山东大学网络空间安全学院

Python高级程序设计课程实验报告

|  |  |  |  |
| --- | --- | --- | --- |
| 学号：202100460065 | 姓名：李昕 | | 班级：21级密码2班 |
| 实验题目：实验十二. 综合实验，密码误用问题 | | | |
| 实验学时：2 | | 实验日期：2022/12/13 | |
| 实验目的：编写代码，完成功能，熟悉模块的用法 | | | |
| 硬件环境：  AMD Ryzen 7 5800H with Radeon Graphics     3.20 GHz  机带 RAM 16.0 GB (13.9 GB 可用 | | | |
| 软件环境：  系统：Windows11  编译器：IDLE | | | |
| 实验步骤与内容：  Insecure RNGs  Folder: bad\_rand\_rng  Here uses a re-implementation of C's rand() function. For this challenge, we use the linear-congruential generator included in glibc. It has an internal 32-bit state next. When calling the RNG, the state is updated using:  next = ((next \* 1103515245) + 12345) & 0x7fffffff  Then, the updated next is returned as the random number.  Challenge. Recover the plaintext of the challenge.  请查看“10 实验题目十-bad\_rand\_rng.rar”，在里面的bad\_rand\_rng.py中加入你的代码 Recover the plaintext of the fiel challenge.enc.  【密码误用问题】  1，在此问题中，使用的密码是基于伪随机数函数生成的，在这种rand函数生成的密码中，随机数序列是不变的的 ，故可以通过随机数数列进行解密。  2，分析代码可知，在enc文件里，偏移量iv传入了程序，而在加密函数中iv先于key生成，那么只要拿到iv的后四个bit就能由此生成key。  3，通过切片解码拿到iv后四位（也就是代码里的d），通过srand将其传入解密程序，传进去之后，开始生成key，此时生成的key就是加密原明文的密钥，之后就可以继续解密。  ADD代码部分：   1. a=iv[12:16] 2. b=struct.unpack('<I',a) 3. c=[i for i in b] 4. d=c[0] 5. srand(d) 6. key=byte\_rand(16)   同时，我们发现程序出现错误提示，原因是main主函数中有一句程序没有赋参数值，我们将其修改赋参：   1. args = parser.parse\_args('c')   之后程序顺利运行出结果。  执行结果：  Smart metering, smart parking, health, environment monitoring, and other applications drive the deployment of the so-called Internet of Things(IoT). Whilst cost and energy efficiency are the main factors that con-tribute to the popularity of commercial devices in the IoT domain, security features are increasingly desired. Security features typically guarantee authenticity of devices and/or data, as well as confidentiality of data in transit. Our study finds that whilst cryptographic algorithms for confidentiality and authenticity are supported in hardware on a popular class of devices, there is no adequate support for random number generation available. We show how to passively manipulate the on-board source for randomness, and thereby we can completely undermine the security provided by (otherwise) strong cryptographic algorithms, with devastating results. | | | |
| 结论分析与体会：  通过本次练习，我熟悉了解了Python的密码学编程知识，理解了伪随机数函数的优缺点，通过补全代码，进一步提升了我的代码能力和对python程序的掌握程度。 | | | |