

opsec  
Internet-Draft  
Intended status: Informational  
Expires: 4 December 2023

F. Gont  
G. Gont  
SI6 Networks  
2 June 2023

## Implications of IPv6 Addressing on Security Operations draft-ietf-opsec-ipv6-addressing-00

### Abstract

The increased address availability provided by IPv6 has concrete implications on security operations. This document discusses such implications, and sheds some light on how existing security operations techniques and procedures might need to be modified accommodate the increased IPv6 address availability.

### 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 <https://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 4 December 2023.

### Copyright Notice

Copyright (c) 2023 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 (<https://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 Revised BSD License text as described in Section 4.e of the [Trust Legal Provisions](#) and are provided without warranty as described in the Revised BSD License.

## Table of Contents

<a href="#">1.</a>	Introduction . . . . .	<a href="#">2</a>
<a href="#">2.</a>	The Semantics of IPv4 Addresses and IPv4 Prefixes . . . . .	<a href="#">3</a>
<a href="#">3.</a>	The Semantics of IPv6 Addresses and IPv6 Prefixes . . . . .	<a href="#">3</a>
<a href="#">4.</a>	Security Operations . . . . .	<a href="#">4</a>
<a href="#">4.1.</a>	Enforcement of Access Control Lists (ACLs) . . . . .	<a href="#">4</a>
<a href="#">4.2.</a>	Network Activity Correlation . . . . .	<a href="#">5</a>
<a href="#">5.</a>	Implications of IPv6 Addressing on Security Operations . . . . .	<a href="#">5</a>
<a href="#">5.1.</a>	Access-Control Lists . . . . .	<a href="#">5</a>
<a href="#">5.2.</a>	Network Activity Correlation . . . . .	<a href="#">6</a>
<a href="#">6.</a>	Advice on Security Operations . . . . .	<a href="#">7</a>
<a href="#">6.1.</a>	Access-Control Lists . . . . .	<a href="#">7</a>
<a href="#">6.2.</a>	Network Activity Correlation . . . . .	<a href="#">10</a>
<a href="#">7.</a>	Security Considerations . . . . .	<a href="#">10</a>
<a href="#">8.</a>	Acknowledgements . . . . .	<a href="#">11</a>
<a href="#">9.</a>	References . . . . .	<a href="#">11</a>
<a href="#">9.1.</a>	Normative References . . . . .	<a href="#">11</a>
<a href="#">9.2.</a>	Informative References . . . . .	<a href="#">12</a>
	Authors' Addresses . . . . .	<a href="#">13</a>

## [1.](#) Introduction

The main driver for the adoption of the IPv6 protocol suite is its increased address space, which can provide a vast number of public addresses for every device attached to the public Internet.

IPv6 addresses [[RFC4291](#)] can differ in a number of properties, such as address scope (e.g. link-local vs. global), stability (e.g. stable addresses vs. temporary addresses), and intended usage type (outgoing communications vs. incoming communications).

IPv6 hosts may configure and use multiple addresses with different combinations of the aforementioned properties, depending on the local host policy and the local network policy. For example, in networks where Stateless Address Auto-configuration (SLAAC) is employed for address configuration, host will typically configure one stable address and one (or more) temporary addresses per network interface, for each prefix advertised advertised for address configuration. On the other hand, in networks that employ Dynamic Host Configuration Protocol for IPv6 (DHCPv6) [[RFC8415](#)] for stateful configuration, it is quite common for hosts to configure a single stable address per network interface.

[Section 2](#) discusses the semantics of IPv6 addresses in terms of the entity or entities the identify, according to the deployed Internet. [Section 3](#) discusses the semantics of IPv6 addresses in terms of the entity or entities the identify, according to the deployed Internet.



[Section 4](#) discusses the usage of IP addresses in security operations. [Section 5](#) discusses the implications of IPv6 addressing on security operations. Finally, [Section 6](#) provides guidance on the usage of IPv6 addresses for security operations.

