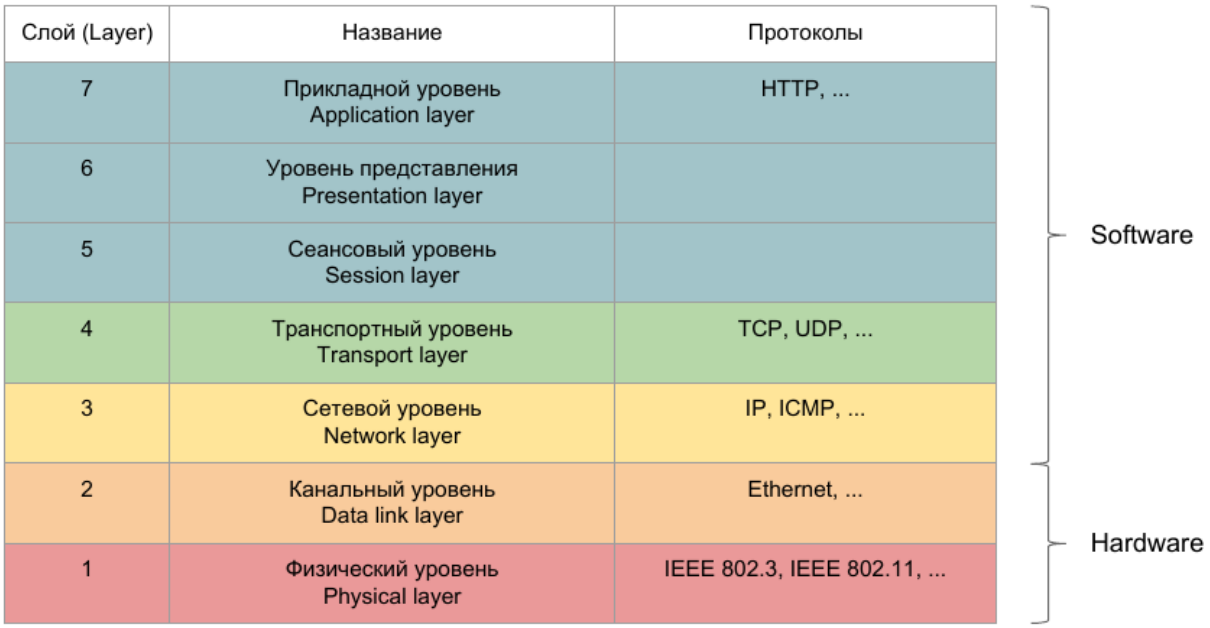
Шифр ненадёжный. Достаточно перебрать все возможные сдвиги, количество которых равно размеру алфавита.

Чтобы не делать перебор, можно воспользоваться частотным анализом. Как правило, все буквы в алфавите равны, но некоторые равнее других и встречаются чаще. Это характерно и для естественных языков, и для языков программирования, и для исполняемых файлов. Если в зашифрованном тексте какая-то другая буква встречается чаще других, значит, она — зашифрованная наиболее часто встречающаяся. По разности их кодов и можно вычислить сдвиг.

2.7 Модель ISO/OSI

Сетевые протоколы разбивают на 7 уровней. Каждый уровень, лежащий ниже, передаёт полезную нагрузку из данных вышележащих протоколов, добавляя свою служебную информацию (см.рисунок 2).

