1993-IEEE Journal on Selected Areas in Communications-Li Gong

Protecting Poorly Chosen Secrets from Guessing Attacks

总结： 防止猜测攻击的基本思想是使攻击者获取的数据是足够不可预测的，攻击者无法离线验证猜测是否正确，而服务器可记录次数并在超过阈值时给出警告。作者以Needham-Schroeder为基准，提出了一系列保护技术和鉴别协议变体，并提出了检测协议是否存在猜测攻击漏洞的方法。

前言

▲典型口令猜测攻击的例子

1) UNIX口令系统，在/etc/passwd文件存储盐值及哈希值，攻击者拿到盐值猜测P'使之与上述哈希值比较来进行猜测攻击。

2) SunOS安全NFS. 引入公钥系统，/etc/publickey文件存储用户名，公钥和私钥，私钥由口令派生的密钥加密，用户登录时可用口令解密私钥，通过公私钥对的关系进行口令猜测攻击。

▲两个关联的系统使用相同口令也会导致危险。

3) kerberos 鉴别系统：服务器给用户的最初响应包含时间戳等信息，此消息加密密钥源于用户口令，攻击者可根据解密结果是否处于合理范围来验证猜测。

▲known Plaintext & verifiable text

已知明文指攻击者可在拿到一部分密文后预测部分或全部明文，即可以通过明密文对验证猜测结果。

可辨识文本则为已知明文攻击提供了基础，比如此文本是攻击者可识别的，则可以用于验证对口令的猜测。（或是攻击者可运算比对的）。

对于口令S，V满足以下条件可认为是可辨识文本.

① 攻击者可识别V. ②有$f()$使$V=f(S)$，攻击者可计算$f(S')$

③ $P\{S=S'|f(S)=f(S')\}>=t$.

若$t=1$，则可进行猜测攻击，$t=\frac{1}{n}$时，才可认为安全（n为可能口令总数）

保护技术

a. 普通两次握手

1. $A\to B: \{n\}_k$
2. $B\to A: \{f(n)\}_k$.

n为随机数，使用k加密，各自是安全的，但两者放在一起，则可猜测k'利用$f()$来检查关联

b. 双密钥两次握手

1. $A\to B: \{n\}_{k_1}$
2. $B\to A: \{f(n)\}_{k_2}$.

使用两个不同的k，只有同时猜测两个k值才能成功，只要其中一个well-chosen就很难猜测

▲k_1可能是公开的，k_2是用户选的弱口令，若$f(n)$很容易求逆如$f(n)=n+1$，则攻击者可对2次击得n'，并用k_1加密与1比对来验证

c. 使用随机数的两次握手

1. $A\to B: \{C, n\}_{k_1}$
2. $B\to A: \{f(n)\}_{k_2}$.

引入大随机数，confounder，合法接收者会忽略此消息。

攻击者仍可攻击2，但它必须拿到k_1或k_2和C才能构造消息1，不能因此无法验证猜测，confounder扮演了一次一密的角色

d. 使用掩码的两次握手

1. $A\to B: \{C_1, C_2, n\}_{k_1}$
2. $B\to A: \{C_2\oplus f(n)\}_{k_2}$

f(n)和n中可能包含可识别信息，用异或⊕来避免，B收到C_2要对$f(n)$做⊕运算，A已知C_2可解$f(n)$而攻击者不能单纯对2攻击，也无法构造1

鉴别协议（应用上述保护技术）

▲一般服务器产生会话密钥，因为确定型状态机生成高质量随机数较难，且客户端很难分辨可能被密码分析攻击的密钥（服务器参与鉴别）

▲一般会发送口令经函数处理后的结果，而非口令本身

a. 双向鉴别协议（k_a和k_b分别为S共享的密钥（可为口令），k为会话密钥）

1. $A\to S: \{A, B, n_{a1}, n_{a2}, c_a, \{t_a\}_{k_a}\}_{k_s}$
2. $S\to B: A, B$
3. $B\to S: \{B, A, n_{b1}, n_{b2}, c_b, \{t_b\}_{k_b}\}_{k_s}$
4. $S\to A: \{n_{a1}, k\oplus n_{a2}\}_{k_a}$
5. $S\to B: \{n_{b1}, k\oplus n_{b2}\}_{k_b}$
6. $A\to B: \{r_a\}_k$
7. $B\to A: \{f_1(r_a), r_b\}_k$
8. $A\to B: \{f_2(r_b)\}_k$.