## **2. The Semantics of IPv4 Addresses and IPv4 Prefixes**

While the original Internet architecture defined IP addresses to identify a network interface, the deployed Internet has embraced Network Address Translation (NAT) over time. Initially, NATs were deployed on customers premises (e.g., in Customer Edge routers). However, as a result of the imminent IPv4 address space exhaustion, Internet Service Providers (ISPs) have resorted to deploying the so-called Carrier-Grade NATs (CGNs).

As a result, in scenarios where an ISP has *\*not\** deployed CGN, an IP address will typically identify one single customer, whereas in scenarios where an ISP has deployed CGN, a single IPv4 address may identify multiple customers. Depending on the type of "customer" (e.g., a home user vs. an educational institution), one or multiple actors might be associated with the "customer" in question.

In the IPv4 Internet, one may assume that an attacker will have control over a single IPv4 address (an IPv4 /32). However, an attacker might be able to leverage DHCP RELEASE messages to switch over different IPv4 addresses (/32s), and hence control more than a single IPv4 address.

## **3. The Semantics of IPv6 Addresses and IPv6 Prefixes**

As noted in [Section 1](#), IPv6 hosts typically configure multiple addresses with different properties. One of the most common deployment scenarios is that in which the subnet employs SLAAC [[RFC4862](#)] for address configuration, and where hosts configure both stable [[RFC8064](#)] [[RFC7217](#)] and temporary [[RFC8981](#)] IPv6 addresses. From this perspective, it is clear that multiple addresses may correspond to the same IPv6 host.

While rather uncommon in legitimate use cases, an IPv6 host may configure addresses from a larger address block. For example, it is common for Internet Service Providers (ISPs) to lease a /56 or a /48 to each subscriber, and thus a skilled user could readily employ the leased prefix in a single or multiple IPv6 hosts (whether virtual or not).

On the other hand, while one might assume that an IPv6 address would correspond to at most one host (strictly speaking, to one network interface of a host), this is not necessarily the case in the



deployed Internet. For example, deployments that employ "Network Address Port Translation + Protocol Translation" (NAPT-PT) [[RFC2766](#)] for IPv6 are not uncommon, whether along with technologies such as Kubernetes, or in IPv6-enabled VPNs. Thus, a single IPv6 address may actually identify multiple IPv6 hosts.

#### **4. Security Operations**

There is a whole range of security processes and operations that involve the usage of IP addresses. This document discusses the implications of IPv6 addressing on security operations via two examples:

- \* Enforcement of Access Control Lists (ACLs)
- \* Network Activity Correlation

The following subsections discuss these two examples in detail.

NOTE: [[RFC9099](#)] provides an overview of the most important aspects of IPv6 security operations, whereas this document elaborates on the implications of IPv6 addressing on security operations.

##### **4.1. Enforcement of Access Control Lists (ACLs)**

It is common for network deployments to implement any of these types of Access Control Lists (ACLs):

- \* Allow-lists
- \* Block-lists

Allow-lists are typically employed as part of a defense-in-depth [[NSA](#)] strategy, where access to specific resources may be allowed only when requests originate from specific IP addresses or prefixes. For example, an organization may employ a Virtual Private Network (VPN), and require that certain resources be accessed only via the VPN, by enforcing that requests originate from the IP address (or addresses) of the VPN concentrator.

On the other hand, block-lists are typically implemented to mitigate threats. For example, a network firewall might be fed with an IP reputation block-list that is dynamically updated to reflect the IP address (or addresses) of known or suspected attackers.

Both types of ACLs have a similar challenge in common: identifying the minimum set of addresses that should be employed in the ACLs definitions such that the ACLs can successfully enforce the controls



they are expected to enforce while minimizing collateral damage. For example, in the case of allow-lists, the corresponding ACLs should encompass possible legitimate changes in the set of legitimate addresses, thus avoiding false negatives (i.e., incorrectly preventing access to legitimate users). On the other hand, in the case of block-lists, the ACLs should encompass the attacker's ability to use different addresses (or vantage points), while minimizing false positives (i.e., incorrectly blocking legitimate users).

