Ссылка на исходники:

https://github.com/sireagie/ctf\_stand/

Название таска: To Do Or Not To Do Dot Ru

Описание таска:

…

Описание стенда:

Весь стенд завёрнут в docker-контейнеры. Состоит стенд их 4х машин:

kali\_terminal (172.20.3.2) – терминальник с Kali, на которой открыто 20 vnc сессий (по одной на команду) + админская сессия; с помощью nginx по путям /team[0-19] доступны веб-интерфейс до vnc; для адимина есть путь /admin\_; у учёток user[0-19] небольшой набор прав, в sudo прописан доступ к apt install \*, по-идеи, большего участникам не надо будет, компромисс между удобством и возможностью помешать другим участникам;

web\_flask (172.20.3.3/todoornottodo.ru + 10.10.3.3) – веб сайт на flask;

ubuntu\_admin (10.10.3.2) – «админская» тачка, обычная ubuntu;

ftp\_server (10.10.3.4) – собственно, FTP-сервер.

172.20.3.0/24 – по легенде, внешние «белые» адреса, в этой сетке находятся терминальник и веб сайт.

10.10.3.0/24 – локальная сетка цели атаки, напрямую из терминальника доступа для тачек из этой сети нет.

Необходимо выделить порт 8888 (можно другой) на сервере для проброса в терминальник.

По нагрузке необходимо тестировать. Есть проблема по тому, что на одну команду всего одна vnc-сессия. Можно бы было, конечно, поднять 100 (20 команд \* 5 участников), но я не уверен, что инфра такое выдержит.

Также из грустного: потенциально первая команда, которая продвинется чуть дальше остальных, может легко перерубить путь для других участников, условно положив вебку после того, как получат ssh-досуп. Немного дизбалансно получается. Пока не знаю, что можно придумать. Либо попросить участников сильно стенд не ломать, либо поднимать 20 стендов (что, как уже выяснили, не получится).

Краткий ход решения таска:

Киллчейн выглядит примерно так:

kali\_terminal (172.20.3.2) -> web\_flask (172.20.3.3/todoornottodo.ru + 10.10.3.3): брут JWT токена, получение ключа от SSH от УЗ john на сайте

web\_flask (172.20.3.3/todoornottodo.ru + 10.10.3.3): получение root благодоря мисконфигу sudo (ssh \*)

web\_flask (172.20.3.3/todoornottodo.ru + 10.10.3.3): скан сети с помощью nmap

web\_flask (172.20.3.3/todoornottodo.ru + 10.10.3.3) -> ubuntu\_admin (10.10.3.2): получение пароля из запущенных процессов, полученных через snmp с дефолтной комьюнити-строкой

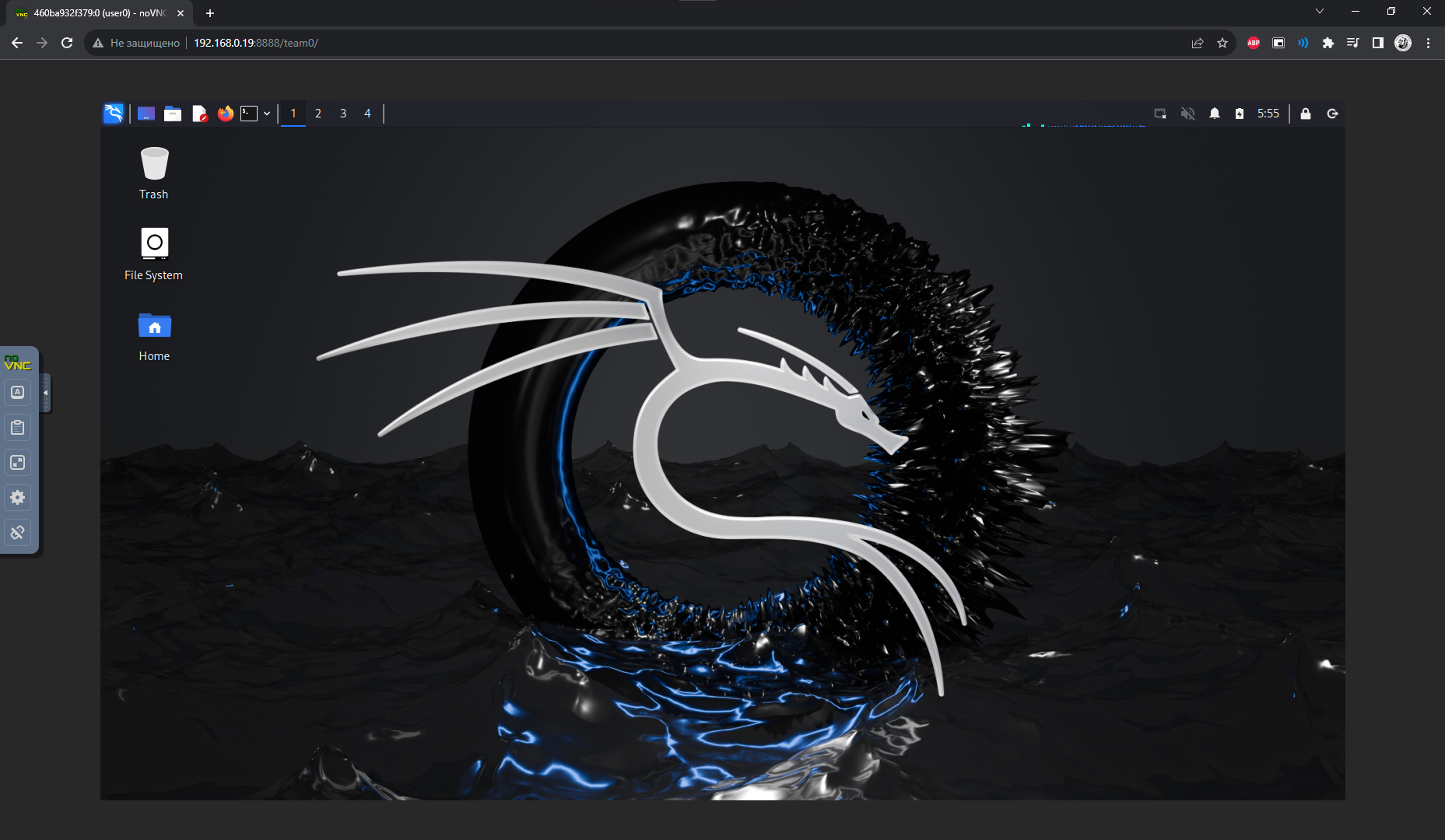
ubuntu\_admin (10.10.3.2): получение root благодоря наличию SUID флга на бинарнике питона

ubuntu\_admin (10.10.3.2): эксплуатация CVE-2023-32784 для получение мастер-пароля от KeePass

