

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: November 30, 2017

R. Winter
University of Applied Sciences Augsburg
M. Faath
Conntac GmbH
F. Weisshaar
University of Applied Sciences Augsburg
May 29, 2017

Privacy considerations for IP broadcast and multicast protocol designers
draft-ietf-intarea-broadcast-consider-03

Abstract

A number of application-layer protocols make use of IP broadcasts or multicast messages for functions like 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/multicast domain can trivially record these messages and analyze their content. Therefore, broadcast/multicast protocol designers need to take special care when designing their protocols.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on November 30, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
1.1. Types and usage of broadcast and multicast	3
1.2. Requirements Language	4
2. Privacy considerations	4
2.1. Message frequency	5
2.2. Persistent identifiers	5
2.3. Anticipate user behavior	6
2.4. Consider potential correlation	6
2.5. Configurability	7
3. Operational considerations	8
4. Summary	8
5. Other considerations	9
6. Acknowledgments	9
7. IANA Considerations	9
8. Security Considerations	10
9. References	10
9.1. Normative References	10
9.2. Informative References	10
Authors' Addresses	12

1. Introduction

Broadcast and multicast messages have a large (and to the sender unknown) receiver group by design. Because of that, these two mechanisms are vital for a number of basic network functions such as auto-configuration or link-layer address lookup. Also 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 [TRAC2016]. That is not entirely surprising. As RFC 919 [RFC0919] puts it, "The use of broadcasts [...] is a good base for many applications". Broadcast and multicast functionality in a subnetwork are therefore important as a lack thereof renders the protocols underlying these mechanisms inoperable [RFC3819].

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 at least two respects:

- (a) The aforementioned large receiver group, consisting of receivers unknown to the sender. This makes eavesdropping without special privileges or a special location in the network trivial for anybody in the broadcast/multicast domain.
- (b) Encryption is more difficult when broadcast/multicast messages, leaving content of these messages in the clear and making it easier to spoof and replay them.

Given the above, privacy protection for protocols based on broadcast or multicast communication is significantly more difficult compared to unicast communication and at the same time invading the privacy is much easier.

Privacy considerations of IETF-specified protocols have received some attention in the recent past (e.g. RFC 7721 [RFC7721] or RFC 7919 [RFC7819]). There is also general guidance available for document authors on when and how to include a privacy considerations section in their documents and on how to evaluate the privacy implications of Internet protocols [RFC6973]. RFC6973 also describes potential threats to privacy in great detail and lists terminology that is also used in this document.

In contrast to RFC6973, this document contains a number of privacy considerations especially for broadcast/multicast protocol designers that are intended to reduce the likelihood that a broadcast/multicast protocol can be misused to collect sensitive data about devices, users and groups of users on a broadcast/multicast domain. These considerations particularly apply to protocols designed outside the IETF for two reasons. For one, non-standard protocols will likely not receive operational attention and support in making them more secure such as e.g. DHCP snooping does for DHCP because they typically are not documented. The other reason is that these protocols have been designed in isolation, where a set of considerations to follow is useful in the absence of a larger community providing feedback. In particular, carelessly designed broadcast/multicast protocols can break privacy efforts at different layers of the protocol stack such as MAC address or IP address randomization [RFC4941].

1.1. Types and usage of broadcast and multicast

In IPv4, two major types of broadcast addresses exist, the limited broadcast which is defined as all-ones (255.255.255.255, defined in section 5.3.5.1 of [RFC1812]) and the directed broadcast with the

given network prefix of an IP address and the host part of all-ones (defined in section 5.3.5.2. of [RFC1812]). Broadcast packets are received by all nodes in a subnetwork. Limited broadcasts never transit a router. The same is true for directed broadcasts by default, but routers MAY provide an option to do this [RFC2644]. IPv6 on the other hand does not provide broadcast addresses but solely relies on multicast [RFC4291].

