Network Working Group C. Huitema
Internet-Draft D. Thaler
Intended status: Informational Microsoft
Expires: September 10, 2015 March 9, 2015

Current Hostname Practice Considered Harmful
draft-huitema-privsec-harmfulname-00.txt

Abstract

Giving a hostname to your computer and publishing it as you roam from network to hot spot is the Internet equivalent of walking around with a name tag affixed to your lapel. The practice can significantly compromise your privacy, and should stop.

There are several possible remedies, such as fixing a variety of protocols or avoiding disclosing a hostname at all. This document studies another possible remedy, which is to replace the static hostnames by frequently changing randomized values. This idea obviously needs more work.

Table of Contents

1. Introduction

There is a long established practice of giving names to computers. In the Internet protocols, these names are referred to as "hostnames." hostnames are normally used in conjunction with a domain name prefix to build the "Fully Qualified Domain Name" (FQDN) of a host. 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 lookups 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 or active monitoring of a variety of Internet protocols, such as for example DHCP, or multicast DNS. They can correlate the hostname with various other information extracted from traffic analysis, and identify the device and its user.

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 enable 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.

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.

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 uses 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."

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, is not enough to identify the laptop’s owner. Suppose however that the adversary observes that the laptop name is "huitema-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 Christian Huitema, 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 by maybe a factor of a thousand. Other information such as the visited sites will quickly complement that data and lead to user identification.

Of course, unique names assigned by manufacturers are even more interesting for such adversaries capable of maintaining a database recording the hostnames of devices used by specific user. With a unique name like "BrandX-123456-7890-abcdef" identification can be pretty much immediate.

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 multicast addresses 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 hostnames and FQDN parameters in DHCP are analyzed in [I-D.ietf-dhc-dhcp-privacy] and [I-D.ietf-dhc-dhcpv6-privacy]. Possible mitigations are described in [I-D.huitema-dhc-anonymity-profile].

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.

4.3. Multicast DNS

Multicast DNS (MDNS) is defined in [RFC6762]. It enables hosts to send DNS queries over a multicast port, 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.

4.4. Link-local Multicast Name Resolution

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

Like MDNS, LLMNR can be used by adversaries to confirm the presence on a network of a specific host, by issuing a multicast requests 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 service discovery

DNS-Based Service discovery (DNS-SD) is described in [RFC6763]. It enables participating host 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 host name of computers advertising any service with DNS-SD.

5. Randomized Host Names 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 our 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.

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 [I-D.huitema-dhc-anonymity-profile]. However, it is unclear that we can identify, revisit an fix all the protocols that publish hostnames.

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 [I-D.huitema-dhc-anonymity-profile]. Let's assume that the operating system, at the time of connecting to a new network, picks a random hostname and start publicizing that random name in protocols such as DHCP or MDNS, instead of the static value. This will frustrate monitoring by adversaries, without preventing protocols such as DNS SD from operating as expected.

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. 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.

7. IANA Considerations

This draft does not require any IANA action.

8. Acknowledgments

Contributions will be gladly acknowledged.

9. Informative References

[I-D.huitema-dhc-anonymity-profile]
Huitema, C., "Anonymity profile for DHCP clients", draft-huitema-dhc-anonymity-profile-01 (work in progress), March 2015.

[I-D.ietf-dhc-dhcp-privacy]
Jiang, S., Krishnan, S., and T. Mrugalski, "Privacy considerations for DHCP", draft-ietf-dhc-dhcp-privacy-00 (work in progress), February 2015.

[I-D.ietf-dhc-dhcpv6-privacy]
Krishnan, S., Mrugalski, T., and S. Jiang, "Privacy considerations for DHCPv6", draft-ietf-dhc-dhcpv6-privacy-00 (work in progress), February 2015.

[RFC1033] Lottor, M., "Domain administrators operations guide", RFC 1033, November 1987.

[RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, RFC 1035, November 1987.

[RFC2131] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, March 1997.

[RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, March 1997.

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

[RFC3315] Droms, R., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003.

[RFC3596] Thomson, S., Huitema, C., Ksinant, V., and M. Souissi, "DNS Extensions to Support IP Version 6", RFC 3596, October 2003.

[RFC4795] Aboba, B., Thaler, D., and L. Esibov, "Link-local Multicast Name Resolution (LLMNR)", RFC 4795, January 2007.

[RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, February 2013.

[RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, February 2013.

[RFC7288] Thaler, D., "Reflections on Host Firewalls", RFC 7288, June 2014.

Authors' Addresses

Christian Huitema
Microsoft
Redmond, WA 98052
U.S.A.

Email: huitema@microsoft.com

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

Email: dthaler@microsoft.com