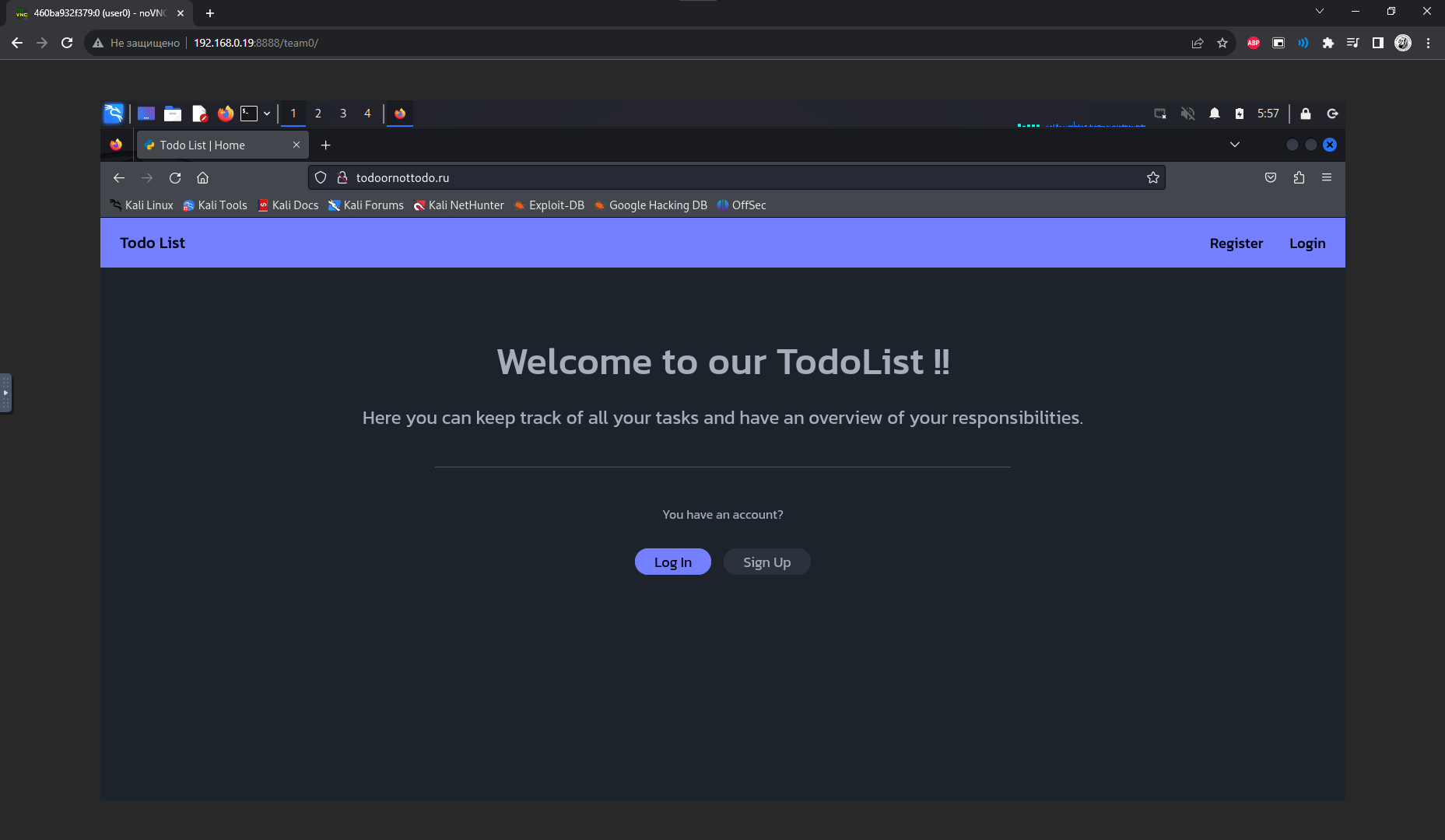
ubuntu\_admin (10.10.3.2) -> ftp\_server (10.10.3.4): подключение и получение флага

Решение таска по шагам:

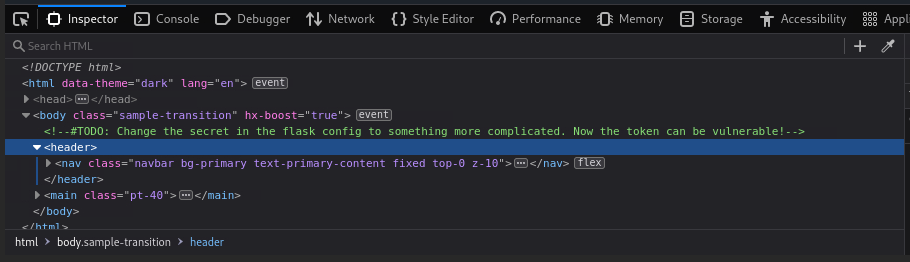
Заходим в веб-интерфейс терминальника http://domain:8888/team0/ (team0-19 в зависимости от команды). Логинимся под паролем, который получили от организаторов, получаем доступ к VNC Kali.