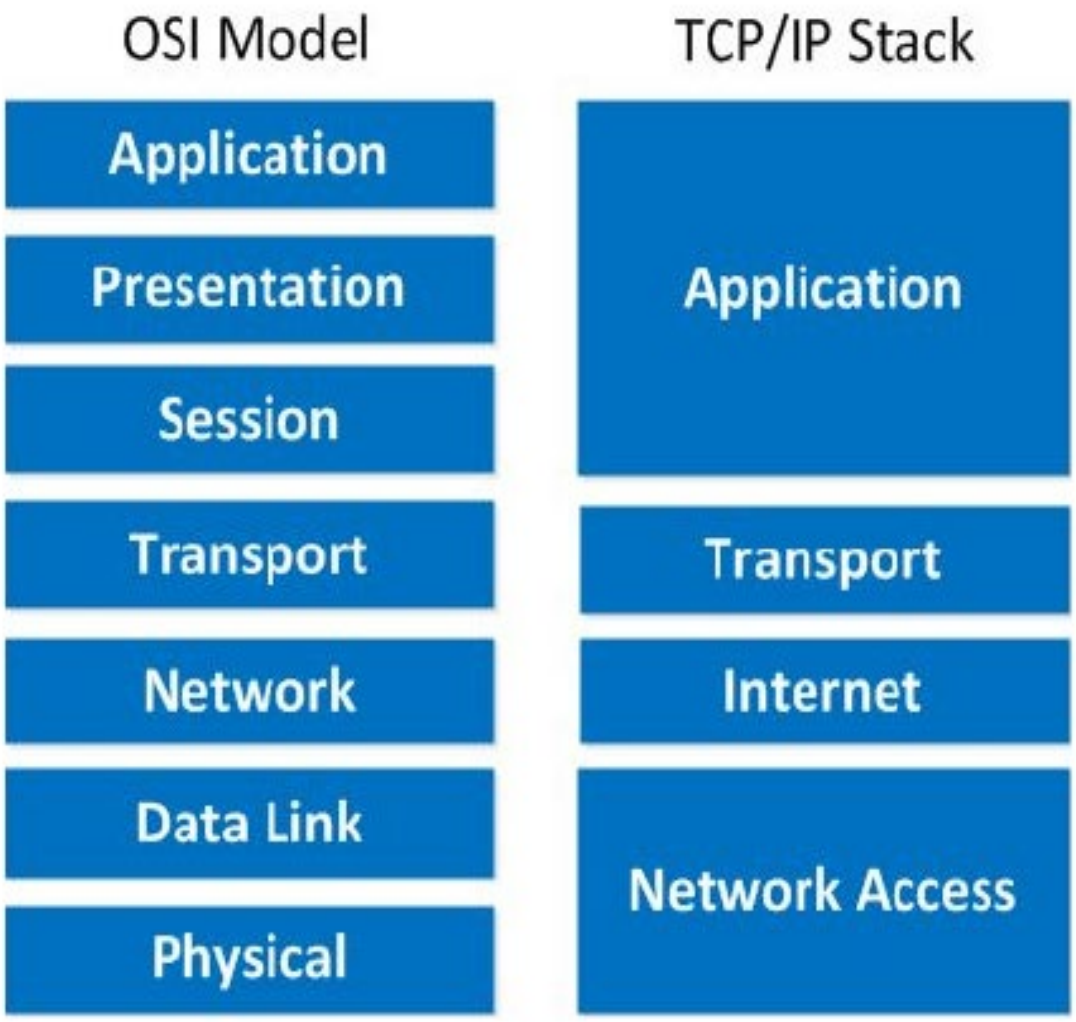
Рисунок 2.



2.8 Стек TCP/IP

На практике 3 верхних (прикладной, сеансовый и уровень представления данных) зачастую реализует прикладная программа, 2 нижних (физический и канальный) обеспечивает аппаратное обеспечение, операционная система обеспечивает транспортный и сетевой. В результате получается такой TCP/IP стек, названный по 2 наиболее используемым протоколам транспортного и сетевого уровней (см.рисунок 3).

Рисунок 3.



2.9 Протокол IP

Internet Protocol - протокол 3 уровня (сетевой уровень). Объединяет сегменты сети в глобальную сеть (Интернет). IP-адрес: уникальный сетевой адрес узла в компьютерной сети.

Версия IPv4

Длина IP-адреса 4 байта

Пример: 192.168.0.3

Версия IPv6

Длина IP-адреса 16 байт

2001:0db8:85a3:0000:0000:8a2e:0370:7334

Для указания, сколько бит кодируют адрес сети, а сколько адрес хоста внутри сети, используют маску. Например, запись:

192.168.0.3/22

показывает, что первые 22 бита — адрес сети, а оставшиеся 10 — адрес компьютера внутри сети.

2.10 Протокол TCP

Transmission Control Protocol (протокол управления передачей). Протокол 4 уровня (транспортный уровень). Работает поверх IP. Устанавливает соединение, передает поток данных, обеспечивает гарантированную доставку.

Используется: Web, SSH, FTP, SMTP, IMAP/POP, …

Протокол подвержен уязвимости Syn Flood, поэтому постепенно заменяется более совершенным протоколом SCTP.

2.11 Протокол UDP

User Datagram Protocol (протокол пользовательских датаграмм). Протокол 4 уровня (транспортный уровень). Работает поверх IP. Не обеспечивает гарантированную доставку. Он быстрее, чем TCP, так как не требуется предварительное «тройное рукопожатие» для установки соединения, а также меньше служебной информации в пакете. Используется: DNS, Media streaming, игры, Tunneling/VPN, …

2.12 Порт

Проблема: IP-адрес один, а программ много. Как отправить данные определенной программе? Для установки соединения или отправки данных протоколы TCP и UDP используют понятие порта. Порт - натуральное число от 1 до 65535. При отправке пакета отправитель указывает порт, на который он хочет передать данные. Начиная слушать порт, получатель указывает, на каком порту он хочет получать данные.

Если же за одним IP-адресом скрыта целая частная сеть, то используется трансляция сетевых адресов (Network Address Translation, NAT). Source NAT - подстановка адреса отправителя, Destination NAT — перенаправление пакета нужному компьютеру во внутренней сети в зависимости от порта.