#### **4.2. Network Activity Correlation**

Another fundamental aspect of security operations is that of network activity correlation (at times with the goal of attribution). That is, a security analyst may want or need to infer the relationship among different network activities, and possibly assess whether they can be attributed to the same actor. This may be necessary for security investigations, but also to e.g. subsequently mitigate a threat by enforcing ACLs that block the alleged attacker.

### **5. Implications of IPv6 Addressing on Security Operations**

#### **5.1. Access-Control Lists**

A key question when implementing ACLs is deciding which granularity to use for the ACL specification. If one were to follow IPv4 practices, one would be tempted to specify ACLs with a /128 granularity (i.e., the equivalent to a /32 in the IPv4 world). However, as noted in [Section 3](#), most IPv6 host implementations employ IPv6 temporary addresses [[RFC8981](#)], and thus an allow-list specified as a /128 would eventually fail. Thus, one might be tempted to specify an allow-list as a /64 -- that is, an entire /64 might need to be "allowed", to accommodate the usage of IPv6 temporary addresses [[RFC8981](#)]). However, since such IPv6 prefix might be shared by other hosts in the same subnet, this would likely result in false-positives (i.e., all hosts in the target /64 would be allowed access) -- which is probably unacceptable in most cases.

In some scenarios, a network administrator might be able to disable the use of temporary addresses [[RFC8981](#)] via e.g. group policies [[GPO](#)], or by enforcing the use of DHCPv6 [[RFC8415](#)], thus having more control on the addresses employed by local hosts. In these specific cases, it might be possible to implement an allow-list for a host by specifying a single IPv6 address (i.e., a /128).

NOTE: Some IPv6 host implementations, notably the one in the Android operating system, do not support DHCPv6. Therefore, the option to enforce DHCPv6 usage might be unfeasible.





On the other hand, implementing block-lists may also be tricky. For example, IP reputation lists (whether commercial or not) are commonly employed in the deployed Internet, and used to e.g. dynamically configure ACLs on devices such as firewalls. However, these IP reputation lists generally specify offending addresses as /128. This means that an attacker could simply regularly change his/her IPv6 address, thus reducing the effectiveness of these lists.

Additionally, an attacker regularly changing his/her address might (whether intentionally or inadvertently) cause the block-list to grow to such an extent that the proper functioning of the associated filtering devices might be affected -- and thus the filtering device may have to resort to trimming the block-list.

Similarly, tools of the kind of [[fail2ban](#)] are commonly employed by system administrators to mitigate e.g. brute-force authentication attacks by banning IP addresses after a certain number of failed authentication attempts. These tools might ban IPv6 addresses on a /128 granularity, thus meaning that an attacker could easily circumvent these controls by changing the IPv6 source address every few attempts (e.g. before an address becomes banned). Additionally, as with the IP reputation lists previously discussed, an attacker performing a brute force attack \*and\* regularly changing his/her addresses could cause the block-list grow to an extent where it might negatively affect the system enforcing the block-list, or might cause other legitimate entries to be discarded in favor of the transient IPv6 addresses.

One might envision that IPv6 reputation lists might aggregate a large number of offending IPv6 addresses into a prefix that encompasses them. However, this practice is not really widespread, and it might also increase the number of false positives. Thus, this is a topic that may warrant further research.

## **[5.2.](#) Network Activity Correlation**

Performing IPv6 network activity correlation can be very tricky, since the semantics of an IPv6 address in terms of what an address may identify (see [Section 3](#)) can be complex. As discussed before, a single IPv6 address could correspond to either a single host, or multiple hosts behind an IPv6 NAPT-PT device -- this being similar to IPv4 scenarios.

However, multiple IPv6 addresses might or might not identify multiple different actors. In some cases, some heuristics might help infer whether a group of addresses belonging to a /64 correspond to the same host. However, as the attacking addresses become more sparse (e.g., an attacker leverages a /48), this may be more challenging. And, while some heuristics could be employed to perform network



activity correlation across multiple addresses, most tools commonly used in the deployed Internet do not implement these kind of features.

NOTE: Section "2.6.2.3. Correlation" of [[RFC9099](#)] discusses network activity correlation for local nodes, whereas [[IPv6-Scanning](#)] discusses the challenges of network activity correlation when detecting IPv6 scanning attacks.