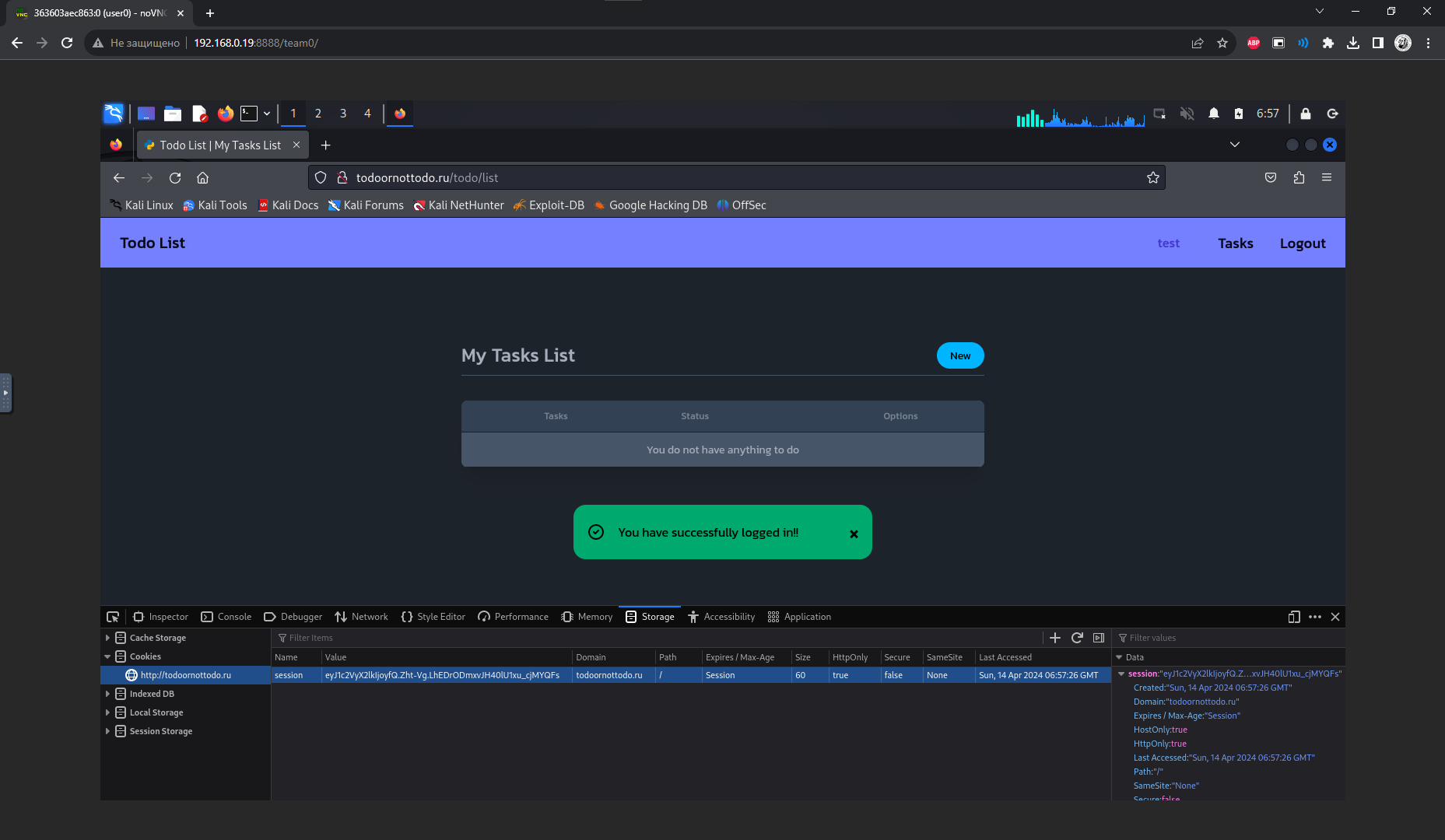
По заданию нам необходимо изучить сайт todoornottodo.ru (172.20.3.3). Зайдя на него, видим небольшой сайт с авторизацией для составления todo-заметок.



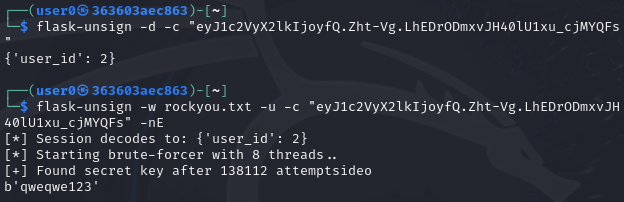
В devtool браузера можем увидеть комментарий «#TODO: Change the secret in the flask config to something more complicated. Now the token can be vulnerable!», что говорит нам о том, что сайт написан на flask и что секретный ключ для генерации JWT токенов выбран слабый.



Зарегистрируемся и залогинимся на сайте, чтобы достать куку session, в которой лежит JWT токен.