2.13 Сканирование портов

Для перебора портов, которые открыты на сервере, анализа ответов по традиционным для них протоколам, поиска версий распространённых программ, которые на их слушают и составления предполагаемого списка уязвимостей используют утилиту nmap («Network Mapper»). Эта утилита с открытым исходным кодом для исследования сети и проверки безопасности. Она была разработана для быстрого сканирования больших сетей, хотя прекрасно справляется и с единичными целями:

-определить какие хосты доступны в сети,

-какие широко распространённые службы (название приложения и версию) они предлагают,

-какие операционные системы (и даже версию) они используют,

-какие типы пакетных фильтров/брандмауэров используются.

Nmap обычно используется для проверки безопасности, контроля структуры сети, учета времени работы хоста или службы. Состояние порта может быть одним из: open (открыт), filtered (фильтруется), closed (закрыт) или unfiltered (не фильтруется). «Открыт» означает, что приложение на целевой машине готово для установки соединения/принятия пакетов на этот порт. «Фильтруется» означает, что брандмауэр, фаерволл, сетевой фильтр или какая-то другая помеха в сети блокирует порт, и nmap не может установить открыт этот порт или закрыт. Закрытые порты не связаны ни с каким приложением, так что они могут быть открыты в любой момент. Nmap может предоставлять дополнительную информацию о целях: преобразованные в IP-адрема доменные имена, предположение о используемой операционной системе, типы устройств и MAC-адреса (если находится в том же ethernet-сегменте).

2.14 HTTP

Hyper Text Transfer Protocol — протокол передачи гипертекста (текста с добавлением разметки, ссылок, мультимедиа). Протокол прикладного уровня. Реализован поверх TCP. По умолчанию используется 80 порт (без шифрования) или 443 (с шифрованием, https). Протокол текстовый, на стороне отправителя и получателя можно читать передаваемые пакеты. Используется технология клиент-сервер.

Принцип работы: запрос — ответ. Для сохранения состояния требуются дополнительные инструменты (например, файлы cookie). Используется для общения между браузером и веб-сервером.

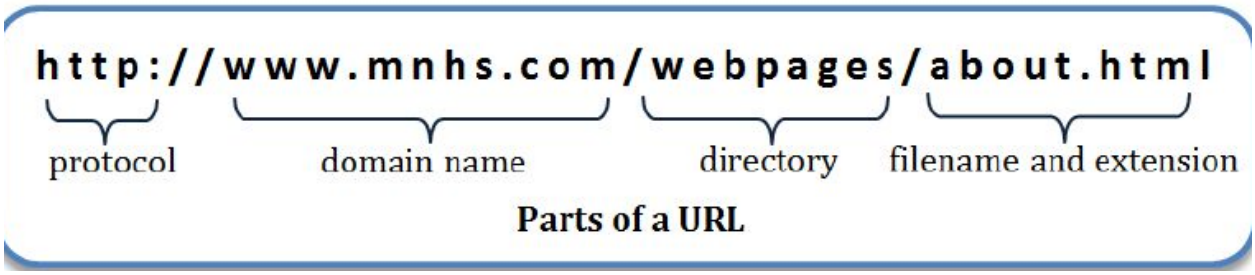
2.15 URL

Universal Resource Locator — универсальный указатель ресурсов — стандарт для указания адреса на web-сайтах.

<схема>://<логин>:<пароль>@<хост>:<порт>/<путь>?<параметры>#<якорь>

(см.рисунок 4).

Рисунок 4.



2.16 HTML

HTML - HyperText Markup Language (язык гипертекстовой разметки). Веб-сервер отдает браузеру описание страницы на языке HTML Браузер парсит HTML и отрисовывает страницу графически.

Взаимодействие web-страницы с пользователем, происходящее на стороне пользователя в его браузере, основывается на HTML, CSS (Cascade Style Sheet) и исполняемом коде на языке JavaScript.

Для подготовки HTML кода, отправляемого пользователю, на стороне сервера может использоваться скрипт на языке PHP (Hypertext PreProcessor).

2.17 Теги HTML

Теги — специальные комбинации символов, обозначающие определённые методы форматирования или вставки дополнительных данных (ссылок, мультимедиа)

Здесь элемент содержит атрибут href, то есть гиперссылку.

Параграф (красная строка)

<table><tr><td></td></tr></table>

< <

> >

& &

<!-- комментарий --!>

<head>Заголовок</head>

3. Организационные вопросы

Индивидуальный вариант участника формируется автоматически во время проведения практического этапа Конкурса предпрофессиональных умений из базы конкурсных заданий.

Индивидуальный вариант участника включает 2 задания в соответствии с демонстрационным вариантом. В заданиях, как правило, есть подзадачи.

На выполнение заданий практического этапа Конкурса отводится 120 минут.

Задание считается выполненным, если ответ участника совпал с эталоном. Задания (или их части) могут быть базовой или повышенной сложности. Для получения максимального балла за теоретический этап Конкурса необходимо дать верные ответы на все задания.