1,3：生成随机数及实时信息解密后可验证身份

4,5：n_{a1}表示S解密了信息，n_{a2}作k的掩码，防止被好奇或内部猜测另C_a, C_b也可防止B内部攻击，它无法构造来验证猜测

6,7,8：挑战与应答

b. 紧凑型协议（Otway 87）

1. $A\to B: \{A, B, n_{a1}, n_{a2}, c_a, \{t_a\}_{k_a}\}_{k_s}, r_a$ 明文
2. $B\to S: \{A, B, n_{a1}, n_{a2}, c_a, \{t_a\}_{k_a}\}_{k_s}, \{B, A, n_{b1}, n_{b2}, c_b, \{t_b\}_{k_b}\}_{k_s}$
3. $S\to B: \{n_{a1}, k\oplus n_{a2}\}_{k_a}, \{n_{b1}, k\oplus n_{b2}\}_{k_b}$
4. $B\to A: \{n_{a1}, k\oplus n_{a2}\}_{k_a}, \{f_1(r_a), r_b\}_k$.
5. $A\to B: \{f_2(r_b)\}_k$.

c. 使用随机数的协议（服务器分发nS，检测重放攻击）

1. $A\to S: A, B$　　2. $S\to A: A, B, n_S$

（剩余5步同b（将t_a, t_b替换为n_S，更简洁）

d. 认证协议（无需会话密钥，只需身份认证，甚至是单方认证）

1. $A\to S: A$
2. $S\to A: n_S$　（意义？（随机））
3. $A\to S: \{A, c_a, \{n_S\}_{k_a}\}_{k_s}$

▲在智能卡领域很有用，只需存k_s即可一个PIN即可使用，不需防篡改，也无需用户管理

e: 私密公钥协议（用户无法记住k_s，可让S为A,B分别派发公钥，消息2）

1. $A\to S: A, B$
2. $S\to A: A, B, n_S, \{k_{sa}\}_{k_a}, \{k_{sb}\}_{k_b}$
3. $A\to B: \{A, B, n_{a1}, n_{a2}, c_a, \{n_S\}_{k_a}, k_{sa}, n_S, r_a, \{k_{sb}\}_{k_b}$
4. $B\to S: \{A, B, n_{a1}, n_{a2}, c_a, \{n_S\}_{k_a}, k_{sa}, \{B, A, n_{b1}, n_{b2}, c_b, \{n_S\}_{k_b}\}_{k_{sb}}$
5. $S\to B: \{n_{a1}, k\oplus n_{a2}\}_{k_a}, \{n_{b1}, k\oplus n_{b2}\}_{k_b}$
6. $B\to A: \{n_{a1}, k\oplus n_{a2}\}_{k_a}, \{f_1(r_a), r_b\}_k$
7. $A\to B: \{f_2(r_b)\}_k$

▲k_{sa}(k_{sb})必须安全，不可公开，否则消息2可进行猜测攻击。同样，k_{sa}也应是随机

f: 直接鉴别协议（可信S不存在，AB共享k_{ab}，生成会话密钥k）（poorly chosen）

1. $A\to B: r_a, \{k_{ab1}\}_{k_{ab}}$
2. $B\to A: \{B, A, n_{b1}, n_{b2}, c_b, \{r_a\}_{k_a}\}_{k_{ab1}}$
3. $A\to B: \{n_{b1}, k\oplus n_{b2}\}_{k_{ab}}$
4. $B\to A: \{f_1(r_a), r_b\}_k$
5. $A\to B: \{f_2(r_b)\}_k$.

k_{ab1}为公钥，A每次会话生成一个

检测漏洞（收集信息并查找是否包含可以验证猜测的信息）

M：所有消息的集合；R：规则集合，组合M可得到他；W：攻击者希望知道的消息

G：包含攻击者选择去猜的secret. V：M之内的可辨识文本

一般一个G中元素对应一个可辨识文本，若攻击者拿到G中g, g'，并得到两个t, t'

若$x\in W\cap V$，则：$P\{g=g'|t=t'\}$足够大，则可验证猜测.

⇒可归类为图论中的路径搜索问题，M中消息为结点，W∪G可达

查看一个W结点路径上是否有G，若有，路径称为可辨识文本.（路径的深度