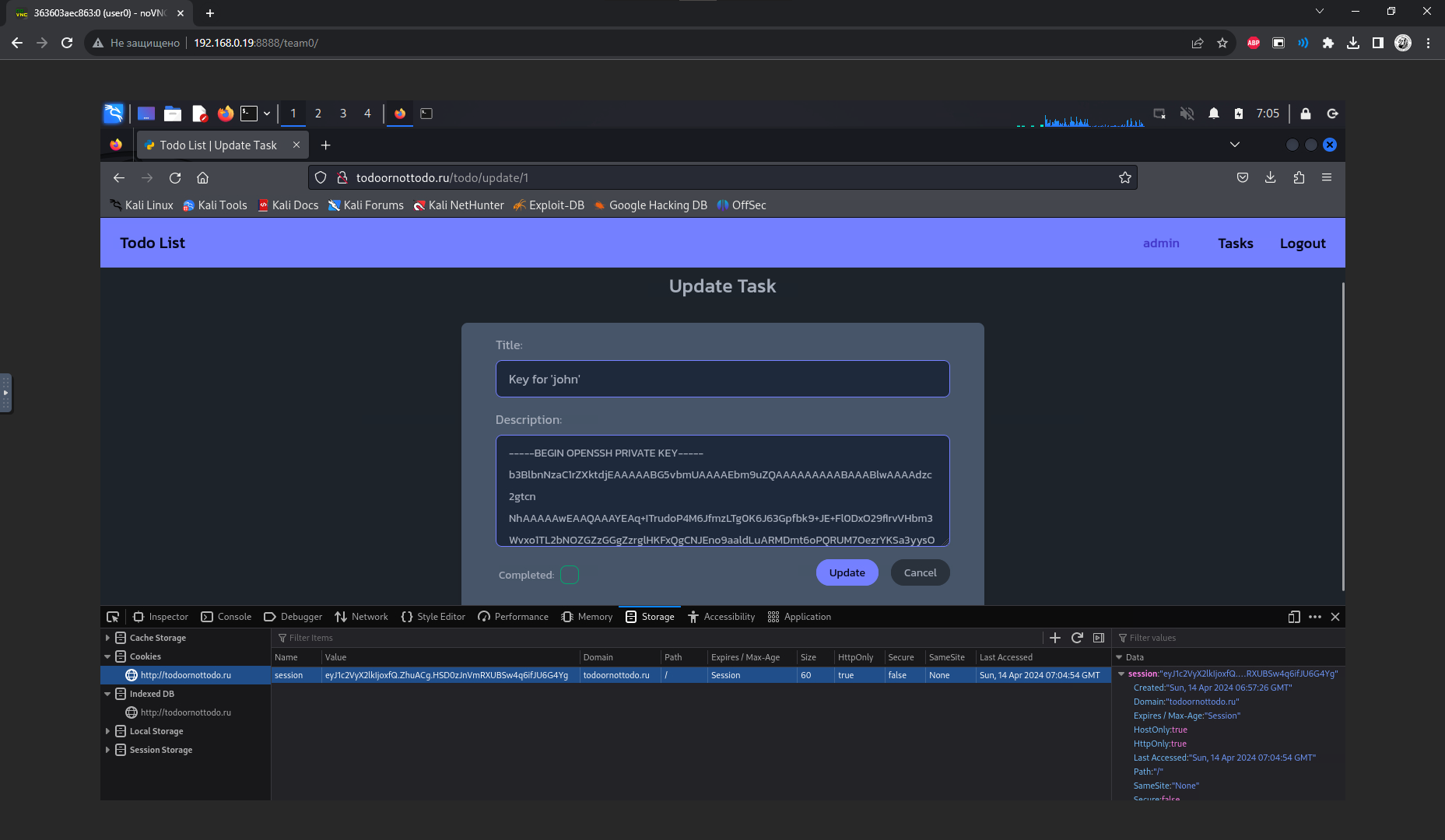
Воспользуемся утилитой flask-unsign для расшифровки куки и брутфорса секретного ключа (qweqwe123).



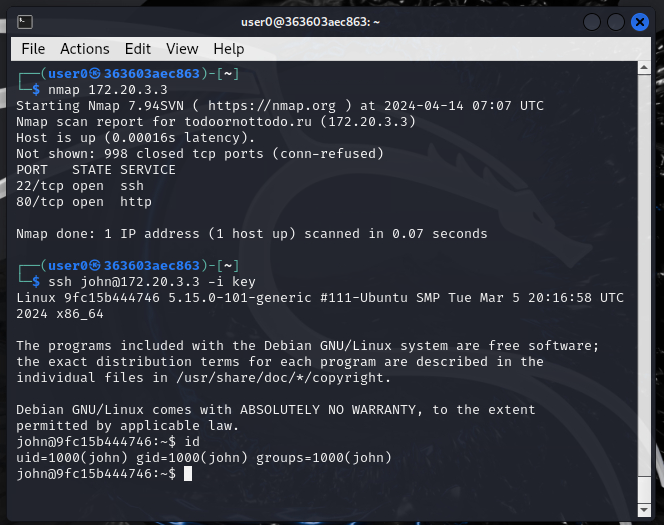
Предпологаем, что нам необходим юзер с id 1. Формируем с помощью той же утилиты куку.



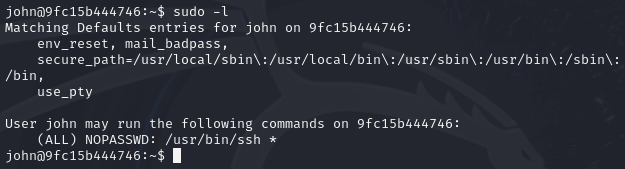
С поощью этой куки получаем доступ до админской учётки, у которой в заметках указан некий ssh-ключ от учётки john.



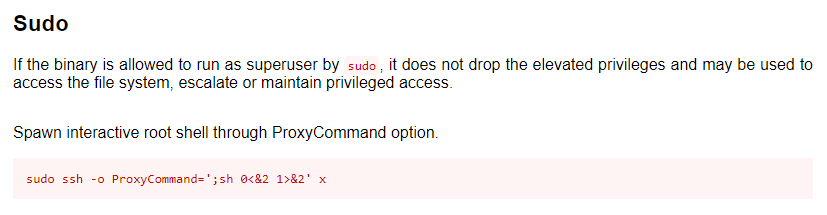
Если посканим с помощью nmap веб-сервер, то увидим открытый 22й порт. Пробуем залогиниться.



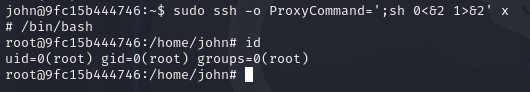
Получаем доступ к учётке john. Далее попробуем эскалировать права до root. Проверяя содержимое файла sudoers, видим, что для всех пользователей можно выполнить команду ssh \* через sudo без пароля.



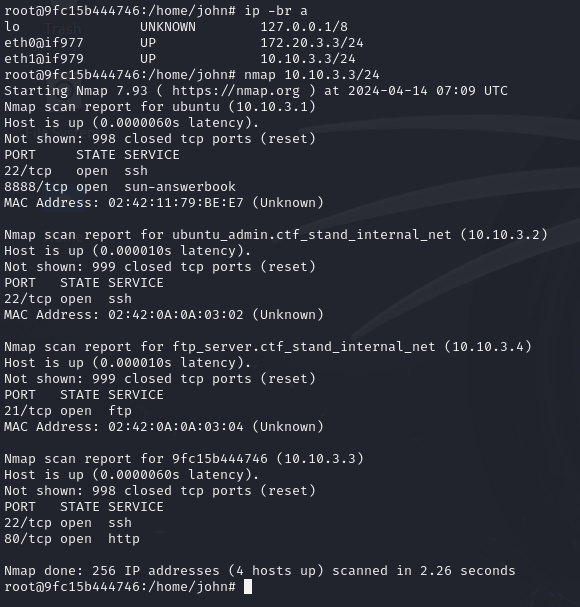
Погуглив, находим, как с помощью мисконфига sudo и ssh можно получить доступ к root.