Участники решают задания в аудиториях площадки организатора/организаторов практического этапа. Участникам будет предоставлен доступ в локальную сеть, в которой находится проверяющая система ejudge, а также сайты и серверы с условно секретной информацией (флагами). На компьютерах будет установлена операционная система Linux. Участникам будет известен формат флага, например:

itclass{условно_секретные_цифры_и_буквы}

Организаторы оставляют за собой право потребовать сдать все электронные устройства и контролировать этот факт при помощи технических средств.

3.1 Технические требования к аудитории.

В аудитории должны быть компьютеры по количеству участников и резерв не менее 10% от количества участников и не менее 2 запасных компьютеров. Оперативная память должна быть не менее 8 Гб. На каждом из них должна быть операционная система семейства Linux, командные интерпретаторы bash, python 3, утилиты nc, nmap, midnight commander, а также

браузеры: хотя бы один из Firefox или Chromium. Участникам потребуется доступ в интернет к сети 93.175.29.0/24. Подключение к ним должно быть доступно со всех машин участников без требования авторизации по SMS или звонкам. Рекомендуется использовать десктопные компьютеры с полноразмерной клавиатурой. Экраны должны быть расположены так, чтобы исключить возможность чтения одним участником того, что происходит на экране другого. Условия задач для участников не печатаются, а доступны в электронном виде.

На площадке во время проведения должен присутствовать технический специалист, поддерживающий компьютерные аудитории с оплатой сверхурочной работы в соответствии с трудовым кодексом. Следует заранее предупредить отдел главного энергетика о проведении соревнования, чтобы на этот день не планировались отключения и ремонтные работы.

Рекомендуется организовать питание и доступ к питьевой воде для участников. При этом место должно быть место для приёма пищи и напитков, чтобы исключить залитие компьютеров. Электропитание устройств подготовки воды (кулеров, бойлеров, кипятильников, кофе-машин и т. п.) должно осуществляться не от того же автоматического выключателя, устройства защитного отключения или дифференциального автомата, что питание компьютеров во избежание отключения их по перегрузке. Для предлагаемых участникам блюд и напитков должен быть указан состав во избежание проявления аллергии.

Рекомендуется дежурство медицинского работника. Полезно установить устройство обеззараживания воздуха (рециркулятор), пригодное для работы с людьми в помещении. Участникам должен быть указан маршрут к санузлам, местам установки санитайзеров с антисептиком, эвакуационным выходам.

4. Решение демонстрационного варианта

4.1 ЗАДАНИЕ № 1

Задание состоит в нахождении флагов (условно секретной информации вида itclass{…}, где между фигурными скобками стоят цифры, маленькие и большие латинские буквы, а также спецсимволы) .

Найдите на сайте: https://judge.mipt.ru/itclass спрятанную информацию.

Информация о решении задачи

Критерии оценивания:

1	https://judge.mipt.ru/itclass_colour	3 балла
2	https://judge.mipt.ru/itclass_source	3 балла
3	https://judge.mipt.ru/itclass_robots	6 баллов
4	https://judge.mipt.ru/itclass_directory	6 баллов
5	Расшифруйте https://judge.mipt.ru/itclass_base	6 баллов
6	Расшифруйте https://judge.mipt.ru/itclass_rome	6 баллов

Каждая неудачная попытка -1 балл.

Ответы:

1	https://judge.mipt.ru/itclass_colour	itclass{flag}
2	https://judge.mipt.ru/itclass_source	itclass{captured}
3	https://judge.mipt.ru/itclass_robots	itclass{utenok_nashel}
4	https://judge.mipt.ru/itclass_directory	itclass{Palantir}
5	Расшифруйте https://judge.mipt.ru/itclass_base	itclass{base_emit-ter_collector}
6	Расшифруйте https://judge.mipt.ru/itclass_rome	itclass{Cesar}

4.1.1 Цвет

На web-страничке текст цвета и фона совпадает. Достаточно выделить текст мышью, чтобы выделенный текст стал другого цвета и отличался от фона.

4.1.2. Исходный текст страницы

Для успешного решения задачи достаточно посмотреть исходный текст страницы и увидеть комментарий в HTML. Во многих браузерах это делается сочетанием клавиш Control + U или из контекстного меню (вызывается по нажатию правой кнопки мыши).

4.1.3. robots.txt

В демонстрационном варианте олимпиады учащимся предложено найти то, что не найдут поисковые роботы. Именно в этом и заключается подсказка. В файле robots.txt в корне сайта по стандарту указано, какие страницы поисковым роботам индексировать, а какие запрещены им для посещения (Disallow). В них и нужно искать. Возможен вариант задания с файлом sitemap.xml, который тоже предназначен для поисковых роботов (краулеров).

4.1.4. Каталог

Это задание демонстрирует, что будет, если не запрещать доступ к каталогу, в котором лежат файлы. В таком случае можно удалить из адресной строки имя файла (оставив протокол, адрес сервера, порт и каталог) и увидеть, какие ещё есть файлы в том каталоге. Ответом является одно из указанных имён файла с учётом требований в конкретном варианте задачи (как правило, именно в этом каталоге, а не в его подкаталогах).

