Security Issues in NIS

James L. O’Brien
November 10, 2000

As environments have become more distributed and heterogeneous over the years, the need for a service capable of maintaining common configuration files, including authentication information, became essential for large-scale Unix environments. The Network Information System (NIS), formerly referred to as Yellow Pages (YP), was born out of this need. While NIS is notoriously insecure, it is still widely used by many organizations.

This paper is divided into four major parts. The first part of this paper provides an overview of NIS. Next this paper identifies security weaknesses in NIS, followed by methods of improving overall NIS security. Finally, the last section of this paper briefly discusses alternatives to NIS.

Introduction to NIS

The original design intent of NIS was to ease the administration of networks by providing centralized management of configuration information and other resources that could be shared across multiple systems. Most commonly, such services provide centralized account information by sharing user and group information across systems on the network. Originally conceived to work hand in hand with Network File System (NFS), it was released by Sun Microsystems in 1985 [1]. It is necessary to maintain synchronization between User IDs (UID) and Group Ids (GID) to share files across multiple networked systems. A failure to do so could result in incorrect access permissions and potential compromises of security. NIS’ purpose was to provide this synchronization through the use of maps [1].

NIS operates along a client-server model with an overall grouping called an NIS domain. All the systems within such a domain can share a variety of information. This can include password entries, groups, and email aliases. An NIS domain has one primary or master server [2], with the option of one or more slave servers. All systems in the domain not serving as an NIS server are considered clients. The NIS master maintains the actual flat text or source files used to create the NIS maps. These files commonly include /etc/passwd and /etc/group (for a more complete list, please see Appendix A). An NIS server does not perform its lookups against these flat text files. Instead, such files are converted to DBM files and are typically stored in a subdirectory of /var/yp. These binary files are indexed sequentially to allow quick lookups of information based on a given key value [1].

Accessing centralized information is performed using the Remote Procedure Call (RPC) protocol from the NIS client to the NIS. The RPC protocol allows a client or caller to send a message to the server for execution remotely. The server then processes the call, computes the results, and returns them to the client [3]. In essence, this protocol allows an application or program running on one system to call and execute subroutines on another computer. An NIS server must run multiple services to provide access for clients to NIS resources. These services on the NIS master are **ypserv**, **ypbind**, **ypxfrd**, **yppasswdd**, and **ypupdated**. On an NIS slave server **ypserv** and **ypbind** must be running, while the others are not required. Finally, an NIS client only needs the **ypbind** service to be able to participate in an NIS domain [4]. In addition to these NIS specific daemons or services, all systems must be running the RPC portmapper daemon (i.e. **portmap** or **rpcbind**). It should be noted that the names of each service might vary slightly depending on the platform.

Weaknesses in NIS

There are a variety of security weaknesses in NIS that leave the environment susceptible to a multitude of attacks. The first weakness in NIS focuses on configuration issues. Many default installations of NIS allow any user with an NIS account access to any system in NIS. Access can be restricted based on netgroup memberships, but enforcement of netgroups is a conscious configuration step. This configuration varies across Operating Systems, but typically involved modifying /etc/nsswitch.conf and the appropriate local files (i.e. /etc/groups, /etc/passwd, /etc/shadow) [4].

The second major weakness is that NIS typically operates in a broadcast mode [4]. The clients locate an NIS server on their subnet by broadcasting to the broadcast address of said subnet. As a result, NIS can be subject to a confidentiality attack resulting in unintentional information disclosure via a network sniffer such as tcpdump and snoop. It is possible to watch a network segment’s broadcasts to collect information on the NIS environment. This information can include the NIS domain name, as well as master or slave servers on the local subnet. In the event there are no NIS servers on the local network segment, NIS environment information can still be acquired by watching communications from NIS clients to NIS servers on other segments.

The third major weakness in NIS deals with password restrictions. Traditional thinking states that if disclosure of the password cannot be prevented, that the password must become more complex and thus more difficult to crack using conventional password cracking utilities. NIS aggravates identity attacks on account information because it does not support the two most common methods of increasing this complexity. NIS does not support password-aging [4]. Password aging is the process by which a password must be changed after a maximum amount of time has passed, but no sooner than a minimum amount of time has elapsed. NIS also does not support password strengthening. Password strengthening focuses on making the password string itself more complex through the requirement of numeric or non-alphanumeric characters, as well as limitations on the occurrence of characters. While password strengthening can be enforced at the NIS master server, it cannot be enforced on the clients or slave servers. As a result, any user who modifies their password on a system other than the NIS master will not have password strengthening parameters enforced during the password change.

Another weakness of NIS is that its clients do not perform any validation of the server they connect to other than to ascertain that the server is operating in the correct domain [5]. First, a system can establish itself as an NIS server by simply specifying the NIS domain name and performing the other necessary configuration steps. As a result, a user could create a false NIS master server, and using an availability attack such as a Denial of Service tool, knock the actual NIS master server off of the network. The false system could then assume the identity of the NIS master, including its name and IP address and respond to NIS client requests. In this manner, a user could gain access to an NIS client system by replacing the real NIS master and its maps with artificial information on a false server. It should be noted that in some NIS implementations, the false system would not even need to assume the real NIS masters IP address. An NIS server on a local network segment can attract client bindings if the NIS master is on a separate network [6].

