USER IDENTIFICATION IN IPV6 NETWORK

MATĚJ GRÉGR, TOMÁŠ PODERMAŃSKI, MIROSLAV ŠVÉDA

Users in IPv4 networks typically use only one IP address per interface configured either statically or dynamically via DHCPv4 server. Several techniques can be used to detect violation of that policy. However, IPv6 protocol brings new techniques and possibilities to obtain an IPv6 address. New concepts – autoconfiguration, multiple IPv6 addresses per interface or temporary IPv6 addresses providing privacy for end users introduce new challenges for users identification. Network administrators have to collect additional information for user identification from more sources, e.g. DHCPv6 log, routers neighbor cache, Radius logs, syslog etc. This paper presents analysis of IPv6 address assignment used in current networks together with guidelines how to identify a user in IPv6 networks.

IPv6, address assignment, user identification

1. Introduction

IPv4 address configuration is mainly based on two methods. Manual address configuration or dynamical configuration via DHCPv4 server. Dynamic configuration via DHCPv4 became the de-facto standard for IPv4 address assignement. Network administrator usually bounds user's MAC address (network-card link-address) to user's IPv4 address in DHCPv4 server configuration. The IPv4 address is than assigned by DHCPv4 server to the user with corresponding MAC address. This allows to the administrator to track malevolent users because there is knowledge which IP address belongs to which MAC address and thus the user. This basically means, that the IPv4 address uniquely identify a user.

IPv6 (Internet Protocol version 6) is a new version of the fundamental Internet Protocol. IPv6 support is available for all operating systems such as Unix, Mac OS, Windows and usually enabled by default. However, address assignement in IPv6 networks is different. New concepts – autoconfiguration, multiple IPv6 addresses per interface or temporary IPv6 addresses introduce new challenges for users' identification. In IPv6 networks, IPv6 address no longer identifies a user.

The following sections describe the address configuration process in IPv6 networks and problems connected with user identification.

2. Autoconfiguration and temporary addresses

The original idea of autoconfiguration was based on the notion of an IPv6 device connecting to a network and autoconfiguring everything automatically, without requiring any interaction from the user. Similar idea exists also in IPv4 network [1], however was not widely deployed.

Stateless Address Autoconfiguration (SLAAC) [2] can be described in simple terms. The network router tells all the connected nodes in a network segment what network they appear in, and what router they should use for packets travelling to the Internet (message RA – Router Advertisement). Of course, announcing alone would not be flexible enough. Hence, a newly connected device may send a request to the network (message RS – Router Solicitation) asking for information about what network it is in, and what is the way out. The whole autoconfiguration mechanism is a part of Neighbor Discovery for IPv6 [4], and all communication takes place using the ICMPv6 protocol. End nodes now have information, which network prefix should they used and how to route packets. However, this information is insufficient. The host still needs to know, how to create a host part (end user identifier) of his IPv6 address. The host part of the IPv6 address can be derived from information that the host already has, such as, a network-card link-address. This creates a mechanism to define the host part of the network address via a modified EUI 64 algorithm.

Because of user privacy, IPv6 addresses with randomly generated 64-bits interface identifiers are preferred instead of IEEE EUI-64. The RFC 4941 standard [3] defines a way to generate and change temporary addresses. The important requirement is that the sequence of temporarily generated addresses on the interface must be totally unpredictable. However, this requirement contradicts the need to identify a malevolent user in local networks. Private, temporary addresses hinder the unique identification of users/hosts connecting to a service. This affects logging and prevents administrators from effectively tracking which users are accessing what services. Figure 1 shows addresses used by one computer during one week. The computer communicates usually by more than one IP address at the same time.

Another problem connected with IPv6 address autoconfiguration is lack of necessary information providing to clients. In order to have complete communication in the network, other details are required, such as, the IPv6 addresses of recursive DNS servers. However, this information is not in the Router Advertisement. In practice, the efforts to resolve this problem have taken three different routes.