4.1.5. base64

Эта задача требует рассмотрения данных и обнаружения характерных признаков кодирования в base64: все данные только буквами и цифрами, а в конце знаки «=» (заполнение). После этого данные легко раскодировать с помощью команды base64 -d. Рекомендуется использовать командную строку

Linux. На тренировках участнику может быть доступен интернет и онлайн-преобразователи, но на экзаменах и олимпиадах, как правило, выход в интернет ограничен, так что нужно уметь пользоваться локальными средствами.

4.1.6. Римский шифр

Это задание может вызвать незначительные трудности. Во-первых, нужно догадаться, каком именно шифр применён. Наиболее известен из шифров времён Римской империи шифр Цезаря. Во-вторых, подбором сдвига нужно добиться, чтобы расшифрованный текст начинался именно с тех букв, которые в стандартном формате флага, описанном для всех заданий. Тогда можно будет декодировать и внутренние данные с помощью того же шифра.

Для автоматизации поиска решения можно написать программу на любом доступном языке.

4.2 ЗАДАНИЕ № 2

На виртуальной машине необходимо выполнить следующие шаги:

1. Зайдите на неё по ssh. Логин itclass, пароль itcla55_CTF, порт 50506, сервер remote.vdi.mipt.ru

В каком каталоге Вы оказались, зайдя по ssh?

2. В каталоге лежит файл под названием flag. Сдайте его содержимое
3. В каталоге лежит файл под названием «-». Сдайте его содержимое.
4. В каталоге лежит файл под названием «Spaces in the file name». Сдайте его содержимое.
5. В каталоге humanreadable лежит только один файл, предназначенный для чтения человеком. Сдайте его содержимое.
6. Найдите файл, который доступен для чтения, неисполняемый и размером 25 байтов. Сдайте его содержимое.

7. Какому пользователю принадлежит файл user?

Критерии оценивания:

1	Сдан каталог	4 балла
2	Сдан файл flag	4 балла
3	Сдан файл «-»	6 балла
4	Сдан файл «Spaces in the file name»	4 балла
5	В каталоге humanreadable лежит только один файл, предназначенный для чтения человеком. Сдайте его содержимое	3 балла
6	Найдите файл, который доступен для чтения, неисполняемый и размером 25 байтов. Сдайте его содержимое.	4 балла
7	Сдан пользователь	5 баллов

Каждая неудачная попытка -1 балл.

Ответы

1	Сдан каталог	/home/itclass
2	Сдан файл flag	itclass{captured_flag}
3	Сдан файл «-»	itclass{captured-flag}
4	Сдан файл «Spaces in the file name»	itclass{Moon, Mars and all space}
5	В каталоге humanreadable лежит только один файл, предназначенный для чтения человеком. Сдайте его содержимое	itclass{Slava_Robotam}

6	Найдите файл, который доступен для чтения, не-исполняемый и размером 25 байтов. Сдайте его содержимое.	itclass{24bytesreadable}
7	Сдан пользователь	itclass{user}

4.2.1. ssh

Необходимо зайти на удалённую машину по протоколу SSH с данным в условии логином, адресом сервера и портом. Далее следует выполнить команду pwd, чтобы вывести текущий каталог.

Команда для входа по ssh может иметь такой вид:

ssh -p 12345 user@server.domain.zone

(вместо 12345 подставьте нужный порт, вместо user — логин из условия, а также укажите нужный сервер вместо server.domain.zone)

При первом подключении Вас спросят, доверяете ли Вы ключу сервера. Как правило, достаточно ответить «yes». Далее Вас попросят ввести пароль. Не удивляйтесь, если при вводе пароля он никак не отображается на экране (даже звёздочками).

Вместо пароля на рабочих местах участников может быть заранее разложен RSA ключ. Приватная часть на машине участника, а публичная на сервере в файле authorized_keys. В таком случае может быть как беспарольный вход, так и необходимость ввести парольную фразу от ключа (получается из условия задачи).

4.2.2. flag

Зайдя по ssh (см. предыдущую задачу), распечатываем файл flag, например, с помощью команды cat.

4.2.3. Файл -

Файл с именем «-» не получится распечатать с помощью команды cat, так как знак «-» обрабатывается как часть опций командной строки. Можно воспользоваться любой командой, которая передаёт имя файла в кавычках.

Пример:

dd if="-"

4.2.3. Пробелы в имени файла

Файл с пробелами в имени так же не получится распечатать с помощью cat, так как слова, разделённые пробелами, будут восприниматься как разные имена файлов. Возможное решение — экранировать пробелы обратным слешом.

Пример:

cat Spaces\ in\ file\ name

Альтернативное решение — воспользоваться файловым менеджером, например, Midnight Commander.

4.2.4. Файл для человека.

