       Considerations for IP broadcast and multicast protocol designers
               draft-winfaa-intarea-broadcast-consider-00

Abstract

   A number of application-layer protocols make use of IP broadcasts or
   multicast messages for functions such as local service discovery or
   name resolution. Some of these functions can only be implemented
   efficiently using such mechanisms. When using broadcasts or
   multicast messages, a passive observer in the same broadcast domain
   can trivially record these messages and analyze their content.
   Therefore, broadcast/multicast protocol designers need to take
   special care when designing their protocols.

Table of Contents

1. Introduction

   Broadcast and multicast messages have a large receiver group by
   design. Because of that, these two mechanisms are vital for a number
   of basic network functions such as auto-configuration. Application
   developers use broadcast/multicast messages to implement things like
   local service or peer discovery and it appears that an increasing
   number of applications make use of it.

   Using broadcast/multicast can become problematic if the information
   that is being distributed can be regarded as sensitive or when the
   information that is distributed by multiple of these protocols can be
   correlated in a way that sensitive data can be derived. This is
   clearly true for any protocol, but broadcast/multicast is special in
   two respects: a) the aforementioned large receiver group which makes
   it trivial for anybody on a LAN to collect the information without
   special privileges or a special location in the network and b)
   encryption is more difficult when broadcasting/multicasting messages.
   This draft documents a number of design considerations for broadcast/
   multicast protocol designers that are intended to reduce the
   likelihood that a broadcast protocol can be misused to collect
   sensitive data about devices, users and groups of users on a LAN.

1.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
document are to be interpreted as described in RFC 2119 [RFC2119].

2. Design considerations

There are a few obvious and a few not necessarily obvious things
designers of broadcast/multicast protocols should consider. Most of
these items are based on protocol behaviour observed as part of
experiments on operational networks.

2.1. Message frequency

Frequent broadcast/multicast traffic caused by an application can
give user behaviour and online times away. This allows a passive
observer to potentially deduce a user's current activity (e.g. a
game) and it allows to create an online profile (i.e. times the user
is on the network). The higher the frequency of these messages, the
more accurate this profile will be. Given that broadcasts are only
visible in the same broadcast domain, these messages also give the
rough location of the user away (e.g. a campus or building).

Besides the privacy implications of frequent broadcasting, it also
represents a performance problem. In particular in certain wireless
technologies such as 802.11, broadcast and multicast are transmitted
at a much lower rate (the lowest common denominator rate) compared to
unicast and therefore have a much bigger impact on the overall
available airtime. Further, it will limit the ability for devices to
go to sleep if frequent broadcasts are being sent. A similar problem
in respect to Router Advertisements is addressed in
[I-D.ietf-v6ops-reducing-ra-energy-consumption].

If a protocol relies on frequent or periodic broadcast/multicast
messages, the frequency SHOULD be chosen conservatively, in
particular if the messages contain persistent identifiers. Also,
intelligent message suppression mechanisms such as the ones employed
in mDNS [RFC6762] SHOULD be implemented.

2.2. Persistent identifiers

A few broadcast/multicast protocols observed in the wild make use of
persistent identifiers. This includes the use of hostnames or more
abstract persistent identifiers such as a UUID or similar. These IDs
e.g. identify the installation of a certain application and might
not change across updates of the software. This allows a passive
observer to track a user precisely if broadcast/multicast messages

   are frequent. This is even true in case the IP and/or MAC address
   changes. Such identifiers also allow two different interfaces (e.g.
   Wifi and Ethernet) to be correlated to the same device. If the
   application makes use of persistent identifiers for multiple
   installations of the same application for the same user, this even
   allows to infer that different devices belong to the same user.

   If a protocol relies on IDs to be transmitted, it SHOULD be
   considered if frequent ID rotations are possible in order to make
   user tracking more difficult. Persistent IDs are considered bad
   practice in general as persistent application layer IDs will make
   efforts on lower layers to randomize identifiers useless or at least
   much more difficult.

