Решение заданий по направлению «Информационная и кибербезопасность», уровень магистратура/специалитет.

Раковский Станислав, ID 786278832.

Стоит упомянуть, что разбор данных задач был выложен на английском языке на моем блоге (скрыл статью с разбором, пока не закончится финал).

Ссылка на статью: <https://rakovskij-stanislav.github.io/posts-ctf/2021-03-27-ya-profi-2021-master-semifinal/>

Также был осуществлен разбор задач первого этапа магистерского уровня, тоже на сайте.

Рекомендуется читать разбор в режиме веб-документа (кнопка в правом нижнем углу, слева от ползунка масштаба).



Задание 3 (16 баллов).

|  |
| --- |
| Определить ключ, содержащийся в пересылаемом по протоколу http заархивированному файлу. Дамп трафика: <https://disk.yandex.ru/d/cDl8SoWPI25SvA>**Формат ответа:**itmo{...} |



Первое, что хочется отметить, - отсутствие мусорных пакетов.



Всё общение ведется между двумя участниками.

Всего нам предложено 119 стримов с sql-инъекцией и 1 стрим с запросом на скачивание архива.



Изучим структуру запросов с инъекцией. Воспользуемся инструментом CyberChef, рецептом URL Decode.

|  |
| --- |
| /?filter=user';select CASE WHEN (SELECT 'true' from cred where (data = crypt('oWF7b8TV8OL5',data)) and name = 'dbadm') ='true' THEN PG\_SLEEP(3) ELSE PG\_SLEEP(0) END;/?filter=user';select CASE WHEN (SELECT 'true' from cred where (data = crypt('nFmLGZKTuT9t',data)) and name = 'dbadm') ='true' THEN PG\_SLEEP(3) ELSE PG\_SLEEP(0) END; |

Посмотрев два запроса, становится видно, что используется классическая уязвимость с экранирующей одинарной кавычкой, и в самих запросах различается пароль. Происходит проверка. В случае успешного запроса ответ придет на три секунды позже. Давайте найдем такую аномалию в траффике. Для ускорения поиска используем фильтр по http-пакетам.



Мы нашли запрос с трехсекундным ожиданием. Пароль: RCTrddxU8GB8

Достанем архив, который находится в последнем пакете. Вариантов несколько: можно сохранить последний http-запрос в виде пакета байт и удалить начало вплоть до расширения архива, а можно воспользоваться экспортом объектов (File – Export Objects – HTTP)





Извлекаем содержимое архива, используя полученный из PCAP пароль.

Флаг: itmo{b6669795b1084948819538b8174c410a}

Задание 5 (18 баллов).

|  |
| --- |
| Средства антивирусной защиты зафиксировали заражение одного из компьютеров сети трояном. Система мониторинга записала трафик зараженного компьютера ([https://disk.yandex.ru/d/JFjIjdnfmkCz-\_w](https://disk.yandex.ru/d/JFjIjdnfmkCz-w)). Проанализируйте его и выясните, как вредоносное ПО попало на компьютер и какую информацию оно передало на свой управляющий сервер.**Рекомендуемое ПО:**для решения этого задания Вам понадобится анализатор пакетов Wireshark, любой дизассемблер (например, Ida Pro), а также будет полезен Python 3.7**Формат флага:**itmo{...} |

Изучим предложенный PCAP.

Первое – проверяем Conversations. Видим, что PCAP весьма мусорный – 172 IPv4 сессии.



Список DNS-запросов внушителен, и глаз цепляется за pastebin[.]com



Среди файлов в рамках HTTP видим и интересные названия файлов, и нестандартные порты 8000 и 5001. Приглядимся подробнее к общению с хостом 84.38.182[.]142.



Посмотрим запрос, связанный с первым файлом, dr.exe. Он был скачан с помощью браузера, и сервис, с которого скачали файл, представился SimpleHTTPServer 0.6, это один из модулей для Python, позволяющий быстро развернуть простой файловый сервер.



Предполагаем, что пользователь сам зашел на сайт и скачал файл. Возможно, в рамках фишинга.

Сам файл представляет собой проект в Visual C++, скомпилированный за 6-7 минут до того, как пользователь скачал файл. Это может быть следом Continuous Development, либо целевым фишингом. Также есть предположение, что нас дурят – заголовки клиента сформированы не браузером, а вредоносным ПО класса Downloader/Dropper (Загрузчик), который просто мимикрирует легитимный запрос с целью обхода систем, анализирующих траффик в инфраструктуре.