В указанном в задаче каталоге есть много файлов, но часть из них — специальные (каталоги, символические ссылки, файлы устройств и т. п.), но есть и обычные файлы. Формат файла можно узнать командой file. Только один из них — обычный текстовый, который и надо сдать как ответ. Если же похожих файлов несколько, следует вспомнить формат флага, а именно, на что должен быть похож правильный ответ.

4.2.5. Файл определённого размера

Следует запустить команду ls -al, чтобы посмотреть атрибуты файлов и выбрать тот, который соответствует условию задачи.

Более продвинутый вариант — воспользоваться утилитой find

find /home/user/ -type f -perm "700"

Ищем файл (f) в каталоге /home/user с правами на чтение, запись и исполнение для владельца (700).

Не рекомендуется запускать команду find на корневом каталоге. Такой глобальный поиск может занять большое время. Кроме того, во многие каталоги пользователя просто не пустят и нужные данные будет потерять среди сообщений об ошибках.

4.2.6. Пользователь

Пользователя, которому принадлежит файл, можно узнать командой ls -al имя_файла. В зависимости от варианта задания может требоваться вывести пользователя или группу.

5. Инструкция по использованию автоматической проверяющей системы EJudge

5.1. Ознакомиться с документацией по автоматической проверяющей системе EJudge можно на официальном сайте https://ejudge.ru.

5.2. Процедура регистрации в системе

5.2.1. ***Создание учётной записи (аккаунта):***

5.2.1.1. Перейдите на страницу входа в контест.
5.2.1.2. Если есть ссылка «Регистрация», перейдите по ней.
5.2.1.3. Перейдите по ссылке «Создать учетную запись».
5.2.1.4. Придумайте/подберите ещё не занятый никем логин, введите его в соответствующее поле.
5.2.1.5. Введите адрес Вашего e-mail.
5.2.1.6. В системе есть контесты с полной и упрощенной процедурой регистрации. При упрощенной процедуре не нужно подтверждать свою регистрацию через ссылку, пришедшую на указанный e-mail, т.к. в этом случае письмо вовсе не высылается, так что можно указать любой адрес.
5.2.1.7. Нажмите на кнопку «Создать учётную запись».
5.2.1.8. Система сгенерирует для Вас пароль и сообщит его на следующей странице. Либо запишите его, либо сразу же после входа в систему измените на такой, который не забудете (по ссылке «Настройки» (первая синяя полоса)).
5.2.1.9. Нажмите на кнопку «Войти».

5.2.2. ***Регистрация своего аккаунта в данном контесте:***

5.2.2.1. Если Вы находитесь на странице «Вход пользователя», перейдите по ссылке «Регистрация», введите свой логин и пароль и нажмите на кнопку «Войти».
5.2.2.2. Нажмите на ссылку НЕ ЗАРЕГИСТРИРОВАН [Подтвердить регистрацию].
5.2.2.3. На открывшейся странице заполните регистрационную информацию о себе (ФИО и т.д.). Для этого нажмите на ссылку «Общая информация» [Редактировать] и заполните необходимые поля. Нажмите кнопку «Сохранить».
5.2.2.4. Когда регистрация будет завершена и контест начнётся, на странице «Просмотр общей информации» появится ссылка «Участвовать» (вторая синяя полоса). Нажмите на неё и приступайте к решению задач.

5.2.3. В тренировочных мероприятиях регистрация может быть свободной, а на практическом этапе конкурса логин и пароль необходимо получить у преподавателя или дежурного по аудитории.

Для участия в практической части конкурса:

- получите у преподавателя логин и пароль;
- откройте страницу по адресу, указанному преподавателем, введите Ваш логин и пароль.
- переходите к пункту 2 данной инструкции, п. 5.2.2. *Регистрация своего аккаунта в данном контесте*.

Для участия в подготовительных мероприятиях со свободной регистрацией в контесте (работа из дома, пробные контесты и т.д.):

- если у Вас уже есть аккаунт, то переходите к пункту 2 данной инструкции, подраздел 5.2.2. «*Регистрация своего аккаунта в данном контесте»*
- если у Вас нет аккаунта или Вы хотите завести другой, то переходите к п.2 данному инструкции, подраздел 5.2.1. *«Создание учетной записи (аккаунта»*

Изменение данных о себе (в том числе фамилии и имени):

- нажмите на ссылку «Выйти из системы» [%username%].
- на странице «Вход пользователя» нажмите ссылку «Регистрация».
- войдите в систему, введя свои логин и пароль.
- отредактируйте информацию о себе.

5.3. Процедура решения

5.3.1. Нажмите на первую (верхнюю) задачу в закладках справа (если не понятно, куда необходимо нажать - листайте вниз и смотрите на картинку).

5.3.2. Внимательно прочитайте условие задачи. Посмотрите примеры того, что вашей программе будет подаваться на вход и что ожидается на выходе.