Unintentional disclosure of NIS maps is a very serious issue. NIS servers do not authenticate the client systems as they insert themselves into an NIS domain [5]. As a result, it is possible to configure a system as an NIS client by just collecting the NIS domain name and an NIS server. Once this is achieved, the user can configure the client and start the ypbind process to access resources in the NIS domain [6]. Once a system belongs to an NIS domain it has access to all information provided in maps in that domain. In essence, a user on an NIS client or server can dump map information using the ypbind or ypmatch commands. This weakness is especially an issue for environments that believe they have implemented the use of shadow passwords. By default, NIS does not support a map for the /etc/shadow file. When NIS maps are created from the original source files the /etc/passwd and /etc/shadow file are merged back together and appear as they would in older Unix releases. This means that hashed passwords are stored in the NIS maps, readily available for capturing using ypcat. For example, by running ypcat passwd.byname > /tmp/userlist, a user can dump the entire contents of an NIS map containing username and password pairs. Password cracking programs such as crack can then be used to attack the passwords until a username-password pair can be compromised. This can be avoided in an all Solaris/SunOS installation, but cross platform installations will not be able to hide encrypted passwords [7].

There are other vulnerabilities and weaknesses in NIS and RPC that allow man in the middle attacks and buffer overflow exploits, however many of these can be corrected with vendor patches.

Improving NIS Security

Although NIS is very insecure, there are many steps that can be taken to improve the security of NIS and to mitigate the exposure an organization may face. The following steps will dramatically improve the security of the NIS domain:

- Apply and maintain Operating System patches, especially security-related patches for NIS [7].
- Configure border routers and firewalls to block all RPC traffic from untrusted networks [7].
- Use compatibility mode for NIS and implement netgroup enforcement to restrict user access to systems [4].
- Restrict access to NIS maps by using /var/yp/securenets [4].
- Restrict access to the portmap service by implementing versions of portmap and rpcbind that support TCP Wrappers [7].
- Hide encrypted passwords from the NIS maps by implementing shadow passwords in NIS through the use of passwd.adjunct in organizations using only Solaris/SunOS [7].
- Supplement passwords through the use of two-factor authentication relying on tokens that generate an additional one-time password.
- Conduct regular audits of the NIS environment to maintain current information and verify a valid working configuration.
- Use assessment tools such as SATAN (SAINT) or commercial tools on the NIS environment to evaluate the environment’s security.

Alternatives to NIS

There are many alternatives for organizations that cannot invest the energy necessary to secure an existing NIS installation or are not comfortable with the security weaknesses of NIS. The two alternatives highlighted are only two of the possible alternatives available. The most common alternative to NIS is NIS+. NIS+ was released by Sun Microsystems's in the early 1990's as a replacement for NIS. The replacement was originally received with a cold welcome due to many reasons including reliability, support overhead, and lack of support by other vendor platforms. NIS+ has several differences with NIS including a more hierarchical domain structure, with multiple sub domains. In addition, security in NIS+ is enhanced to support a network logon; thus allowing for authentication of the client and server and reducing the likelihood of insertion type attacks by unauthorized clients [5].

The Lightweight Directory Access Protocol (LDAP) has emerged as a new mechanism for Unix authentication and authorization. The remarkable opportunity with LDAP is its support for application authentication and not just the Operating System. This scalability stands to position LDAP as a standard for enterprise wide authentication and other services. LDAP servers may be able to address many of the weaknesses in NIS as well as NIS+. LDAP provides a dedicated TCP/IP port for communications. This allows easy filtering of the protocol without disrupting other services. Support for Secure Socket Layer (SSL) encryption is being developed. In addition, client authentication to the server with support for access control lists allows increased control and granularity of access [8].

The Network Information System still serves the need of maintaining synchronized configurations across groups of systems on a network. Security concerns have increased over time; however, NIS has not reflected these concerns. As a result, NIS is still plagued by numerous security weaknesses that strongly merit the consideration of alternatives such as NIS+ or LDAP. While such conversions or implementations can be difficult and time consuming, the services provided by NIS are central to any distributed computing environment. Weaknesses at this scale in the foundation of network computing can prove very destructive or disruptive to an organization if they are exploited.

Appendix A

Common files shared via NIS:

- /etc/passwd
- /etc/group
- /etc/hosts
- /etc/services
- /etc/protocols
- /etc/aliases
- /etc/rpc
- /etc/netgroup
- /etc/ethers
- /etc/bootparams

References

[1] "Using NIS." URL: http://userpages.umbc.edu/~jack/ifsm498d/llb-nis.html (6 Nov. 2000).

[2] "NIS – Network Information System." URL: http://www.luv.asn.au/overheads/NIS.html (6 Nov. 2000).

[3] "RFC1831 – RPC: Remote Procedure Call Protocol Specification Version 2." August 1995. URL: http://www.faqs.org/rfcs/rfc1831.html (6 Nov. 2000).

[4] "NIS Product Support Document (PSD)." Revision: 2.0. URL: http://www.ebsinc.com/solaris/network/nis.html (7 Nov. 2000).

[5] "NIS to NIS+ Transition Guide." Revision: A. December 1993. URL: http://sunsolve.sun.com/private-cgi/retrieve.pl?type=2&doc=stb/1141 (8 Nov. 2000).

[6] Galvin, Peter. "NIS+ Part 1: What's in a Name (Service)?" URL: http://www.sunworld.com/sunworldonline/swol-09-1996/swol-09-security.html (8 Nov. 2000).

[7] "Securing NIS (formerly YP)." URL: http://www.eng.auburn.edu/users/doug/nis.html (8 Nov. 2000).

[8] Retkowski, Greg. "LDAP Nameservice Howto." 20 November 1998. URL: http://www.rage.net/ldap/ldapns-howto/LdapNS-howto.html (8 Nov. 2000).