Сразу интересная строка: payload.py.enc



Заглянем в соседние строки:



Нашли захардкоженный адрес CnC, который совпадает со вторым запросом.

Представленный файл является вредоносным ПО класса Dropper/Downloader – его цель скачать и запустить нагрузку с управляющего центра, после чего избавиться от временных файлов. Тут есть первый изъян разработчика: программа не удаляет себя, хотя это было бы логично.



В программу статически слинкована библиотека libcurl, хотя это несколько излишне. Через эту библиотеку и происходит скачивание зашифрованной нагрузки.

Заодно находим второй изъян программы – она продолжает работать, даже если не смогла скачать нагрузку. Нормальным сценарием было бы удаление себя и завершение работы либо обеспечение закрепления в системе с периодическим опросом CnC на предмет наличия нагрузки.

Сама нагрузка расшифровывается с помощью алгоритма RC4 c ключом “Key\x00” (4 байта). Этого можно и не знать, стоит просто поставить точку останова на один из системных вызовов на скриншоте выше, так как вредонос сам расшифрует для нас файл payload.py.enc (под таким именем он curl-ом скачивает файл) и положит расшифрованный файл в payload.py. Наличие захардкоженного ключа для RC4 не является изъяном, это обычная практика в массовой малвари – захардкоженные ключи от XOR, RC4, TEA и так далее, но мы бы усложнили немножко магистрам жизнь, если б запрашивали ключ для расшифровки с CnC.

Также изъяном можно было бы посчитать обязательное наличие python на компьютере жертвы. Но это же соревнование, а не реальная малварь, поэтому можно считать это ограничительной мерой, чтобы участник случайно не запустил программу у себя на компьютере.

Сам пэйлоад взаимствует реализацию AES из публичного <https://github.com/boppreh/aes/blob/master/aes.py>, убрав режимы obf и ctr.

Скрипт можно классифицировать как Backdoor и Stealer – он имеет фунциональность:

* запуска произвольного кода (f\_82f3);
* отправки файлов на CnC (f\_a174);
* сохранения файлов на устройстве жертвы (f\_7c3e);
* изменения периодичности опроса CnC на новые команды (f\_b349).

Ниже – скрин с первым функционалом.



Бэкдор имеет два этапа:

1. Разведка – он отправляет на управляющий центр содержание глобальных переменных COMPUTERNAME, USERDOMAIN, PROCESSOR\_ARCHITECTURE, USERNAME. В качестве результата управляющий центр может отправить uid, назначенный данной копии ВПО. Этот uid бэкдор будет добавлять в каждый последующий запрос.
2. Отстук. Бэкдор спрашивает CnC на наличие команд, CnC может отправить \x00 или ссылку на pastebin, контент которой будет исполнен. На момент проведения контеста все ссылки уже были погашены. Этот этап как раз управляется четырьмя командами, описанными выше.

Структура команды с pastebin:

1. Первые два байта – количество исполняемых команд
2. Последующие байты – тело одной или нескольких команд. Тип команды идет по первым двум байтам блока. У команд есть идентификатор и параметры.

Второй этап ведет общение, накрытое AES ECB. Протокол не сложный: тело запроса состоит из вектора инициализации и зашифрованной нагрузки. Ключ для AES ECB зашит в скрипте: cfbb0862a99566223beb5c65cd35ce85264fe385870796958ee1d604d243127f

Напишем небольшой скрипт по расшифровке сообщений:

|  |
| --- |
| **from** payload **import** \* # comment out beacon init before running**from** binascii **import** unhexlify **as** unhex**while** True: dd = input() iv = dd[:32]; data = dd[32:]; a = AES(bytes.fromhex(key)).decrypt\_cbc(unhex(data), unhex(iv)); **print**(“>>>”) **try**: **print**(unhex(a)) **print**(unhex(a).decode()) **except**: **print**(a) **print**(a.decode()) **print**(“<<<”) |

Отфильтруем PCAP, чтобы получить только нужные пакеты (ip.addr == 84.38.182.142)



Первые два стрима относятся к скачиванию загрузчика и зашифрованного бэкдора.

Третий пакет содержит информацию, которую жертва отправляет о себе.