5.3.3. Если для решения нужна программа, то напишите, сохраните (!), скомпилируйте (если язык компилируемый), запустите (!) и проверьте программу у себя на компьютере. Программа должна быть сохранена в файл с каким-то именем и лежать в известном Вам месте. Если программа успешно работает, пошлите ее исходный текст проверяющей системе.

Будьте внимательны: читайте условие, в зависимости от него отправляйте в систему **результат работы** или **исходный файл** (для C/C++ это файлы с расширениями .c и .cpp), а НЕ всякие файлы настроек, проекта (.vcproj) или бинарные исполняемые файлы (.exe).

Для того, чтобы избежать путаницы, включите в операционной системе отображение расширений файлов или/и откройте файл, который Вы собираетесь отправлять, в Блокноте (или другом текстовом редакторе) и убедитесь, что это и есть Ваша исходная программа. Хотя бы первое время, пока не освоитесь. Затем в EJudge нажмите на кнопку «Browse» (Обзор) и укажите, где на компьютере лежит исходный текстовый файл с программой.

5.3.4. В выпадающем списке «Язык» выберите язык, на котором вы написали программу, например:

- для C: gcc - GNU C

- для C++: g++ - GNU C++
- для Pascal: fpc - FreePascal
- для Basic: yabasic - YaBasic
- для Python 2: python
- для Python 3: python3

5.3.5. Нажмите на кнопку «Отправить!» Программа передана системе. Ваша посылка должна появиться в таблице посылок (ниже кнопки «Отправить»), ее статус *Компилируется*.

5.3.6. Обновите страницу (нажмите F5 в браузере), чтобы посмотреть результаты проверки задачи системой.

5.3.7. Если результат проверки Вас не устроил, то можно в колонке *Посмотреть протокол* нажать на ссылку *Просмотр* и посмотреть, какие тесты не прошли и почему. Как правило, подробно расписываются не все тесты, а только первые 1-2.

5.3.8. За каждую дополнительную посылку по задаче у вас может сниматься небольшое количество баллов. Не расстраивайтесь и не придавайте этому большого значения: это сделано против спам-ботов, а также тех, кто любит отправлять решения в систему, поленившись запустить их у себя.

5.3.9. Если результат проверки *ОК и*ли *Ожидает проверки*, то перейдите к решению следующей задачи, нажав на ссылку Следующая задача (ниже таблицы посылок).

5.3.10. Переключаться между задачами можно по закладкам справа от задачи. Цвет закладки показывает статус задачи:
- серый - не было ни одной посылки задачи,
- зеленый - задача решена успешно,
- красный - неполное решение,
- белый - текущая задача

5.3.11. Можно задать вопрос по условию задачи дежурному преподавателю, нажав на ссылку *Отправить вопрос*, и по ссылке *Сообщения* прочитать ответ. Также, если вместо зелёной полоски ТУРНИР ИДЕТ вы увидите желтую полоску 1 непрочитанных сообщений / ТУРНИР ИДЕТ, то лучше просмотреть Сообщения, возможно, Ваш вопрос уже кто-то задавал.

5.3.12. В задаче могут быть бонусы — дополнительные баллы для первых нескольких сдавших.

5.3.13. В задаче может быть настроено уменьшение количества баллов с течением времени.

5.4. Возможные ошибки

5.4.1. Ошибка компиляции

Ваша программа не компилируется. Это может быть вызвано следующими причинами:
- в Вашей программе ошибка,
- Вы выбрали не тот язык, когда посылали решение.

5.4.2. Неправильный ответ

Проверяющая система нашла столько чисел и строк, сколько она ожидала. Но числа (слова) оказались другими. Если вывод численный, то у вас ошибка в решении. Если в задаче нужно вывести слово, то посмотрите, правильные ли Вы буквы используете (обратите внимание на то, какие буквы - большие или маленькие).

5.4.3. Неправильный формат вывода

Когда Ваша программа должна напечатать, например, одно число, а она печатает два числа, или одно число и какой-то текст, или количество строк другое, или надо напечатать два числа, а она печатает их слитно (проверяющая система не владеет телепатией, чтобы понять, где Вы хотели закончить одно число и начать другое). Если в условии задачи требуют вывести два числа через пробел, поставьте между числами пробел, а не запятые, двоеточия или другие знаки. Система разрешает ставить лишние пробелы в конце строки и не ругает за это.

5.4.4. Превышено максимальное время работы

На тесте программа выполняется дольше, чем разрешено. Придумайте алгоритм попроще / рациональнее. Может, программа просто зациклилась?

5.4.5. Ошибка выполнения (Runtime Error)

Ваша программа вернула не нулевой код возврата. Это может быть вызвано одной из двух причин:
- ваша программа выполнила недопустимую операцию во время выполнения, например, поделила на 0.
- ваша программа написана на C/C++ и Вы заканчиваете выполнение функции main с кодом возврата, не равным 0 (забыли написать return 0; ?).

