Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: April 21, 2016

R. Winter
M. Faath
F. Weisshaar
University of Applied Sciences Augsburg
October 19, 2015

Considerations for IP broadcast and multicast protocol designers
draft-winfaa-broadcast-consider-01

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 2
1.1. Requirements Language 3
2. Design considerations 3
2.1. Message frequency 3
2.2. Persistent identifiers 3
2.3. Anticipate user behaviour 4
2.4. Rember - You are not alone 4
2.5. Configurability 4
3. IANA Considerations 5
4. Security Considerations 5
5. Normative References 5
Appendix A. Additional Stuff 5
Authors' Addresses 5

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 really, 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 priviledges 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 likelyhood 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 an experiment on an operational network.

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 decuct 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).

If a protocol relies on frequent or periodic broadcast/multicast messages, the frequency should be chosen conservatively, in particular if the messages contain persisten identifiers.

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 persitent 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.

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 to currently 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.

2.4. Rember - You are not alone

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. IANA Considerations

This memo includes no request to IANA.

4. Security Considerations

TBD

5. Normative References

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

Appendix A. Additional Stuff

This becomes an Appendix.

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