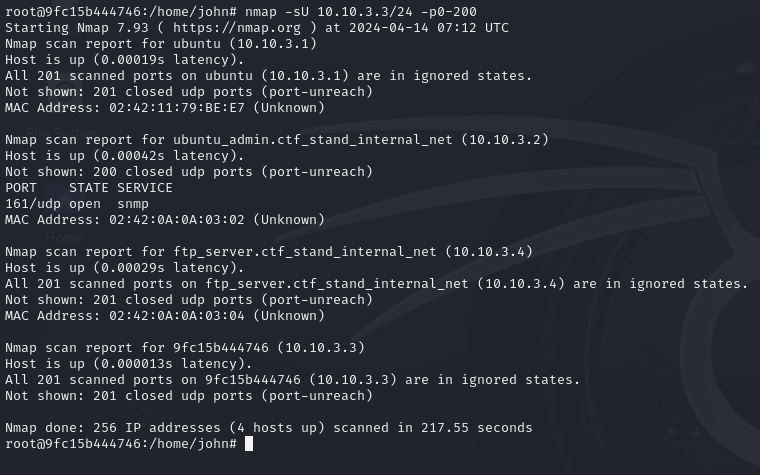
Пробуем, получаем доступ.



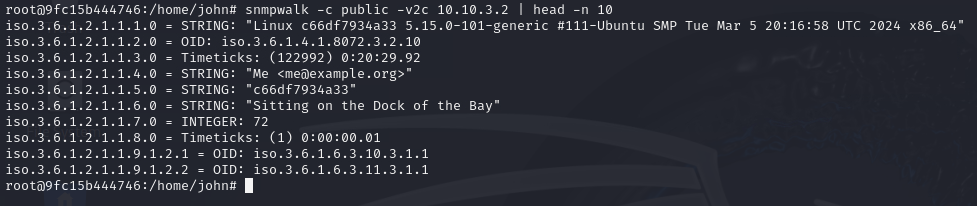
Далее будем расширяться по сети. Проверяем локальный адрес сервера, накатываем nmap и сканим сеть. Видим два хоста: у одного открыт порт 21 (ftp), у второго – 22 (ssh).



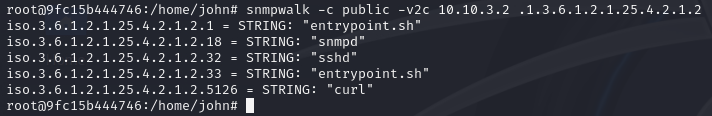
Подвох в том, что нужно не забывать сканить udp порты. Так у 10.10.3.2 видим открытый порт 161 (snmp).



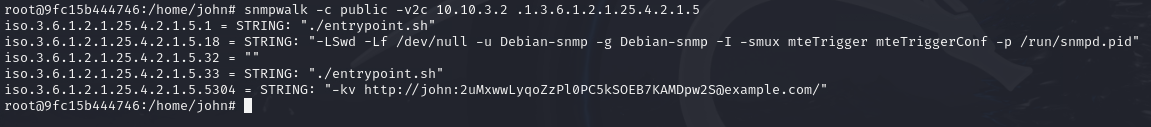
Установим snmp утилиты и проверим, есть ли у нас доступ. Как видим, можем получать данные по стандартной комьюнити-строке public.



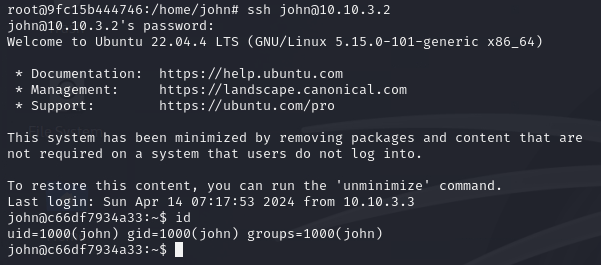
Из интересного, по OID 1.3.6.1.2.1.25.4.2.1.2 можем получить запущенные процесс. Здесь нас может заинтересовать запущенный постоянно curl.



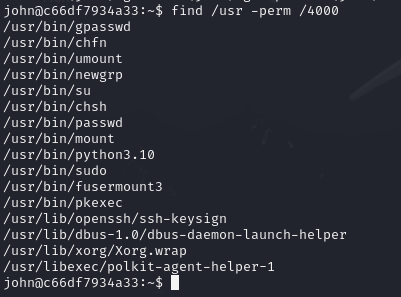
По OID 1.3.6.1.2.1.25.4.2.1.5 можем получить аргументы, с которыми были запущены процессы.



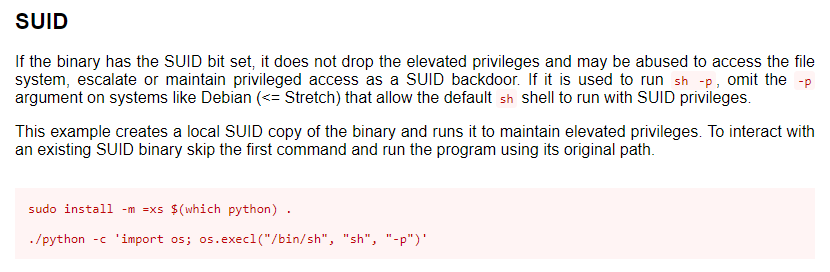
Тут видим, что у нас постоянно курлится сайт example.com с указанием кредов. Вспоминая, что у нас уже была учётка john для ssh веб-сервера, попробуем по этим кредам подключится к 10.10.3.2 под этими кредами.



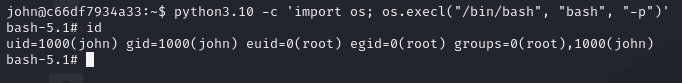
Получив доступ к учётке john будем опять пытаться эскалировать привелегии до рута. На тачке можем найти уязвимое место – у бинарника python3.10 установлен SUID флаг.



