**网络安全技术第三次作业**

陆圣珩

2017221302009

1. **身份认证技术**

**测试点 3-1**

1. **编制一个口令强度检测程序（语言不限）要求如下：**

**口令必须包含大写字母，小写字母，数字，特殊字符四种中的三种，长度要求8到30位。**

|  |
| --- |
| input: function (newVal) {        this.label = ""        if (newVal.length < 8 || newVal.length > 30) {          this.label += "密码长度要求8到30位; "        }        let rules = {          Capital: newVal.search(/[A-Z]/) === -1 ? -1 : 1,          Lower: newVal.search(/[a-z]/) === -1 ? -1 : 1,          number: newVal.search(/[0-9]/) === -1 ? -1 : 1,          mark: newVal.search(/[^\w\s]+/) === -1 ? -1 : 1        }        if (rules.Capital + rules.Lower + rules.number + rules.mark < 2) {          this.label +=            "密码必须包含大写字母，小写字母，数字，特殊字符四种中的三种;"        }      } |
|  |
| **备注：**采用 vue 完成编写，输入框使用 antd ，如果需要本地启动，需要本地配有 js + yarn 环境，进入 code 目录下运行 yarn install， yarn serve 即可访问 |

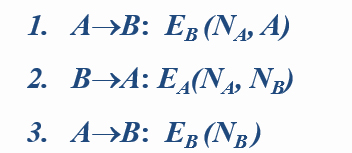
**测试点 3-2**

1. **针对基于密码的身份认证，主要存在重放攻击和中间人攻击两种安全风险，试分析基于对称密码的Needham-Schroeder 协议流程，判断该协议是否存在安全风险？如果存在，请给出攻击过程和改进方案。**

存在安全风险。

可以进行重放攻击。如果攻击者 C 保存了一个旧的会话 S，他便拥有旧会话密钥 KAB。 C 可以启动一个新的会话，直接把原先的数据包{KAB,A}KBS 发送给 Bob，然后 Bob 解密数据包后获得密钥 KAB。Bob 把{NB}KAB 发给 C，C 很容易计算出{NB-1}KAB，从而可以 冒充 Alice 与 Bob 进行会话。与 Oscar 之间的握手也不能验证 Oscar 身份。

改进方案：在 KBS 中加入时间戳，定期更换，必须使得在 A 和 B 建立会话时使用的 KBS 不能是以前使用过的，从而防止了重放攻击。

1. **使用公钥密码的Needham-Schroeder协议流程如下，分析该协议中存在的安全风险，给出攻击过程和改进后的方案：**

假设攻击者为 C，C 可以通过两次 NS 公钥协议来进行攻击。

第一次：

消息 1.1 A->C:EC(NA,A) 此时，攻击者开始第二次 NS 协议

消息 2.1 C->B:EB(NA,A)

消息 2.2 B->C:EA(NA,NB)

消息 1.2 C->A：EA(NA,NB)

消息 1.3 A->C：EC(NB)

消息 2.3 C->B:EB(NB)

这个过程结束后，B 以为它与 A 共享 NB，实际上是与攻击者 C 共享 NB。

改进方案：在步骤 2 中增加发送者的身份，即 B->A:EA(NA,NB,B)。这样保证步骤 3 中 A 只能 把 NB 发送给 B，不会被中间人 C 窃取到。