In contrast to broadcast addresses, multicast addresses represent an identifier for a set of interfaces that can be a set different from all nodes in the subnetwork. All interfaces that are identified by a given multicast address receive packets destined towards that address and are called a multicast group. In both IPv4 and IPv6, multiple pre-defined multicast addresses exist. The ones most relevant for this document are the ones with subnet scope. For IPv4, an IP prefix is reserved for this purpose called the Local Network Control Block (224.0.0.0/24, defined in section 4 of [RFC5771]). For IPv6, the relevant multicast addresses are the two All Nodes Addresses, which every IPv6-capable host is required to recognize as identifying itself (see section 2.7.1 of [RFC4291]).

Typical usage of these addresses include local service discovery (e.g. mDNS [RFC6762] and LLmNR [RFC4795] make use of multicast), autoconfiguration (e.g. DHCPv4 [RFC2131] uses broadcasts and DHCPv6 [RFC3315] uses multicast addresses) and other vital network services such as address resolution or duplicate address detection. But besides these core network functions, also applications make use of broadcast and multicast functionality, often implementing proprietary protocols. In sum, these protocols distribute a diverse set of potentially privacy sensitive information to a large receiver group.

1.2. 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. Privacy considerations

There are a few obvious and a few not necessarily obvious things designers of broadcast/multicast protocols should consider in respect to the privacy implications of their protocol. Most of these items are based on protocol behavior observed as part of experiments on operational networks [TRAC2016].

2.1. Message frequency

Frequent broadcast/multicast traffic caused by an application can give user behavior 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/multicasts are only visible in the same broadcast/multicast domain, these messages also give the rough location of the user away (e.g. a campus or building).

This behavior has e.g. been observed by a synchronization mechanism of a popular application, where multiple messages have been sent per minute via broadcast. Given this behavior, it is possible to record a device's time on the network with a sub-minute accuracy given only the traffic of this single application installed on the device. But also services used for local name resolution in modern operating systems utilize broadcast/multicast protocols (e.g. mDNS, LLMNR or NetBIOS) to announce for example their shares regularly and allow a tracking of the online time of a device.

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 (see next subsection). Also, intelligent message suppression mechanisms such as the ones employed in mDNS [RFC6762] SHOULD be implemented. The lower the frequency of broadcast messages, the harder traffic analysis and surveillance becomes.

2.2. Persistent identifiers

A few broadcast/multicast protocols observed in the wild make use of persistent identifiers. This includes the use of host names or more abstract persistent identifiers such as a UUID or similar. These IDs, which e.g. identify the installation of a certain application might not change across updates of the software and are therefore extremely long lived. 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.

The aforementioned broadcast messages from a synchronization mechanism of a popular application also included a persistent

identifier in every broadcast. This identifier did never change after the application was installed and allowed to track a device even when it changed its network interface or when it connected to a different network.

If a broadcast/multicast 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 for broadcast and multicast communication as persistent application layer IDs will make efforts on lower layers to randomize identifiers (e.g. [I-D.huitema-6man-random-addresses]) useless or at least much more difficult.

2.3. Anticipate user behavior

A large number of users name their device after themselves, either using their first name, last name or both. Often a host name 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 behavior [TRAC2016]. For protocols using the host name as part of the messages, this clearly will reveal personally identifiable information to everyone on the local network. This information can also be used to mount more sophisticated attacks, when e.g. the owner of a device is identified (as an interesting target) or properties of the device are known (e.g. known vulnerabilities).

A popular operating system vendor includes the name the user chooses for the user account during the installation process as part of the host name of the device. The name of the operating system is also included, revealing therefore two pieces of information, which can be regarded as private information if the host name is used in broadcast/multicast messages.

Where possible, the use of host names and other user provided information 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 behavior. More host name considerations can be found in [I-D.ietf-intarea-hostname-practice]. More information on user participation can be found in RFC 6973 [RFC6973].

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 host name might just do that and it can be correlated using e.g. the MAC address of the device's interface.

In the experiments described in [TRAC2016], it was possible to correlate frequently sent broadcast messages that included a unique identifier with other broadcast/multicast messages containing usernames (e.g. mDNS, LLMNR or NetBIOS), but also relationships to other users. This allowed to reveal the real identity of the users of many devices but it also gave some information about their social environment away.

A broadcast protocol designer should be aware of the fact that even if - in isolation - the information a protocol leaks 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/multicast protocols do not include the means to declare "safe" environments (e.g. based on the SSID of a WiFi network and the MAC addresses of the access points). 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 behavior could be restricted to "safe" environments.