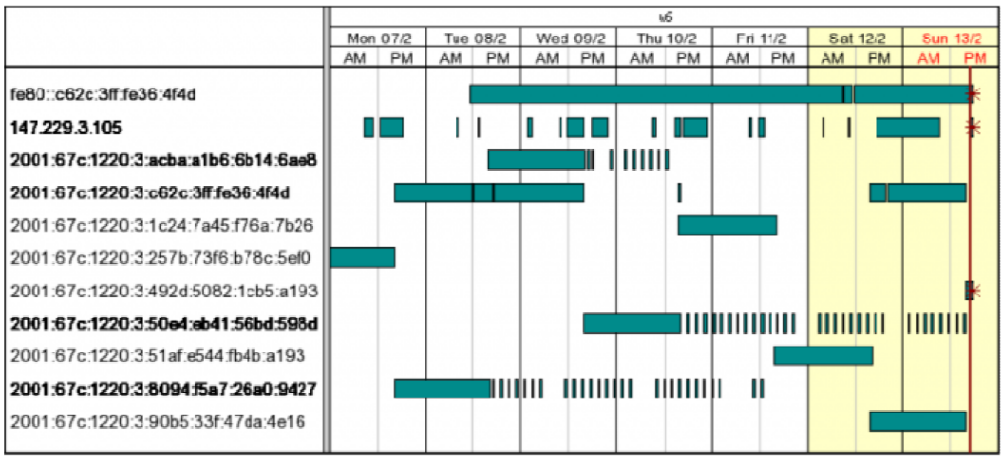

Figure 1 Addresses used by one computer during one week

Adding recursive DNS server information to the information transmitted within SLAAC. The standard suggests the addition of two items to Router Advertisement messages, namely recursive DNS server addresses and Domain Search List. So far, this support has been implemented in the RA daemon tool for Linux/UNIX (radvd[1]), but it is not supported in other current systems - both routers and client systems. At the moment, it is very difficult to estimate how willing manufacturers would be to implement this extension to their systems. SLAAC is usually processed at the OS kernel level, and expansion to other items would necessarily mean changing it directly. We probably cannot expect support for this addition during a normal update.

Using anycast addresses for recursive DNS servers. The client would send the translation request to this anycast address and the nearest Recursive DNS Server would provide an answer. Because the specification used Site-Local addresses, which were deprecated by RFC 3879[2] in 2004, this proposal was abandoned. An implementation can be found on Microsoft systems.

Using different protocol. A third proposed solution was the transmission of Recursive DNS Server addresses, and perhaps other parameters, with a different protocol, independent of the SLAAC mechanism. Quite logically, there is an opportunity to use something already known, and that is **DHCP**.

[1] http://www.litech.org/radvd/
[2] http://tools.ietf.org/html/rfc3879

3. DHCPv6

Assigning addresses with a DHCP server became the de-facto standard for IPv4. There has been an effort to re-use this mechanism in the IPv6 world. However, contrary to what one might expect, DHCPv6 is not merely DHCP that has been adapted to IPv6, with mostly the same functions.

DHCPv6 features two basic modes. In practice, the first mode, Stateless DHCPv6, is only a layer on top of the autoconfiguration mechanism described above (SLAAC) and is used to provide only recursive DNS server addresses. Two special flags are used for this purpose in the router advertisement: *M* – managed, *O* – other. These tell the client that it should ask in the relevant network for more information related to the connection parameters, through DHCPv6. If the *M* flag is set, stateful DHCPv6 is used. If the *O* flag is set, SLAAC will most likely be combined with stateless DHCPv6. If both flags are reset, the end-user stations know that there is no DHCPv6 server available in the network.

The behaviour of stateful DHCPv6 is more like the behaviour of DHCP that is known from IPv4. The server assigns an address to the client for a definite period, and the assignment is confirmed. It would seem that the SLAAC mechanism could be completely de-activated, and everything would depend on DHCPv6 alone. This idea is certainly right - except for one detail. All of the required parameters can be transferred through DHCPv6 except the most important one, which is the default gateway. The client expects to receive this information only via the Router Advertisement. This means that the client can create "uncontrolled" addresses, either with EUI 64 or Privacy Extensions. This behaviour could be suppressed by setting (or resetting) the *Autonomous* option in the Router Advertisements.

If an administrator decides to assign addresses through DHCPv6 and would like to use same functionality as with DHCPv4 server (MAC to IPv6 address binding) he faces to a problem. DHCPv6 does not use a MAC address to identify the client; instead, it uses a specially created unique identifier called a DUID (DHCP Unique Identifier) [5]. The main idea behind creating such an identifier is to free the clients from dependence on hardware and on a specific network interface. The advantage is that a change of network adapter or a connection through another interface (such as WiFi instead of Ethernet) would not mean that the end-user station would start to behave as "someone else". The standard defines three ways to obtain a DUID. The creator of the DHCPv6 client decides which one he chooses to use. In practice, this means that each system creates a DUID in a different way. If a PC has *multiboot*, with more than one operating system, then each system will have a different DUID. Most likely, the DUID will also change after

reinstallation of the operating system. To use DHCPv6 in a network, while retaining a sufficient overview of who has which address, there is no other solution than to create completely different mechanisms and methods to register clients and end-user stations.

