Network Working Group
Internet-Draft
26 January 1999
Expires: 26 July 1999

Keith Moore, ed.
University of Tennessee

Privacy Considerations for the Use of Hardware Serial Numbers in End-to-End Network Protocols

draft-iesg-serno-privacy-00.txt

Abstract

This memo describes privacy risks associated with the use of hardware serial numbers in network protocols, and some countermeasures which can ameliorate those risks.

1. Introduction

Some network protocols use of hardware serial numbers, or quantities derived from hardware serial numbers, as protocol elements. Such numbers are finding increasing use in network protocols to form a globally-unique identifier, either for the host itself, or for some other purpose.

Examples of hardware serial numbers include 48-bit IEEE 802 Media Access Control (MAC) addresses, 64-bit Extended Unique Identifier (EUI-64) addresses, and the serial numbers which Intel Corporation has proposed to include in some of its future CPU products. Some protocols which use components based on hardware serial numbers are IPv6 (when global addresses are obtained using Stateless Address Autoconfiguration [RFC-1971]), IPv6 over ATM [IPv6/ATM] (when the Interface Tokens are

derived from 48-bit MAC addresses, EUI-64 values, or AESA values), and WebDAV [RFC-YYYY] (which use UUIDs/GUIDs [UUIDs] which may be generated from 48-bit MAC addresses).

Use of protocols that directly or indirectly expose hardware serial numbers may compromise the privacy of a user or group of users. This memo attempts to document known risks of exposing such information, and countermeasures which might ameliorate those risks.

2. Risks of Exposing Hardware Serial Numbers

When a hardware serial number is associated with a particular host, the number may be used to track network-based activity of that host. Such tracking may be done by communication among the parties with which the host communicates, or by eavesdroppers who can observe the host's traffic. When the hardware serial number appears in an IPv6 address, the information may be available to eavesdroppers even when the higher level traffic is encrypted via IPSEC or TLS/SSL. In higher level protocols, hardware serial numbers might be transmitted through firewalls.

In many cases it is feasible for an observer to combine hardware serial numbers with information which is visible either to the host's communications peers, or to an eavesdropper (if the transmission is unencrypted). For example, if a mobile host (e.g. laptop) were used to access the net from several different locations, an eavesdropper would be able to track the movement of that mobile host (and probably also its human user) from place to place, even if the communications were encrypted.

Certain bits of a hardware serial numbers are usually reserved to identify the vendor of the hardware. Hardware serial numbers can therefore leak information about the kind of computer hardware which is being used, and of the different types of computers in use by a particular group.

Some software products themselves emit serial numbers or other registration information. If such software products are used on a host that exposes its hardware serial number, an eavesdropper can determine which copies of the software are running on a which hosts.

3. Recommendations on the Use of Hardware Serial Numbers

Protocols intended to be used over the global Internet SHOULD NOT depend on the inclusion of hardware serial numbers. Protocols intended to be used only in a local IP-based network, which use hardware serial numbers, SHOULD define a means to keep those serial numbers from escaping into the global Internet.

Implementations of protocols which use protocol elements derived from hardware serial numbers SHOULD provide users with the ability to either omit those elements entirely, or select an alternative means of deriving those protocol elements. For instance, to avoid exposure, a user might prefer to set the IPv6 address via manual configuration or DHCPv6 [DHCPv6] rather than by using stateless autoconfiguration.

Protocol elements that contain hardware serial numbers should be considered opaque to any applications that use them. Applications SHOULD NOT attempt to interpret the hardware serial number portion of such protocol elements, and MUST NOT depend on the hardware serial numbers for proper operation.

4. Countermeasures

Countermeasures should be evaluated in relation to risk. For instance, there is little additional risk in exposing the hardware address of a single stationary host that is assigned a static IP address.

Depending on the environment, there may be one or more means of instructing an operating system or application to use a different serial number in various protocols. For instance, it may be possible to set the MAC address of an ethernet card to some value other than the default.

In some environments, it may be possible to use network address translators (NATs), firewalls, or proxies to hide use of particular hosts, or make substitutions for protocol elements that contain hardware serial numbers. However, such solutions have severe limitations which are beyond the scope of this memo. [NAT-ARCH1], [NAT-ARCH2].

References

[IPv6-ATM] Greenville Arimtage, Peter Schulter, Markus Jork. IPv6 over ATM Networks. Internet-draft draft-ietf-ion-ipv6-atm-03.txt, Octo-

ber 17, 1998. (work in progress)

[IPv6-SAA] S. Thomson, T. Narten. IPv6 Stateless Address Autoconfiguration. RFC 1971, August 1996.

[NAT-ARCH1] T. Hain. Architectural Implications of NAT. Internet-draft draft-iab-nat-implications-02.txt, October 1998. (work in progress)

[NAT-ARCH2] Yakov Rekhter. Implications of NATs on the TCP/IP architecture. Internet-draft draft-ietf-nat-arch-implications-00.txt, August 1998. (work in progress)

[UUIDs] ISO (International Organization for Standardization). Information technology - Open Systems Interconnection - Remote Procedure Call (RPC). ISO/IEC 11578:1996.

[WebDAV] Y. Y. Goland, E. J. Whitehead, A. Faizi, S. R. Carter, D. Jensen. HTTP Extensions for Distributed Authoring -- WebDAV. Internet-draft draft-ietf-webdav-protocol-10.txt, November 16, 1998. (work in progress, approved by IESG for publication as an RFC)