## **6. Advice on Security Operations**

### **6.1. Access-Control Lists**

This section provides advice on the usage of IPv6 ACLs, whether as allow-lists or block-lists.

#### **6.1.1. Allow-lists for the Destination Address of Incoming Packets**

This type of ACLs are typically enforced when a network firewall is meant to allow incoming packets to subset of nodes on the local subnet.

The feasible granularity of such allow-lists will depend on the address configuration method employed in the local network. ACLs with a granularity of /128 will only be feasible if:

- \* DHCPv6 IA\_NA is employed to lease stable addresses to local hosts, and IA\_TA (DHCPv6 temporary addresses) is disabled, or,
- \* SLAAC is employed for host address configuration, and use of temporary addresses [[RFC8981](#)] is disabled, or,
- \* Manual configuration is employed for the local hosts.

It should be noted that, as a result of Neighbor Cache Exhaustion (NCE) attacks [[RFC6583](#)], it might be desirable to limit the allowed destination address ranges to the IPv6 addresses or prefixes that are actually in use in the target network. For example, in scenarios where DHCPv6 is employed, allow-lists for the destination address of incoming packets could be specified with the same granularity as the DHCPv6 address pool -- e.g. in a /64 subnet where a DHCPv6 server leases addresses from a /112, a /112 prefix could be used to specify an allow-list for such group of DHCPv6 hosts. However, when SLAAC is employed on the local subnet, and IPv6 temporary addresses [[RFC8981](#)] are enabled, the entire /64 would need to be allowed when specifying an allow-list for the Destination Address of incoming packets.



NOTE: The only alternative to specifying a /64 allow list would be to configure (stateless) ACLs for the stable addresses of the IPv6 hosts, and allow for the dynamic creation of stateful rules for packets that originate from the local network.

### **6.1.2. Allow-lists for the Source Address of Incoming Packets**

These type of ACLs are typically employed to allow incoming packets only when they originate from a specific IP addresses or prefix.

In a similar vein as the allow-lists from [Section 6.1.1](#), the granularity of these allow-lists will depend on the address configuration method employed at the network where packets originate.

Allow-lists with a granularity of /128 will only be feasible if:

- \* DHCPv6 employed for address configuration, and IA\_TA (DHCPv6 temporary addresses) is disabled, or,
- \* SLAAC is employed for host address configuration, and temporary addresses [[RFC8981](#)] are disabled, or,
- \* Manual configuration is employed for the local hosts.

NOTE: SLAAC [[RFC4862](#)] does not provide a mechanism to convey a policy as to whether temporary addresses [[RFC8981](#)] should be configured. This policy is typically a local host policy, which may be overridden via out-of-band mechanisms such as GPOs [[GPO](#)]. Since temporary address are typically preferred over stable addresses, a granularity of /128 will only be feasible if temporary addresses are disabled.

In scenarios where DHCPv6 is employed at the remote network, allow-lists for the source address of incoming packets could be specified with the same granularity as the DHCPv6 addresss pool of the remote network. For example, in a /64 subnet where a DHCPv6 server leases addresses from a a /112, a /112 prefix could be used to specify an allow-list for the group of DHCPv6 hosts.

In all other cases, it would be unfeasible to specify an allow-list for the source address of incoming packets with a granulary other than /64, since their addresses would be randomly selected from the /64 prefix.



**6.1.3. Block-lists for the Source Address of Incoming Packets**

As noted in [Section 3](#), attackers may have control over large IPv6 address blocks, and might be able to change their IPv6 addresses within such address blocks, rendering /128 IPv6 block-lists ineffective.

Security technologies meaning to enforce IPv6 block-lists should be able to infer when an attacker has control over an IPv6 address block, such that the granularity of the associated block-lists can be dynamically adapted to effectively enforce the intended controls. This subsection describes one possible way to implement this.