2.3. Anticipate user behaviour

   A large number of users name their device after themselves, either
   using their first name, last name or both. Often a hostname includes
   the type, model or maker of a device, its function or includes
   language specific information. Based on gathered data, this appears
   currently to be prevalent user behaviour. For protocols using the
   hostname as part of the messages, this clearly will reveal personally
   identifiable information to everyone on the local network.

   Where possible, the use of hostnames in broadcast/multicast protocols
   SHOULD be avoided. If only a persistent ID is needed, this can be
   generated. An application might want to display the information it
   will broadcast on the LAN at install/config time, so the user is at
   least aware of the application's behaviour. More hostname
   considerations can be found in [I-D.ietf-intarea-hostname-practice].

2.4. Consider potential correlation

   A large number of services and applications make use of the
   broadcast/multicast mechanism. That means there are various sources
   of information that are easily accessible by a passive observer. In
   isolation, the information these protocols reveal might seem
   harmless, but given multiple such protocols, it might be possible to
   correlate this information. E.g. a protocol that uses frequent
   messages including a UUID to identify the particular installation
   does not give the identity of the user away. But a single message
   including the user's hostname might just do that and it can be
   correlated using e.g. the MAC address of the device's interface.

   A broadcast protocol designer should be aware of the fact that even
   if the protocol's information seems harmless, there might be ways to
   correlate that information with other broadcast protocol information
   to reveal sensitive information about a user.

2.5. Configurability

   A lot of applications and services using broadcast protocols do not
   include the means to declare "safe" environments (e.g. based on the
   SSID of a WiFi network). E.g. a device connected to a public WiFi
   will likely broadcast the same information as when connected to the
   home network. It would be beneficial if certain behaviour could be
   restricted to "safe" environments.

   An application developer making use of broadcasts as part of the
   application SHOULD make the broadcast feature, if possible,
   configurable, so that potentially sensitive information does not leak
   on public networks.

3. Operational considerations

   Besides changing end-user behavior, choosing sensible defaults as an
   operating system vendor (e.g. for suggesting host names) and the
   considerations for protocol designers mentioned in this document,
   there are things that the network administrators/operators can do to
   limit the above mentioned problems.

   A feature not uncommonly found on access points e.g. is to filter
   broadcast and multicast traffic. This will potentially break certain
   applications or some of their functionality but will also protect the
   users from potentially leaking sensitive information.

4. Acknowledgements

   This work is supported by the European Commission under grant
   agreement FP7-318627 mPlane. Support does not imply endorsement.

5. IANA Considerations

   This memo includes no request to IANA.

6. Security Considerations

   TBD

7. Normative References

   [I-D.ietf-intarea-hostname-practice]
              Huitema, C. and D. Thaler, "Current Hostname Practice
              Considered Harmful", draft-ietf-intarea-hostname-
              practice-00 (work in progress), October 2015.

   [I-D.ietf-v6ops-reducing-ra-energy-consumption]
              Yourtchenko, A. and L. Colitti, "Reducing energy
              consumption of Router Advertisements", draft-ietf-v6ops-
              reducing-ra-energy-consumption-03 (work in progress),
              November 2015.

   [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate
              Requirement Levels", BCP 14, RFC 2119, March 1997.

   [RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762,
              DOI 10.17487/RFC6762, February 2013,
              <http://www.rfc-editor.org/info/rfc6762>.

Authors' Addresses

   Rolf Winter
   University of Applied Sciences Augsburg
   Augsburg
   DE

   Email: rolf.winter@hs-augsburg.de

   Michael Faath
   University of Applied Sciences Augsburg
   Augsburg
   DE

   Email: michael.faath@hs-augsburg.de

   Fabian Weisshaar
   University of Applied Sciences Augsburg
   Augsburg
   DE

   Email: fabian.weisshaar@hs-augsburg.de