4. User identification and long term monitoring

Long-term network monitoring, accounting and backtracking of security incidents is often achieved in IPv4 networks using NetFlow probes and collectors. This can be a problem if IPv6 is deployed and privacy extensions are allowed in the network. Same user can than communicate with different addresses. That means that address cannot be used as a unique identifier anymore. As the part of deploying IPv6 we tried to develop extension to existing monitoring systems to allow easier tracking users in an IPv6 network.

The main idea of the extension is collecting and putting together data obtained from differed parts of the network. A neighbor cache database on routers and forwarding databases on switches can provide to us information about relation between IPv6 address port on switch and a MAC address used by user. In the next step a MAC address can be used for identifying user in the database provided by radius server. All of these pieces of information, together, provide a complex view of the network and can help to identify a host. A tuple *(IPv6 address, MAC address, Login name)* is sufficient to identify a host/user. In practice, an extended tuple is built: (*Timestamp, IPv6 address, MAC address, Switch port, Login)* Timestamp is added to provide a history of communication. Switch port is necessary if the user is blocked or if an unregistered MAC address is used on some port. In addition to these values, the VLAN number and interface statistics are stored; however, these data are not necessary for host identification. Data are collected using the SNMP protocol and stored in the central database where the network administrator can search data using the IPv6, IPv4 or MAC addresses as keys.

The time dependency of the gathering of different data is crucial when accessing the ND Cache. This temporary memory at the router stores information needed to build the link between the IPv6 address and the MAC address. Because IPv6 addresses change in time and have limited validity, if the ND entry is lost, there is no way to link the IPv6 address and the user/host. To ensure that all information is stored properly in the monitoring system, the SNMP polling interval has to be shorter than the expiration timeout of the ND Cache. Otherwise, some entries in the ND Cache could expire without being downloaded into the central system.

5. Conclusion

The IPv6 autoconfiguration options are not straightforward. There are two different sets of mechanisms and protocols, and one cannot work effectively without the other. Currently, it is not possible to configure Recursive DNS Servers addresses with SLAAC, but with DHCPv6 it is not possible to configure the default gateway address. As a result, the only working method is to use both protocols. Failure of either mechanism, whether through faulty configuration, poorly debugged software or targeted attacks, leads to denial of the communication for the end-node, and thus, for the user. Moreover, diagnostics are fairly complicated in this situation and require a good understanding of the way both mechanisms work. These problems, together with lack of security mechanism are probably also the reason, why the IPv6 is still not widely deployed. ISP will not deploy a new protocol which allows to an attacker to restrict the connectivity to the ISP's customers. Even though that IPv4 and IPv6 protocols are incompatible, when both protocols are deployed, inproper functionality of IPv6 would have also an impact to IPv4 connections – e.g. long delay when a user is accessing a website. Unfortunately, considering the standartization process in IETF, it does not look like, there will be any changes in the near future.

6. Acknowledgement

This work is part of the project VG20102015022 supported by Ministry of the Interior of the Czech Republic and was partially supported by the research plan MSM0021630528.

Bibliography

[1] Deering, S. *ICMP Router Discovery Messages.* 1991. http://tools.ietf.org/html/rfc1256
[2] Thomson S., Narten T., Jinmei T. *IPv6 Stateless Address Autoconfiguration*. 2007. [online] url: **http://tools.ietf.org/html/rfc4862**
[3] Narten, T., Draves, R. and Krishnan, S. RFC 4941 - *Privacy Extensions for Stateless Address Autoconfiguration in IPv6*. [online] 2007. url: http://tools.ietf.org/html/rfc4941.
[4] Narten, T., Nordmark E., Simpson W., and Soliman H. *Neighbor Discovery for IP version 6 (IPv6)*. RFC 4861, September 2007. url: http://tools.ietf.org/html/rfc4861
[5] R.Droms, J.Bound, B.Volz, T.Lemon, and C.Perkins. *Dynamic Host Configuration Protocol for IPv6 (DHCPv6)* . RFC 3315, July 2003. url: http://tools.ietf.org/html/rfc3315

Ing. Matěj Grégr; Ing. Tomáš Podermański.; prof. Ing. Miroslav Švéda, CSc.
Brno University of Technology, Faculty of Information Technology
Božetěchova 2, 6126 6 Brno, Czech Republic
email: igregr@fit.vutbr.cz, tpoder@cis.vutbr.cz, sveda@fit.vutbr.cz