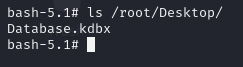
Погуглив, находим, как с помощью питона можно получить root права.



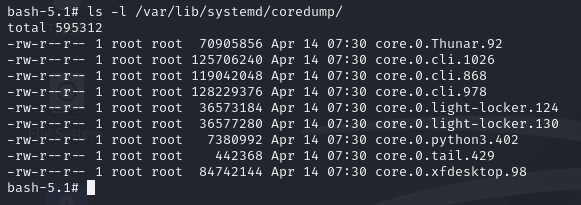
Так получаем root доступ на 10.10.3.2.



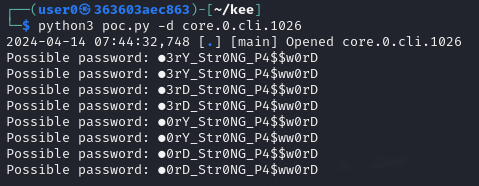
В домашней директории рута можем найти базу с паролями для keepass.



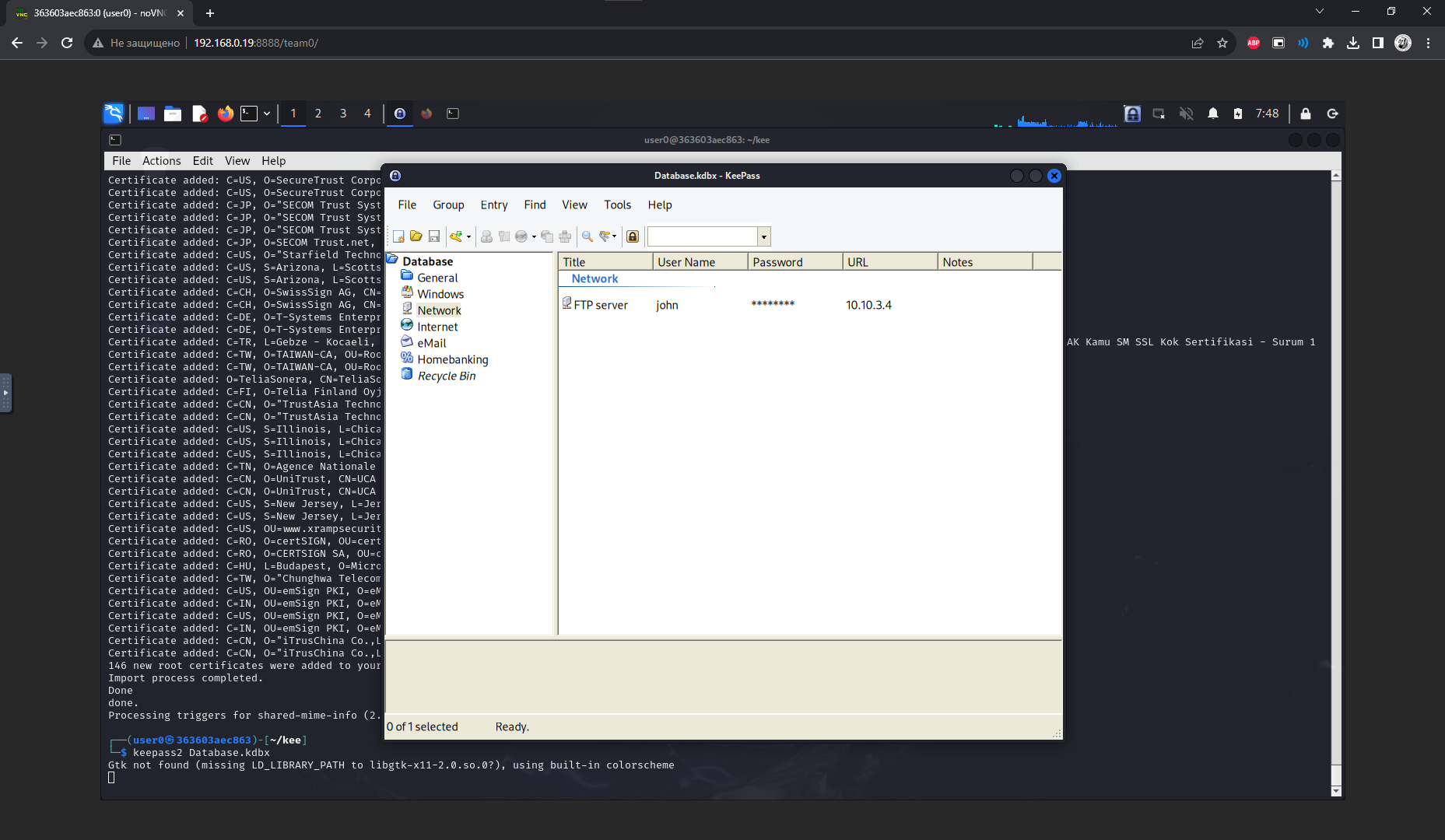
Опять же погуглив уязвимости keepass находим CVE-2023-32784, которая гласит о том, что мы можем вытащить мастер-пароль от базы в открытом виде из памяти процесса. Поскольку запущенного процесса keepass у нас нет, поищем в системе какие-нибудь дампы памяти. Находим несколько в /var/lib/systemd/coredump.