A popular operating system e.g. allows the user to specify the trust level of the network the device connects to, which for example restricts specific system services (using broadcast/multicast messages for their normal operation) to be used in untrusted networks. Such functionality could be implemented as part of an application.

An application developer making use of broadcasts/multicasts as part of the application SHOULD make the broadcast feature, if possible, configurable, so that potentially sensitive information does not leak on public networks, where the threat to privacy is much larger.

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. Summary

Increasingly, applications rely on broadcast and multicast messages. For some, broadcasts/multicasts are the basis of their application logic, others use broadcasts/multicasts to improve certain aspects of the application but are fully functional in case broadcasts/multicasts fail. Irrespective of the role of broadcast and multicast messages for the application, the designers of protocols that make use of them should be very careful in their protocol design because of the special nature of broad- and multicast.

It is not always possible to implement certain functionality via unicast, but in case a protocol designer chooses to rely on broadcast/multicast, the following should be carefully considered:

- o IETF-specified protocols, such as mDNS [RFC6762], SHOULD be used if possible as operational support might exist to protect against the leakage of private information. Also, for some protocols privacy extensions are being specified, which can be used if implemented. E.g. for DNS-SD privacy extensions are documented in [I-D.ietf-dnssd-privacy]
- o Avoid using user-specified information inside broadcast/multicast messages as users will often use personal information or other information aiding attackers, in particular if the user is unaware about how that information is being used
- o Avoid persistent IDs in messages as this allows user tracking, correlation and potentially has a devastating effect on other privacy protection mechanisms
- o If you really must use a broadcast/multicast protocol and cannot use an IETF-specified protocol, then:

- * Be very conservative in how frequently you send messages as an effort in data minimization
- * Seek advice from IETF-specified protocols such as message suppression in mDNS
- * Try to design the protocol in a way that the information cannot be correlated with other information in broadcast/multicast messages
- * Let the user configure safe environments if possible (e.g. based on the SSID)

[Note: discussions on this document should be take place on the Intarea mailing list of the IETF. Subscription: <https://www.ietf.org/mailman/listinfo/int-area>, Mailing list archive: <https://www.ietf.org/mail-archive/web/int-area/current/maillist.html>]

5. Other considerations

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]. In that respect broadcasts can be used for another class of attacks that not related to privacy. The potential impact on network performance should nevertheless be considered by broadcast protocol designers.

6. Acknowledgments

We would like to thank Eliot Lear, Joe Touch and Stephane Bortzmeyer for their valuable input to this document.

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

7. IANA Considerations

This memo includes no request to IANA.

8. Security Considerations

This document deals with privacy-related considerations of broadcast- and multicast-based protocols. It contains advice for designers of such protocols to minimize the leakage of privacy-sensitive information. The intent of the advice is to make sure that identities will remain anonymous and user tracking will be made difficult.

It should be noted that certain applications could make use of existing mechanisms to protect multicast traffic such as the ones defined in [RFC5374]. Examples of such applications can be found in Appendix A. of [RFC5374]. Given the required infrastructure and assumptions about these applications and the security infrastructure, many applications will not be able to make use of such mechanisms.

9. References

9.1. Normative References

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

9.2. Informative References

[I-D.huitema-6man-random-addresses]
Huitema, C., "Implications of Randomized Link Layers Addresses for IPv6 Address Assignment", draft-huitema-6man-random-addresses-03 (work in progress), March 2016.

[I-D.ietf-dnssd-privacy]
Huitema, C. and D. Kaiser, "Privacy Extensions for DNS-SD", draft-ietf-dnssd-privacy-00 (work in progress), October 2016.

[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.

