|  |
| --- |
|  |
|  |
| **实验题目：** **ProVerif验证实验** |
| **实验摘要：**  1. ProVerif环境配置与验证Hello.pv文件：  这个实验要求首先配置ProVerif环境，按照实验手册安装ProVerif。  随后，对一个简单的Hello.pv文件进行验证，确保ProVerif的基本功能正常。  2.Handshake Protocol的验证与改进：  学习ProVerif手册的3.1-3.3部分，了解ProVerif的基本使用和语法。  对给定的ex\_handshake.pv文件进行验证，分析ProVerif的验证结果，识别存在的攻击手段。  基于ProVerif的分析结果，提出改进ex\_handshake协议的方案，以解决中间人攻击。  3. Needham-Schroeder密钥共享协议的分析：  学习Needham-Schroeder密钥共享协议的基本原理与协议流程。  利用ProVerif分析附件中的Needham-Schroeder密钥共享协议的三个变体。  详细说明各个协议变体存在的缺陷，并尝试给出攻击方式。 |
| **题目描述**  1. ProVerif环境配置与验证Hello.pv文件：  这个实验要求首先配置ProVerif环境，按照实验手册安装ProVerif。  随后，对一个简单的Hello.pv文件进行验证，确保ProVerif的基本功能正常。  2.Handshake Protocol的验证与改进：  对给定的ex\_handshake.pv文件进行验证，分析ProVerif的验证结果，识别存在的攻击手段。  基于ProVerif的分析结果，提出改进ex\_handshake协议的方案，以解决中间人攻击。  3. Needham-Schroeder密钥共享协议的分析：  利用ProVerif分析附件中的Needham-Schroeder密钥共享协议的三个变体。  详细说明各个协议变体存在的缺陷，并尝试给出攻击方式。 |
| **过程：**  1.在Centos7.6上安装必要的组件ocaml,因为ProVerif是基于OCaml语言开发的，因此在安装ProVerif之前需要安装OCaml      2.安装ProVerif软件包，复制到虚拟机，解压并进入文件夹，编译。    3.将测试文件hello.pv复制到proverif文件夹下，验证    4.验证handshark.pv，其输出显示了对握手协议的验证结果      根据ProVerif的分析结果，攻击者成功获取了私有比特字符串 s 的路径是通过对数字签名和加密消息的操作，以及中间人攻击。攻击者能够获取签名的内容，然后伪造数字签名，并将其发送给服务器，进而获取加密的比特字符串。接下来，攻击者成功解密加密的消息，最终获得了私有比特字符串 s。  改进建议：  1.增强数字签名验证： 加强数字签名的验证，确保在验证签名之前检查签名中使用的公钥是否与协议中预期的公钥匹配。这可以防止中间人伪造数字签名。  2.使用证书： 引入证书以确保公钥的真实性。客户端和服务器可以交换证书，用于验证对方的公钥。这样可以避免中间人伪造公钥的问题。  3.会话密钥协商： 考虑采用会话密钥协商的机制，以避免在通信过程中直接使用长期密钥。这可以增加协议的前向保密性。  5.验证Needham-Schroeder协议三个变体。攻击者能够截获消息并通过适当的解密操作获取Na,Nb,实施中间人攻击        解决方式：  1. 引入唯一标识符，确保每个消息都有唯一性。避免重复使用相同的标识符  2.引入密钥更新机制，周期性地更新会话密钥，降低长期密钥被攻击的风险。  3.加强身份认证，确保通信各方的合法性。 |
| **总结**  通过本次实验，我对协议设计和验证有了更深入的理解。ProVerif是一个强大的验证工具，为我们提供了形式化分析协议的能力。在实验中，我学到了如何使用ProVerif来建模协议、运行查询，以及解释验证结果。  主要遇到的问题是安装ocaml的管理依赖组件opam时出错，切换了yum源后解决。 |
| **参考文献**  https://bblanche.gitlabpages.inria.fr/proverif/manual.pdf  https://blog.csdn.net/qq\_39743001/article/details/120706734 |