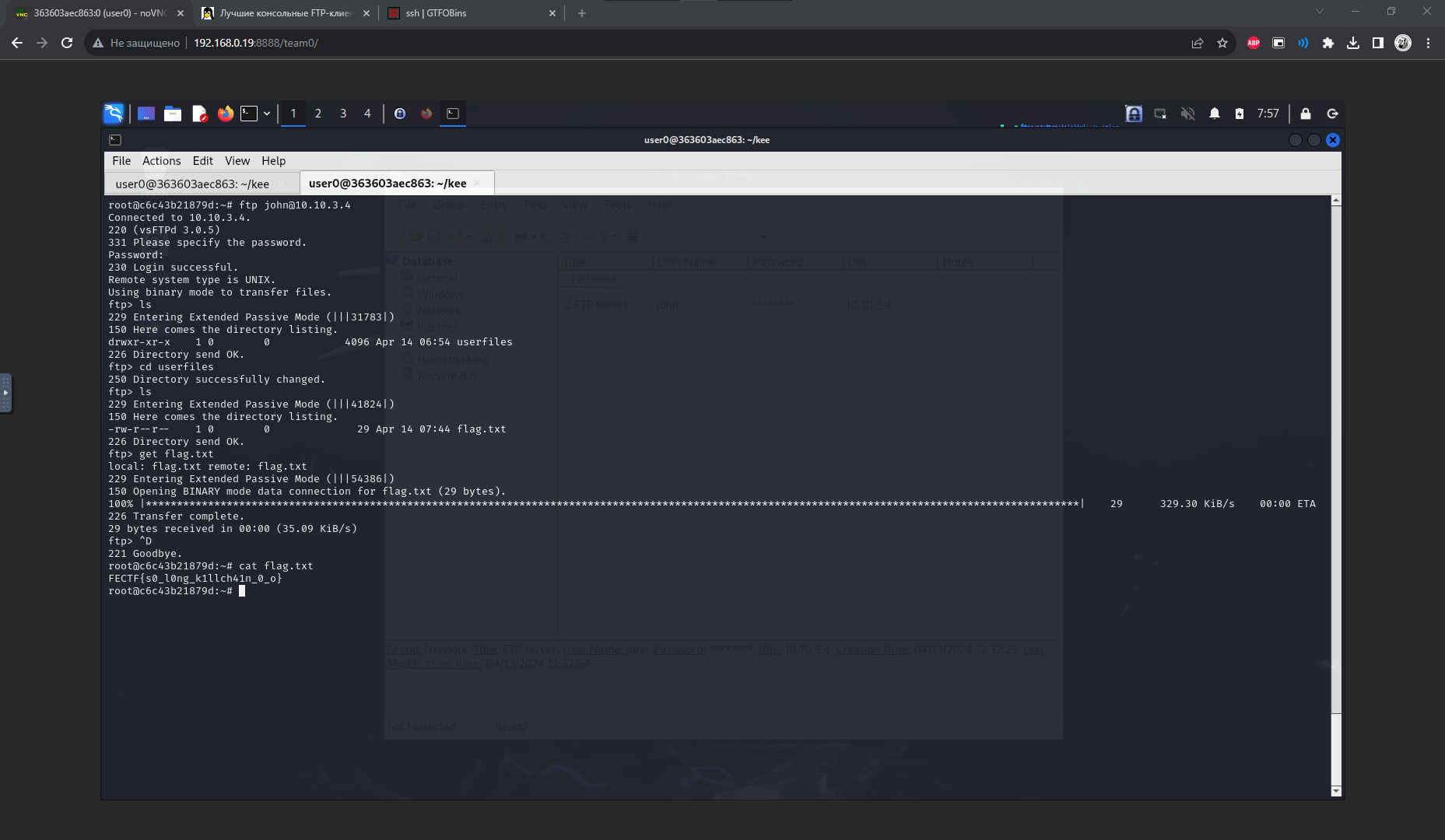
Копируем всё найденное себе на хост для удобства. Проэксплуатировать уязвиммость можно с помощью любого скрипта с github (например, https://github.com/dawnl3ss/CVE-2023-32784).



Потенциальный пароль выгружается без первого символа, что нужно учитывать. Вероятнее всего, что пароль V 3rY\_Str0NG\_P4$$w0rD. Пробуем открыть базу в keepass. Видим пароль от FTP.



Подключаемся в FTP-серверу и находим на нём файл flag.txt с флагом.



Всякие креды (учатникам раздать только первые, для user[0-19]):

# креды от терминальника для команд (и для учётки, и для vnc) kali\_terminal

user0:1k4NsyIGU0ZE8gaC6uwpbcVADWvQfq2R

user1:sud4pHfihQlza3R7MUb81qeoLEDyKwNT

user2:PERI6umDK5LC8zkgYpJ3ioNZAHUt9rlM

user3:1WX9oNjqGMm2OVtzTcsQhk5n8FuZepAD

user4:6bjt5GoqrsA8ixuTBfWFRkyDaOHgzehX

user5:cgbsO0voQ2LhjW6NFw7V1qHR95YZX3TG

user6:uV6tfZNcxKOP3zAoLGU8lHYr7wBCQXbm

user7:iQelNdx4c2IEqnBjhXTK8SUHkbuomasC

user8:nOFao3QPbIRK2lj0LiGC8ZeSMDt6kTgB

user9:okus21zvq9Fnrgei8MQ3HVGfZUj5O0Wl

user10:mBUO5RoMqhlWpySn402aFdrAwubZtkEe

user11:P6IvLwQZnKRGg4C0fi1TYoJ3eMjtUBkq

user12:ZdTH73Be960gvXLzWPRbOoJApyGaS5nI

user13:ZmiuMte8T7c0os4dfQFBDK3U1CnERgJN

user14:ljhNwz61JZGerg7W8xaoH4nYbi3pfvck

user15:oQlH1VjPYAOXbFNR3ycriCT5m0huqdZD

user16:1GcOKtTo7ZHPqu3iQ9NmMaxszpClVRWr

user17:ZoIGaViJuB0vYyw21tkLd8fPghnAMCWm

user18:tXMJwD0ni9psTzqaUlgyVIB5O8jQ3G61

user19:B7vLntfY0X19NDsSIRAlqmZaJeQpbMVT

# креды от теримнальника от админской учётки (для vnc) kali\_terminal

root:ZCDezj5lBsFwFUKr7I4y

# креды от учётки на сайте todoornottodo.ru

admin:hPr\*g"ZRs8I&1cV?xZI!