- [RFC0919] Mogul, J., "Broadcasting Internet Datagrams", STD 5, RFC 919, DOI 10.17487/RFC0919, October 1984, <<http://www.rfc-editor.org/info/rfc919>>.
- [RFC1812] Baker, F., Ed., "Requirements for IP Version 4 Routers", RFC 1812, DOI 10.17487/RFC1812, June 1995, <<http://www.rfc-editor.org/info/rfc1812>>.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, DOI 10.17487/RFC2131, March 1997, <<http://www.rfc-editor.org/info/rfc2131>>.
- [RFC2644] Senie, D., "Changing the Default for Directed Broadcasts in Routers", BCP 34, RFC 2644, DOI 10.17487/RFC2644, August 1999, <<http://www.rfc-editor.org/info/rfc2644>>.
- [RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July 2003, <<http://www.rfc-editor.org/info/rfc3315>>.
- [RFC3819] Karn, P., Ed., Bormann, C., Fairhurst, G., Grossman, D., Ludwig, R., Mahdavi, J., Montenegro, G., Touch, J., and L. Wood, "Advice for Internet Subnetwork Designers", BCP 89, RFC 3819, DOI 10.17487/RFC3819, July 2004, <<http://www.rfc-editor.org/info/rfc3819>>.
- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", RFC 4291, DOI 10.17487/RFC4291, February 2006, <<http://www.rfc-editor.org/info/rfc4291>>.
- [RFC4795] Aboba, B., Thaler, D., and L. Esibov, "Link-local Multicast Name Resolution (LLMNR)", RFC 4795, DOI 10.17487/RFC4795, January 2007, <<http://www.rfc-editor.org/info/rfc4795>>.
- [RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007, <<http://www.rfc-editor.org/info/rfc4941>>.
- [RFC5374] Weis, B., Gross, G., and D. Ignjatic, "Multicast Extensions to the Security Architecture for the Internet Protocol", RFC 5374, DOI 10.17487/RFC5374, November 2008, <<http://www.rfc-editor.org/info/rfc5374>>.

- [RFC5771] Cotton, M., Vegoda, L., and D. Meyer, "IANA Guidelines for IPv4 Multicast Address Assignments", BCP 51, RFC 5771, DOI 10.17487/RFC5771, March 2010, <<http://www.rfc-editor.org/info/rfc5771>>.
- [RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <<http://www.rfc-editor.org/info/rfc6762>>.
- [RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, DOI 10.17487/RFC6973, July 2013, <<http://www.rfc-editor.org/info/rfc6973>>.
- [RFC7721] Cooper, A., Gont, F., and D. Thaler, "Security and Privacy Considerations for IPv6 Address Generation Mechanisms", RFC 7721, DOI 10.17487/RFC7721, March 2016, <<http://www.rfc-editor.org/info/rfc7721>>.
- [RFC7819] Jiang, S., Krishnan, S., and T. Mrugalski, "Privacy Considerations for DHCP", RFC 7819, DOI 10.17487/RFC7819, April 2016, <<http://www.rfc-editor.org/info/rfc7819>>.
- [TRAC2016] Faath, M., Weisshaar, F., and R. Winter, "How Broadcast Data Reveals Your Identity and Social Graph", 7th International Workshop on TRaffic Analysis and Characterization IEEE TRAC 2016, September 2016.

Authors' Addresses

Rolf Winter
University of Applied Sciences Augsburg
Augsburg
DE

Email: rolf.winter@hs-augsburg.de

Michael Faath
Conntac GmbH
Augsburg
DE

Email: faath@conntac.net

Fabian Weisshaar
University of Applied Sciences Augsburg
Augsburg
DE

Email: fabian.weisshaar@hs-augsburg.de

Network Working Group
Internet-Draft
Intended status: Informational
Expires: August 7, 2017

C. Huitema
Private Octopus Inc.
D. Thaler
Microsoft
R. Winter
University of Applied Sciences Augsburg
February 3, 2017

Current Hostname Practice Considered Harmful
draft-ietf-intarea-hostname-practice-05.txt

Abstract

Giving a hostname to your computer and publishing it as you roam from one network to another is the Internet equivalent of walking around with a name tag affixed to your lapel. This current practice can significantly compromise your privacy, and something should change in order to mitigate these privacy threats.

There are several possible remedies, such as fixing a variety of protocols or avoiding disclosing a hostname at all. This document describes some of the protocols that reveal hostnames today and sketches another possible remedy, which is to replace static hostnames by frequently changing randomized values.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on August 7, 2017.

