preprint of ARES WS paper

6G Network Needs to Support Embedded Trust

Raimo Kantola
Department of Communications and Networking
Aalto University
Finland
raimo.kantola@aalto.fi

ABSTRACT

A slogan coined at the recent first Levi 6G Summit by Peter Wetter of Nokia Bell Labs was "the 6G is about the 6th sense". This can be understood in at least two ways. One is that the network just knows what to do in all kinds of situations because of the use of AI and the second is that 6G radio will be widely used to sense the environment where the users are. In this view, 6G is seen as a continuation of the merge of the physical and the virtual words. An outcome of the Summit is a 6G White Paper documenting the ideas of some 60 invited people from the 300 participants about the future generation coming after 5G. This paper provides further discussion and justification of the trust and security aspects of the Networking Chapter in the White Paper. Opinions expressed here are of the author of this paper who was also the main editor of the Networking Chapter. The members of the White paper group on Networking or the editors of the White paper should not be held liable for the views expressed in the paper.

CCS CONCEPTS

• Network protocols, Network algorithms, Network properties.

KEYWORDS

6G, trust, reputation, networking, ID/Locator split

ACM Reference format:

FirstName Surname, FirstName Surname and FirstName Surname. 2018. Insert Your Title Here: Insert Subtitle Here. In *Proceedings of ACM Woodstock conference (WOODSTOCK'18). ACM, New York, NY, USA, 2 pages.* https://doi.org/10.1145/1234567890

1. Introduction

A starting point for the 6G requirements [1] is the ongoing merger of the physical and the digital worlds. People talk about "programming the world"[1]. Already 5G is offering lots of use cases and solutions in this area and 6G will take it further. In terms of security, it follows that the security risks will grow as compared to the world where computers and hackers stay in the confines of the digital world. When cars, homes, trucks, moving machines in harbors, logistics centers and warehouses, the electricity distribution networks, energy generation and storage units etc. are connected to a data network, if hacking is feasible, stealing goods, destroying other people's property and many other real-world crimes can be aided by hacking. It will even be possible to kill people either by accident or commit murder intentionally by the same approach. Moreover, the boundary of the digital and physical worlds is a new battlefield in a Hybrid war and therefore how we build it is a matter of national security for all developed nations. Lack of security of the boundary is not what our societies can tolerate. Sorting out all this crime would require a new kind of police force that would cost the governments lots of money. Based on this reasoning, the White paper suggests that we should embed trust into the network itself, automate trace back of attacks, prevent distributed denial of service (DDoS) attacks from succeeding as an inherent feature of the network and be able to add any kind of additional security features utilizing cloud technology to assist low cost and low power devices.

Besides consumer use for entertainment and infotainment, the 6G network will be used for vertical industries. This is stating already in 5G. Many of the vertical use cases are close to the boundary of the physical and the digital worlds. Many of the use cases naturally require a high level of security. Even today we should not accept that from time to time, we cannot access our money in our bank account because some teenager or a pissed off young man decided to use some openly available DDoS tools against the bank and require some "respect" or ransom. As our dependence on the Internet keeps growing, the White Paper suggests to work with the goal of making this impossible at least in anything that resembles or clearly belongs to the critical societal services such as e-government, banking, industrial logistics, road traffic, electricity grid etc.

This paper focuses on the outline of the solution to embed trust and DDoS mitigation into the networks. The white paper has a wider purpose but we exclude other requirements and topics from this paper.

2. State of the art in end user device and Internet security

DDoS attacks can be executed by first renting a botnet from hackers and then by using freely available attack tools on the bots.

[1] Lauri Oksanen of Nokia Bellabs.

If the purpose is more than just annoying the victims, an option is to contact the victim and require ransom. Instead of paying, the victim can engage a service provider to defend against the attack. Typically, then all the incoming traffic to the service (e.g. a bank) is routed through a brush-up center of the service provider. The brush-up center uses powerful cloud computing resources to filter the traffic, drop the attack traffic and admit the legitimate traffic. Once the attack is over, the temporary routing arrangement is removed and the network setup returns to normal. Sometimes, however, the attack has such a high volume that the brush-up center is overwhelmed at least initially. In this case, it may be possible to engage even more resources from the cloud and restart the mitigation.

A method that is used for DDoS mitigation is to withdraw a BGP route distributing the withdrawal announcement in a special BGP community among the Internet Service Providers (ISPs). This typically causes some disturbance to legitimate traffic. Overall, the methods for DDoS mitigation are commercial services that are not standardized and are an area of technical and business competition between the providers running brush-up services. Brush-up services are often provided using similar technology and software that is also used in Content Distribution Networks (CDN).

In many parts of the world (e.g. EU, India) ISPs work under Net neutrality regulation [2,3]. For example, the EU regulation forbids the ISP filtering the traffic based on user consent. Filtering is limited to cases when the network or a large set of users are threatened by the attack and also for only as long as is necessary. Under EU regulation the ISPs are prevented from offering differentiated security services to end users based on user consent. For the ISPs and mobile operators (MNO), security provided from the network nodes or the operator cloud is a cost item, not a revenue source and the ISPs act based on instructions from the National Cyber Security Center.