# креды от линуксовой учётки на веб сервере web\_flask

john:vYy@aEqBhFdTZ728z15NcyyeB^Cn6%%I

# креды от линуксовой учётки на админской машине ubuntu\_admin

john:2uMxwwLyqoZzPl0PC5kSOEB7KAMDpw2S

# креды от учётки на ftp сервере ftp\_server

john:VU5GNkumWT7SpI^!KAVZwSiiK1Dz@h!M

# мастер-пароль от кипаса

V3rY\_Str0NG\_P4$$w0rD

# ssh ключ от web\_flask

-----BEGIN OPENSSH PRIVATE KEY-----

b3BlbnNzaC1rZXktdjEAAAAABG5vbmUAAAAEbm9uZQAAAAAAAAABAAABlwAAAAdzc2gtcn

NhAAAAAwEAAQAAAYEAq+ITrudoP4M6JfmzLTg0K6J63Gpfbk9+JE+Fl0DxO29flrvVHbm3

Wvxo1TL2bNOZGZzGGgZzrglHKFxQgCNJEno9aaldLuARMDmt6oPQRUM7OezrYKSa3yysOZ

PvVu1Klc3eKiq4IV+njB2NpBdU6mqeeWTHW4IFK9sSAGxnRfqcdPaLlhJdn00DuD9fxj9j

Bzje6qQ6Ct7p62/qInQma9YKV/vzRE69FQEQOokw0U47WeLXDoKGGWm9m+Pkv37Rb2fDqJ

guPnFpVNQaBmj2BE5Kg/mUXzreIdsTD73DXf5TLrCAMJdQyl9X08PuxEiocKJ87ECgg9K+

g1egdvHydhwSth3rKB0GDXEXTChkj/wjJ5zituwO97dgiEix8SpbeJvxGb8jmzqA7B01Al

xQyiz0fK+UviiEjH/Ds55Wq+Ju+1ZGOqAA1jEsYhhZNTWaIqpsNcC24+KAOkqP/R1lDMbb

h+Jn7w9DBgPz3yTVrg2VpDjJ2UoLZlv+HPmXyzX9AAAFiBYRgjMWEYIzAAAAB3NzaC1yc2

EAAAGBAKviE67naD+DOiX5sy04NCuietxqX25PfiRPhZdA8TtvX5a71R25t1r8aNUy9mzT

mRmcxhoGc64JRyhcUIAjSRJ6PWmpXS7gETA5reqD0EVDOzns62Ckmt8srDmT71btSpXN3i

oquCFfp4wdjaQXVOpqnnlkx1uCBSvbEgBsZ0X6nHT2i5YSXZ9NA7g/X8Y/Ywc43uqkOgre

6etv6iJ0JmvWClf780ROvRUBEDqJMNFOO1ni1w6ChhlpvZvj5L9+0W9nw6iYLj5xaVTUGg

Zo9gROSoP5lF863iHbEw+9w13+Uy6wgDCXUMpfV9PD7sRIqHCifOxAoIPSvoNXoHbx8nYc

ErYd6ygdBg1xF0woZI/8Iyec4rbsDve3YIhIsfEqW3ib8Rm/I5s6gOwdNQJcUMos9HyvlL

4ohIx/w7OeVqvibvtWRjqgANYxLGIYWTU1miKqbDXAtuPigDpKj/0dZQzG24fiZ+8PQwYD

898k1a4NlaQ4ydlKC2Zb/hz5l8s1/QAAAAMBAAEAAAGAFU+G7DkRsko1Y+71GaatxTk0bW

6c3WLqu1rGr+2j2yAgZGJudmyHwLap0sEqT2zevw+yehiRsCLDWUw0yOZXEXzpb1fJz12F

lAqatwhXwramubRXrJhMAV6bDdp7IrZx+GfZYzX/tVeUQOmLHays4RGOC3QEyaIxvPHA5h

+KoGEFk/TbcqRdSjp0uxvUZpKGVydkcAgjkCR6MTAjgdkPyG/1wVniG/Z9OAT1O6S10kWW

UzsuOcTSFoWHNjRHsLxqy6wKOVwksxUBq6G+/u05KuQhPbdvAmy66smLaxTayjFTqTNsbF

yNhPi84P/tk58xcqCp1+8RtPr2FBT7NLr/TCDYQkbGrNq51gb/0UrxKgYs+wvz6OTHGjG5

iYNqnbcpDOjQ3CDUh3l30pyiQX9FDYBPIH/jzUDOrvH+QkSNxPCSOtBxFjO17H6QevV1C9

TfTZ7vhrgw4dLxmBAdaWpDmxco+YTG4OrF3B9rgWalZs2BzgO6bQTkcXQvCazCR/JbAAAA

wDFM0BEM+9gZ+j7CXDG/MRZ35VZ9xBEmTpFx3bketD7qAHzi4TaRa3rqsb7sf7qVzyGnVU

ym4kWNf87nHe2MTmFMQxLzG1C9wVcQnw/7qCUNZeJsrCgF2YLU7jcKVMMuaB4PHt1W3K99

lF1R90iYVVeUTVU+d7IxP7o9t7TpT+l5/1U4cHjaxaaZN9j3x/rWjGWRWyWBj3hqtS3hpc

Qit6gzMl3D4b84F+WkeCq1HD+gibd6vW0BVA5b30KEFJuzIQAAAMEA7graKciM7X+1UQik

de8h29R06vlnWaGzeVdVsb540GdPu8ypq/MYSDg+QYGn2t4pql3M2yjUQo2ltinrD16qEM

zOb7nHDE/F85jKT/ghU2yJer1lIdA4yiIoVuxZvPpi0sWVsKtQvcAdZCqAtN3i/uGtHPh6

cVJnzfvjz6pMoKNYH4Ts7tvvbFq71lAv/TYaVphxMexPjY2L4knXmcfhSAkAZx784T1QcT

DrtMywREgbD0AgSYS3pz8yGGOzL63LAAAAwQC42YlPutmoOW8PZDoDVsaPTSOocuGV90N3

7xbO26WCOnhNgXkcP3gWFHQEUNiTukP9iqm3HYMApXSTjXoeriwNZ3NvbSMw2cifh24qzx

GZrmWi82dQDtcz786YAgd7YbXiiZGuTirZ4vJwSeKIhOLQTs2sVjKB/7e2ieK/zmuR+FZ4

7nqf4Ht0kNQWdNgI+nAzt3W5mGXTEZD1aiu6ydlbxZatFQFswgdIro9Ys0Raem3CJkjYzo

879z4KrCtAslcAAAANdWJ1bnR1QHVidW50dQECAwQFBg==

-----END OPENSSH PRIVATE KEY-----

# флаг

FECTF{s0\_l0ng\_k1llch41n\_0\_o}