Copyright Notice

Copyright (c) 2017 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction	2
2. Naming Practices	3
3. Partial Identifiers	4
4. Protocols that leak Hostnames	4
4.1. DHCP	5
4.2. DNS Address to Name Resolution	5
4.3. Multicast DNS	5
4.4. Link-local Multicast Name Resolution	6
4.5. DNS-Based Service Discovery	6
4.6. NetBIOS-over-TCP	7
5. Randomized Hostnames as Remedy	7
6. Security Considerations	8
7. IANA Considerations	9
8. Acknowledgments	9
9. Informative References	9
Authors' Addresses	11

1. Introduction

There is a long established practice of giving names to computers. In the Internet protocols, these names are referred to as "hostnames" [RFC7719]. Hostnames are normally used in conjunction with a domain name suffix to build the "Fully Qualified Domain Name" (FQDN) of a host [RFC1983]. However, it is common practice to use the hostname without further qualification in a variety of applications from file sharing to network management. Hostnames are typically published as part of domain names, and can be obtained through a variety of name lookup and discovery protocols.

Hostnames have to be unique within the domain in which they are created and used. They do not have to be globally unique identifiers, but they will always be at least partial identifiers, as discussed in Section 3.

The disclosure of information through hostnames creates a problem for mobile devices. Adversaries that monitor a remote network such as a

Wi-Fi hot spot can obtain the hostname through passive monitoring or active probing of a variety of Internet protocols, such as for example DHCP, or multicast DNS (mDNS). They can correlate the hostname with various other information extracted from traffic analysis and other information sources, and can potentially identify the device, device properties and its user [TRAC2016].

2. Naming Practices

There are many reasons to give names to computers. This is particularly true when computers operate on a network. Operating systems like Microsoft Windows or Unix assume that computers have a "hostname." This enables users and administrators to do things such as ping a computer, add its name to an access control list, remotely mount a computer disk, or connect to the computer through tools such as telnet or remote desktop. Other operating systems maintain multiple hostnames for different purposes, e.g. for use with certain protocols such as mDNS.

In most consumer networks, naming is pretty much left to the fancy of the user. Some will pick names of planets or stars, other names of fruits or flowers, and other will pick whatever suits their mood when they unwrap the device. As long as users are careful to not pick a name already in use on the same network, anything goes. Very often however, the operating system is suggesting a hostname at install time, which can contain the user name, the login name and information learned from the device itself such as the brand, model or maker of the device [TRAC2016].

In large organizations, collisions are more likely and a more structured approach is necessary. In theory, organizations could use multiple DNS subdomains to ease the pressure on uniqueness, but in practice many don't and insist on unique flat names, if only to simplify network management. To ensure unique names, organizations will set naming guidelines and enforce some kind of structured naming. For example, within the Microsoft corporate network, computer names are derived from the login name of the main user, leading to names like "huitema-test2" for a machine that one of the authors used to test software.

There is less pressure to assign names to small devices, including for example smart phones, as these devices typically do not enable sharing of their disks or remote login. As a consequence, these devices often have manufacturer assigned names, which vary from very generic like "Windows Phone" to completely unique like "BrandX-123456-7890-abcdef" and often contain the name of the device owner, the device's brand name, and often also a hint as to which language the device owner speaks [TRAC2016].

3. Partial Identifiers

Suppose an adversary wants to track the people connecting to a specific Wi-Fi hot spot, for example in a railroad station. Assume that the adversary is able to retrieve the hostname used by a specific laptop. That, in itself, might not be enough to identify the laptop's owner. Suppose however that the adversary observes that the laptop name is "dthaler-laptop" and that the laptop has established a VPN connection to the Microsoft corporate network. The two pieces of information, put together, firmly point to Dave Thaler, employed by Microsoft. The identification is successful.

In the example, we saw a login name inside the hostname, and that certainly helped identification. But generic names like "jupiter" or "rosebud" also provide partial identification, especially if the adversary is capable of maintaining a database recording, among other information, the hostnames of devices used by specific users. Generic names are picked from vocabularies that include thousands of potential choices. Finding the name reduces the scope of the search significantly. Other information such as the visited sites will quickly complement that data and can lead to user identification.