The following table specifies one possible set of parameters to be employed with this implementation:

```

+=====+=====+=====+=====+
| LEVEL | PREF_LEN | AGGR_THRES | ACL_LIFETIME |
+=====+=====+=====+=====+
| 1 | /128 | 10 | 1 hour |
+-----+-----+-----+-----+
| 2 | /64 | 10 | 45 min |
+-----+-----+-----+-----+
| 3 | /56 | 10 | 30 min |
+-----+-----+-----+-----+
| 4 | /48 | N/A | 15 min |
+-----+-----+-----+-----+
    
```

Table 1: ACL Granularities

The meaning of each of the parameters is as follows:

**LEVEL:**

ACLs may be enforced with different granularity levels, ranging from 1 to N, where 1 corresponds to the finest granularity, and N corresponds to the coarsest granularity.

**PREF\_LEN:**

Prefix Length corresponding to each granularity level. In our table, the finest granularity is a /128, whereas the coarsest granularity is a /48.

**AGGR\_THRES:**

A threshold specifying the number of ACLs of this level that, if/when possible, should be aggregated into an ACL of level (n+1).

**ACL\_TIME:**

The maximum lifetime of an ACL for this level.





The algorithm would work as follows:

1. If/when malicious activity is detected for an IPv6 address, create LEVEL=1 ACL (i.e., an ACL with a /128 granularity, and a lifetime of ACL\_LIFETIME(1)).
2. If/when possible, aggregate at least AGGR\_THRES(n) LEVEL(n) ACLs into a single LEVEL(n+1) ACL (with a ACL\_LIFETIME(n+1) lifetime).
3. Once ACL\_LIFETIME(n) has elapsed, eliminate the associated LEVEL(n) ACL.

As an example, if e.g. offending activity were detected for the IPv6 address 2001:2b8:0:1::1, a 2001:2b8:0:1::1/128 ACL would be created. If offending activity was subsequently detected for the IPv6 address 2001:2b8:0:1::2, a 2001:2b8:0:1::2/128 ACL would be created. If a total of 10 (AGGR\_THRES(1)) offending IPv6 addresses were detected, the associated LEVEL(1) ACLs would be aggregated into a single 2001:2b8:0:1::/64 (LEVEL(2))ACL, with a lifetime of 1 hour (ACL\_LIFETIME(2)). If offending activities were detected for IPv6 addresses in the 2001:2b8:0:2::/64 prefix, individual /128 ACLs would be created for each IPv6 address, which would eventually be aggregated into a single 2001:2b8:0:2::/64 ACL. Then, if 10 (AGGR\_THRES(2) /64 (AGGR\_LEVEL(2)) ACLs were eventually created in the 2001:2b8::/56 (AGGR\_LEVEL(3)) prefix, these ACLs would be aggregated into a 2001:2b8::/56 (AGGR\_LEVEL(3)) ACL, etc.

## **6.2. Network Activity Correlation**

As discussed in [Section 3](#), performing IPv6 network activity correlation can be tricky. As the bare minimum, security tools should allow security analysts to select the granularity to be employed when performing network activity correlation. For example, security tools should allow security analysts to specify that all activities within a given /128, /64, /56, or /48 correspond to the same actor.

## **7. Security Considerations**

This entire document is about the implications of IPv6 addressing on security operations. It analyzes the impact of IPv6 addressing on a number of security operations areas, raising awareness about the associated challenges, and providing guidance on how IPv4 security operation practices should be adapted to embrace IPv6.



## 8. Acknowledgements

The authors of this document would like to thank (in alphabetical order) Henri Alves de Godoy, Nick Buraglio, Rich Compton, Oliver Gasser, William Herrin, Henrik Kramselund, Ted Lemon, Jen Linkova, Markus Reschke, Andrew Ruthven, Eduard Vasilenko, Eric Vyncke, and Andrew Walding, for providing valuable comments on earlier versions of this document.

This document borrows some text and analysis from [[I-D.gont-v6ops-ipv6-addressing-considerations](#)], authored by Fernando Gont and Guillermo Gont.

Fernando would also like to thank Nelida Garcia and Jorge Oscar Gont for their love and support.