Отправляем:

|  |
| --- |
| DESKTOP-RNRS5K4\x00DESKTOP-RNRS5K4\x00AMD64\x00adm |

Получаем:

|  |
| --- |
| d660a9ee08f2b8db74f2f2c84e14679f4edf1a0fc315245e6ec1161f07dffe20 |

Этот наш uid.

Дальше я пропущу запросы, на которые CnC ответило нулём.

На отстук с клиента CnC отправило ссылку - https://pastebin[.]com/raw/YB1ttGcK

Результат, отправленный клиентом:

|  |
| --- |
| 000000SLEEP SET0001 Volume in drive C has no label. Volume Serial Number is 3808-37B4 Directory of C:\Users\adm\Downloads01/15/2021 10:59 AM <DIR> .01/15/2021 10:59 AM <DIR> ..01/15/2021 04:45 AM 186,684 5cd48b964566910009be0289-eight.jpg01/15/2021 10:59 AM 405,504 dr.exe01/15/2021 04:45 AM 1,176,071 Helping-your-new-cat-or-kitten-settle-in-1.png01/15/2021 04:24 AM 4,304,792 KeePass-2.47-Setup.exe01/15/2021 10:59 AM 14,507 payload.py01/15/2021 04:46 AM 28,204,528 python-3.9.1-amd64.exe01/15/2021 04:46 AM 92,482 \_115537288\_hi000618024.jpg 7 File(s) 34,384,568 bytes 2 Dir(s) 7,511,887,872 bytes free0002\\DESKTOP-RNRS5K4-------------------------------------------------------------------------------adm Administrator DefaultAccountf.savage Guest WDAGUtilityAccountThe command completed successfully. |

То есть были выполнены команды по обзору текущей директории (dir) и по итерации пользователей на компьютере (net user).

Следующая команда, которая была отправлена на исполнение: https://pastebin[.]com/raw/EJpXYJXb

Ответ от бэкдора – содержание пользовательской папки:

|  |
| --- |
| 0003 Volume in drive C has no label. Volume Serial Number is 3808-37B4 Directory of C:\Users\f.savage01/15/2021 04:36 AM <DIR> .01/15/2021 04:36 AM <DIR> ..01/15/2021 04:28 AM <DIR> 3D Objects01/15/2021 04:28 AM <DIR> Contacts01/15/2021 04:36 AM <DIR> Desktop01/15/2021 07:33 AM <DIR> Documents01/15/2021 04:28 AM <DIR> Downloads01/15/2021 04:28 AM <DIR> Favorites01/15/2021 04:31 AM 24 keepas.txt01/15/2021 04:28 AM <DIR> Links01/15/2021 04:28 AM <DIR> Music01/15/2021 04:30 AM <DIR> OneDrive01/15/2021 04:29 AM <DIR> Pictures01/15/2021 04:28 AM <DIR> Saved Games01/15/2021 04:29 AM <DIR> Searches01/15/2021 04:28 AM <DIR> Videos 1 File(s) 24 bytes 15 Dir(s) 7,509,471,232 bytes free |

Примечательно наличие файла от хранилища ключей keepass, но почему-то с расширением txt, а не kdbx. Видимо, злоумышленнику тоже приглянулся файл, и он решил его забрать следующей командой: https://pastebin[.]com/raw/rytChjvF

Результат:

|  |
| --- |
| 000400C:\Users\f.savage\keepas.txt00,k&VTaJ3kY5F&#L,Nf\_``C8> |

Команда: https://pastebin[.]com/raw/Q44gYdX5

Ответ:

|  |
| --- |
| 0005 Volume in drive C has no label. Volume Serial Number is 3808-37B4 Directory of C:\Users\f.savage\Documents01/15/2021 07:33 AM <DIR> .01/15/2021 07:33 AM <DIR> ..01/15/2021 05:32 AM 26 flag.txt 1 File(s) 26 bytes 2 Dir(s) 7,504,375,808 bytes free |

Злоумышленник решил проверить папку с документами.

Команда: https://pastebin[.]com/raw/NDmiyWft

Ответ:

|  |
| --- |
| 000600C:\Users\f.savage\Documents\flag.txt00itmo{4t3g3g3g343434g34g3g} |

Последняя команда от сервера: https://pastebin[.]com/raw/eE0MqKUr

Ответ:

|  |
| --- |
| 000700SLEEP SET |

Флаг: itmo{4t3g3g3g343434g34g3g}