Also the special circumstances of the network can play a role. Experiments on operational networks such as the IETF meeting network have shown that with the help of external data such as the publicly available IETF attendees list or other data sources such as LDAP servers on the network [TRAC2016], the identification of the device owner can become trivial given only partial identifiers in a hostname.

Unique names assigned by manufacturers do not directly encode a user identifier, but they have the property of being stable and unique to the device in a large context. A unique name like "BrandX-123456-7890-abcdef" allows efficient tracking across multiple domains. In theory, this only allows tracking of the device but not of the user. However, an adversary could correlate the device to the user through other means, for example the one-time capture of some clear text traffic. Adversaries could then maintain databases linking unique host name to user identity. This will allow efficient tracking of both the user and the device.

4. Protocols that leak Hostnames

Many IETF protocols can leak the "hostname" of a computer. A non exhaustive list includes DHCP, DNS address to name resolution, Multicast DNS, Link-local Multicast Name Resolution, and DNS service discovery.

4.1. DHCP

Shortly after connecting to a new network, a host can use DHCP [RFC2131] to acquire an IPv4 address and other parameters [RFC2132]. A DHCP query can disclose the "hostname." DHCP traffic is sent to the broadcast address and can be easily monitored, enabling adversaries to discover the hostname associated with a computer visiting a particular network. DHCPv6 [RFC3315] shares similar issues.

The problems with the hostname and FQDN parameters in DHCP are analyzed in [RFC7819] and [RFC7824]. Possible mitigations are described in [RFC7844].

4.2. DNS Address to Name Resolution

The domain name service design [RFC1035] includes the specification of the special domain "in-addr.arpa" for resolving the name of the computer using a particular IPv4 address, using the PTR format defined in [RFC1033]. A similar domain, "ip6.arpa", is defined in [RFC3596] for finding the name of a computer using a specific IPv6 address.

Adversaries who observe a particular address in use on a specific network can try to retrieve the PTR record associated with that address, and thus the hostname of the computer, or even the fully qualified domain name of that computer. The retrieval may not be useful in many IPv4 networks due to the prevalence of NAT, but it could work in IPv6 networks. Other name lookup mechanisms, such as [RFC4620], share similar issues.

4.3. Multicast DNS

Multicast DNS (mDNS) is defined in [RFC6762]. It enables hosts to send DNS queries over multicast, and to elicit responses from hosts participating in the service.

If an adversary suspects that a particular host is present on a network, the adversary can send mDNS requests to find, for example, the A or AAAA records associated with the hostname in the ".local" domain. A positive reply will confirm the presence of the host.

When a new responder starts, it must send a set of multicast queries to verify that the name that it advertises is unique on the network, and also to populate the caches of other mDNS hosts. Adversaries can monitor this traffic and discover the hostname of computers as they join the monitored network.

mDNS further allows to send queries via unicast to port 5353. An adversary might decide to use unicast instead of multicast in order to hide from e.g. intrusion detection systems.

4.4. Link-local Multicast Name Resolution

Link-local Multicast Name Resolution (LLMNR) is defined in [RFC4795]. The specification did not achieve consensus as an IETF standard, but it is widely deployed. Like mDNS, it enables hosts to send DNS queries over multicast, and to elicit responses from computers implementing the LLMNR service.

Like mDNS, LLMNR can be used by adversaries to confirm the presence of a specific host on a network, by issuing a multicast request to find the A or AAAA records associated with the hostname in the ".local" domain.

When an LLMNR responder starts, it sends a set of multicast queries to verify that the name that it advertises is unique on the network. Adversaries can monitor this traffic and discover the hostname of computers as they join the monitored network.

4.5. DNS-Based Service Discovery

DNS-Based Service Discovery (DNS-SD) is described in [RFC6763]. It enables participating hosts to retrieve the location of services proposed by other hosts. It can be used with DNS servers, or in conjunction with mDNS in a server-less environment.

Participating hosts publish a service described by an "instance name", typically chosen by the user responsible for the publication. While this is obviously an active disclosure of information, privacy aspects can be mitigated by user control. Services should only be published when deciding to do so, and the information disclosed in the service name should be well under the control of the device's owner.