## 9. References

### 9.1. Normative References

- [RFC4291] Hinden, R. and S. Deering, "IP Version 6 Addressing Architecture", [RFC 4291](#), DOI 10.17487/RFC4291, February 2006, <<https://www.rfc-editor.org/info/rfc4291>>.
- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", [RFC 4862](#), DOI 10.17487/RFC4862, September 2007, <<https://www.rfc-editor.org/info/rfc4862>>.
- [RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", [RFC 7217](#), DOI 10.17487/RFC7217, April 2014, <<https://www.rfc-editor.org/info/rfc7217>>.
- [RFC8064] Gont, F., Cooper, A., Thaler, D., and W. Liu, "Recommendation on Stable IPv6 Interface Identifiers", [RFC 8064](#), DOI 10.17487/RFC8064, February 2017, <<https://www.rfc-editor.org/info/rfc8064>>.
- [RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", [RFC 8415](#), DOI 10.17487/RFC8415, November 2018, <<https://www.rfc-editor.org/info/rfc8415>>.



- [RFC8981] Gont, F., Krishnan, S., Narten, T., and R. Draves, "Temporary Address Extensions for Stateless Address Autoconfiguration in IPv6", [RFC 8981](#), DOI 10.17487/RFC8981, February 2021, <<https://www.rfc-editor.org/info/rfc8981>>.
- [RFC9099] Vyncke, É., Chittimaneni, K., Kaeo, M., and E. Rey, "Operational Security Considerations for IPv6 Networks", [RFC 9099](#), DOI 10.17487/RFC9099, August 2021, <<https://www.rfc-editor.org/info/rfc9099>>.

## 9.2. Informative References

- [fail2ban] fail2ban, "fail2ban project", <<https://www.fail2ban.org/>>.
- [GPO] Microsoft, "Windows Server 2012 R2 and Windows Server 2012", 2016, <[https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh831791\(v=ws.11\)](https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/hh831791(v=ws.11))>.
- [I-D.gont-v6ops-ipv6-addressing-considerations] Gont, F. and G. Gont, "IPv6 Addressing Considerations", Work in Progress, Internet-Draft, [draft-gont-v6ops-ipv6-addressing-considerations-02](#), 1 June 2022, <<https://datatracker.ietf.org/doc/html/draft-gont-v6ops-ipv6-addressing-considerations-02>>.
- [IPv6-Scanning] Richter, P., Gasser, O., and A. Berger, "Illuminating Large-Scale IPv6 Scanning in the Internet", IMC '22: Proceedings of the 22nd ACM Internet Measurement Conference, Pages 410-418, <<https://doi.org/10.1145/3517745.3561452>>, October 2022, <<https://olivergasser.net/papers/richter2022illuminating.pdf>>.
- [NSA] NSA, "Defense in Depth: A practical strategy for achieving Information Assurance in today's highly networked environments", <[https://web.archive.org/web/20121002051613/https://www.nsa.gov/ia/\\_files/support/defenseindepth.pdf](https://web.archive.org/web/20121002051613/https://www.nsa.gov/ia/_files/support/defenseindepth.pdf)>.
- [RFC2766] Tsirtsis, G. and P. Srisuresh, "Network Address Translation - Protocol Translation (NAT-PT)", [RFC 2766](#), DOI 10.17487/RFC2766, February 2000, <<https://www.rfc-editor.org/info/rfc2766>>.



[RFC6583] Gashinsky, I., Jaeggli, J., and W. Kumari, "Operational Neighbor Discovery Problems", [RFC 6583](#), DOI 10.17487/RFC6583, March 2012, <<https://www.rfc-editor.org/info/rfc6583>>.

#### Authors' Addresses

Fernando Gont  
SI6 Networks  
Evaristo Carriego 2644  
1706 Haedo  
Provincia de Buenos Aires  
Argentina  
Email: fgont@si6networks.com  
URI: <https://www.si6networks.com>

Guillermo Gont  
SI6 Networks  
Evaristo Carriego 2644  
1706 Haedo  
Provincia de Buenos Aires  
Argentina  
Email: ggont@si6networks.com  
URI: <https://www.si6networks.com>