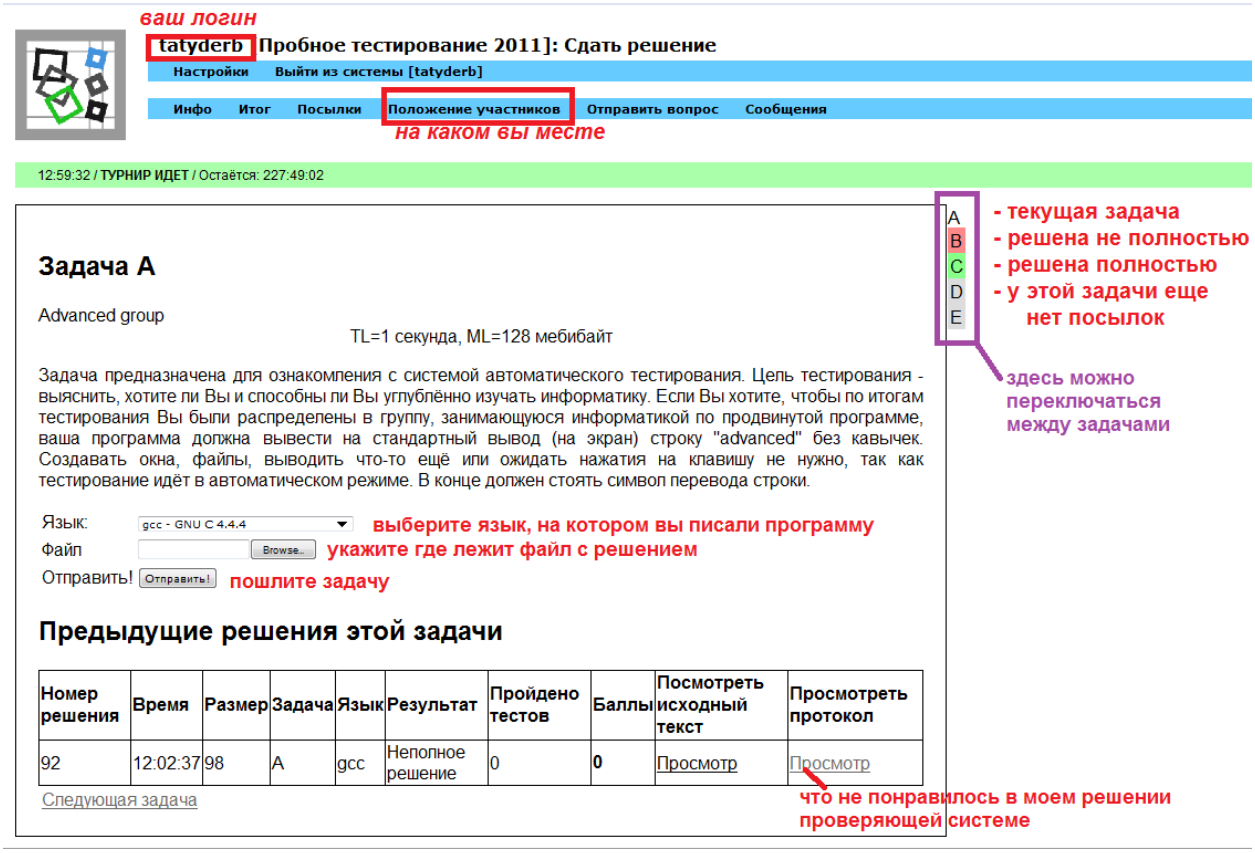
5.4.6. Ошибка проверяющей системы (System error)

Эта ошибка не должна появляться. Она означает, что во время проверки Вашего решения в системе возникли какие-то ошибки (система не может найти необходимых файлов, или при выполнении проверяющей программы возникла ошибка исполнения). Кроме того, в случае ElJudge, она может означать Memory Limit Exceeded. Если получите такую ошибку - срочно бегите к преподавателю или администратору. Впрочем, если в аудитории уже сообщили об этом — повторяться нет смысла, администраторы уже работают над проблемой.

5.4.7. Ожидает проверки

Решения по информационной безопасности чаще представляют собой не программы, а готовые ответы. Как правило, они находятся в состоянии «Ожидают проверки» до самого конца соревнования. Система не зависла, она даёт Вам возможность изменить своё решение. Засчитывается последнее из отправленных. Не стоит изменять верное на неправильное.

5.4.8. Пример интерфейса Ejudge



6. Литература

1. Столяров Андрей Викторович. «Программирование: введение в профессию». Том 1-3. М., «МАКС Пресс», 2016

2. Олифер В. Г., Олифер Н. А. «Компьютерные сети. Принципы, технологии, протоколы». Спб., «Питер», 2019.

3. Девянин П.Н., Тележников В.Ю., Третьяков С.В. «Основы безопасности операционной системы Astra Linux Special Edition. Управление доступом» ISBN 978-5-9912-0996-0 ББК 32.973.2-018.2я73 УДК 004.056