In theory there should not be any privacy issue, but in practice the publication of a service also forces the publication of the hostname, due to a chain of dependencies. The service name is used to publish a PTR record announcing the service. The PTR record typically points to the service name in the local domain. The service names, in turn, are used to publish TXT records describing service parameters, and SRV records describing the service location.

SRV records are described in [RFC2782]. Each record contains 4 parameters: priority, weight, port number and hostname. While the

service name published in the PTR record is chosen by the user, the "hostname" in the SRV record is indeed the hostname of the device.

Adversaries can monitor the mDNS traffic associated with DNS-SD and retrieve the hostname of computers advertising any service with DNS-SD.

4.6. NetBIOS-over-TCP

Amongst other things, NetBIOS-over-TCP ([RFC1002]) implements a name registration and resolution mechanism called the NetBIOS Name Service. In practice, NetBIOS resource names are often based on hostnames.

NetBIOS allows an application to register resource names and to resolve such names to IP addresses. In environments without an NetBIOS Name Server, the protocol makes extensive use of broadcasts from which resource names can be easily extracted. NetBIOS also allows querying for the names registered by a node directly (node status).

5. Randomized Hostnames as Remedy

There are several ways to remedy the hostname practices. We could instruct people to just turn off any protocol that leaks hostnames, at least when they visit some "insecure" place. We could also examine each particular standard that publishes hostnames, and somehow fix the corresponding protocols. Or, we could attempt to revise the way devices manage the hostname parameter.

There is a lot of merit in "turning off unneeded protocols when visiting insecure places." This amounts to attack surface reduction, and is clearly beneficial -- this is an advantage of the stealth mode defined in [RFC7288]. However, there are two issues with this advice. First, it relies on recognizing which networks are secure or insecure. This is hard to automate, but relying on end-user judgment may not always provide good results. Second, some protocols such as DHCP cannot be turned off without losing connectivity, which limits the value of this option. Also, the services that rely on protocols that leak hostnames such as mDNS will not be available when switched off. In addition, not always are hostname-leaking protocols well-known as they might be proprietary and come with an installed application instead of being provided by the operating system.

It may be possible in many cases to examine a protocol and prevent it from leaking hostnames. This is for example what is attempted for DHCP in [RFC7844]. However, it is unclear that we can identify,

revisit and fix all the protocols that publish hostnames. In particular, this is impossible for proprietary protocols.

We may be able to mitigate most of the effects of hostname leakage by revisiting the way platforms handle hostnames. This is in a way similar to the approach of MAC address randomization described in [RFC7844]. Let's assume that the operating system, at the time of connecting to a new network, picks a random hostname and starts publicizing that random name in protocols such as DHCP or mDNS, instead of the static value. This will render monitoring and identification of users by adversaries much more difficult, without preventing protocols such as DNS-SD from operating as expected. This has of course implications on the applications making use of such protocols e.g. when the hostname is being displayed to users of the application. They will not as easily be able to identify e.g. network shares or services based on the hostname carried in the underlying protocols. Also, the generation of new hostnames should be synchronized with the change of other tokens used in network protocols such as the MAC or IP address to prevent correlation of this information. E.g. if the IP address changes but the hostname stays the same, the new IP address can be correlated to belong to the same device based on a leaked hostname.

Some operating systems, including Windows, support "per network" hostnames, but some other operating systems only support "global" hostnames. In that case, changing the hostname may be difficult if the host is multi-homed, as the same name will be used on several networks. Other operating systems already use potentially different hostnames for different purposes, which might be a good model to combine both static hostnames and randomized hostnames based on their potential use and threat to a user's privacy.

Obviously, further studies are required before the idea of randomized hostnames can be implemented.

6. Security Considerations

This draft does not introduce any new protocol. It does point to potential privacy issues in a set of existing protocols.

There are obvious privacy gains to changing to randomized hostnames and also to change these names frequently. Wide deployment might however affect security functions or current practices. For example, incident response using hostnames to track the source of traffic might be affected. It is common practice to include hostnames and reverse lookup information at various times during an investigation.

7. IANA Considerations

This draft does not require any IANA action.

8. Acknowledgments

Thanks to the members of the INTAREA Working Group for discussions and reviews.

9. Informative References

- [RFC1002] NetBIOS Working Group in the Defense Advanced Research Projects Agency, Internet Activities Board, and End-to-End Services Task Force, "Protocol standard for a NetBIOS service on a TCP/UDP transport: Detailed specifications", STD 19, RFC 1002, DOI 10.17487/RFC1002, March 1987, <<http://www.rfc-editor.org/info/rfc1002>>.
- [RFC1033] Lottor, M., "Domain Administrators Operations Guide", RFC 1033, DOI 10.17487/RFC1033, November 1987, <<http://www.rfc-editor.org/info/rfc1033>>.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, DOI 10.17487/RFC1035, November 1987, <<http://www.rfc-editor.org/info/rfc1035>>.
- [RFC1983] Malkin, G., Ed., "Internet Users' Glossary", FYI 18, RFC 1983, DOI 10.17487/RFC1983, August 1996, <<http://www.rfc-editor.org/info/rfc1983>>.
- [RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, DOI 10.17487/RFC2131, March 1997, <<http://www.rfc-editor.org/info/rfc2131>>.
- [RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, DOI 10.17487/RFC2132, March 1997, <<http://www.rfc-editor.org/info/rfc2132>>.
- [RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, DOI 10.17487/RFC2782, February 2000, <<http://www.rfc-editor.org/info/rfc2782>>.
- [RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, DOI 10.17487/RFC3315, July 2003, <<http://www.rfc-editor.org/info/rfc3315>>.

- [RFC3596] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", RFC 3596, DOI 10.17487/RFC3596, October 2003, <<http://www.rfc-editor.org/info/rfc3596>>.
- [RFC4620] Crawford, M. and B. Haberman, Ed., "IPv6 Node Information Queries", RFC 4620, DOI 10.17487/RFC4620, August 2006, <<http://www.rfc-editor.org/info/rfc4620>>.
- [RFC4795] Aboba, B., Thaler, D., and L. Esibov, "Link-local Multicast Name Resolution (LLMNR)", RFC 4795, DOI 10.17487/RFC4795, January 2007, <<http://www.rfc-editor.org/info/rfc4795>>.
- [RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, DOI 10.17487/RFC6762, February 2013, <<http://www.rfc-editor.org/info/rfc6762>>.
- [RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, DOI 10.17487/RFC6763, February 2013, <<http://www.rfc-editor.org/info/rfc6763>>.
- [RFC7288] Thaler, D., "Reflections on Host Firewalls", RFC 7288, DOI 10.17487/RFC7288, June 2014, <<http://www.rfc-editor.org/info/rfc7288>>.
- [RFC7719] Hoffman, P., Sullivan, A., and K. Fujiwara, "DNS Terminology", RFC 7719, DOI 10.17487/RFC7719, December 2015, <<http://www.rfc-editor.org/info/rfc7719>>.
- [RFC7819] Jiang, S., Krishnan, S., and T. Mrugalski, "Privacy Considerations for DHCP", RFC 7819, DOI 10.17487/RFC7819, April 2016, <<http://www.rfc-editor.org/info/rfc7819>>.
- [RFC7824] Krishnan, S., Mrugalski, T., and S. Jiang, "Privacy Considerations for DHCPv6", RFC 7824, DOI 10.17487/RFC7824, May 2016, <<http://www.rfc-editor.org/info/rfc7824>>.
- [RFC7844] Huitema, C., Mrugalski, T., and S. Krishnan, "Anonymity Profiles for DHCP Clients", RFC 7844, DOI 10.17487/RFC7844, May 2016, <<http://www.rfc-editor.org/info/rfc7844>>.

[TRAC2016]

Faath, M., Weisshaar, F., and R. Winter, "How Broadcast Data Reveals Your Identity and Social Graph", 7th International Workshop on TRaffic Analysis and Characterization IEEE TRAC 2016, September 2016.

Authors' Addresses

Christian Huitema
Private Octopus Inc.
Friday Harbor, WA 98250
U.S.A.

Email: huitema@huitema.net

Dave Thaler
Microsoft
Redmond, WA 98052
U.S.A.

Email: dthaler@microsoft.com

Rolf Winter
University of Applied Sciences Augsburg
Augsburg
DE

Email: rolf.winter@hs-augsburg.de