At the same time, consumer markets lack compulsory security certification for network connected consumer devices. Typically, computers owned by companies have firewalls and other kinds of security software. Even the computers owned by consumers can be well protected from hacking through the Internet connection using open source or proprietary firewalls, virus detection etc. The level of security in other consumer devices, on the other hand, can vary and is often poor. Security awareness of the users of these devices is also typically low. There are rather recent examples, like the January 2019 attack on GitHub with a volume of more than 1 Terabit/s or the earlier DDoS attacks using the Mirai botnet, that was built by hackers using poor security consumer devices. The botnet was used e.g. to take down Netflix for a while. Mitigating of an attack of this magnitude is likely to require up-to a hundred powerful cloud-based computers each with a network interface card of 40 Gbit/s or more. The datacenter network access capacity needs to be several times the maximum volume of the attack, too.

To summarize, the weaknesses of the existing technology include (1) a high capacity DDoS attack manages to cause a disturbance to legitimate traffic at least for a period, some of the mitigation methods themselves cause a temporary disturbance and (2) the mitigation method typically has a maximum capability. Due to net neutrality regulation ISPs are prevented from or lack motivation for developing better DDoS mitigation methods that would apply perfectly accurate filtering to the attacks and would be able to overwhelm any attack volume below network capacity essentially removing the DDoS "business" opportunity for the hackers.

3. Framework for embedding trust into the network

The framework we are proposing divides the end to end connection to three parts: (a) originating customer network, (b) public wide area network and (c) destination customer network. End to end communication takes place over a chain of trust spanning all those parts.

Figure 1 shows the framework that allows embedding trust into the network. In the framework, end to end network connectivity is from one customer network to another across the wide area. The edge node has a registry of served hosts, it assigns and maintains stable IDs for all served hosts and translates IDs to addresses and addresses to IDs on request. The edge node collects evidence of behavior of all seen entities against the ID of the entity supporting and using the idea of reputation of the hosts and network entities. The edge node maintains a reputation value for all seen Internet entities such as DNS servers, hosts and remote edge nodes. The reputation values can be either based on self-collected evidence or many nodes can share their experience.

In each of these areas, technology choices are independent of the other areas. This is a consequence of applying the generic flow abstraction introduced in Software Defined Networking (SDN), implemented e.g. using OpenFlow[2]. This abstraction, when applied to current technology, implies that the same control methods are used to control flows that are formatted to use IPv4, IPv6, MPLS, Ethernet, VXLAN, Geneva, GRE or IPSEC. If we have e.g. an Open Flow Switch (OVS) on the boundary of the areas, it can mangle packets between all the listed formats. These formats are generically called forwarding protocols.

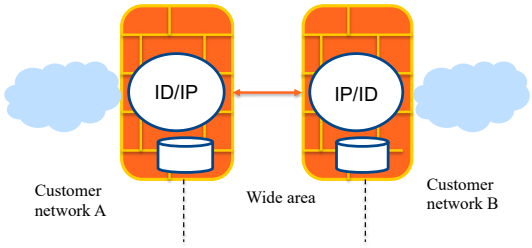

Figure 1: Communications Trust Framework.

[2] By 6G we expect that the SDN technology available is more advanced than OpenFlow.

Embedding trust into the network can be done by implementing an *end-to-end connectivity layer* on top of the forwarding protocol layer. The connectivity layer deals with willingness and refusal to communicate. For example, if the destination host does not trust the initiator of the connection or the initiator's network, it should be able to refuse communication as a part of the normal routine at the beginning of the communication session. For the trust decision the destination network or host will need reliable information on the past behavior of all kinds of Internet entities. A proof of concept implementation of this kind of IP/locator split using host names as IDs is reported in [4].

There is a difficulty in collecting information about the behavior of Internet entities, in particular, the behavior of host computers. Lots of them use dynamic or NATted IP addresses that are allocated on-demand by some network box serving the host. In case of a network source address translator (SNAT), the client host is actually identified by a combination of the NAT outbound address and an outbound port number that are typically allocated on a session or similar level of granularity. So, the same host may be seen with a quite different ID the next minute it did something suspicious or malicious. To overcome this difficulty, the white paper suggests that 6G network will apply an ID/routing locator (Rloc) split architecture.

In the ID/Rloc split, we separate the two functions of an IP address that tries to be both the locator used in routing and the identity of the host. The Rloc and Rloc prefixes are used for routing edge to edge and we have some other identifier for the host such that the edge node that manages the addressing and the identification will always translate one to the other on request. The node will also have the responsibility of logging all new ID assignments and address assignments. This architecture assumes that the edge node will always have a binding state for communications sessions similar to a NAT binding state in the state of the art.

Trust can be implied by a good reputation based on past history of behavior. To make this work, all entities must have a stable ID against which the evidence of behavior can be collected by all other nodes. Every edge node that needs to make trust decisions can either just use the evidence that it has collected itself or has been collected within a trust domain by several nodes. Nodes can share either the evidence itself or the conclusions made on the pieces of evidence. Both distributed and hierarchical structures for processing the evidence are interesting possibilities and can be researched in this framework.

The trust framework depicted in Figure 1 should also support the possibility that one of the edge nodes tells the other that your host is being too aggressive, restrain it when it communicates with me and even suggest the type of restraining like: block it altogether or rate limit its flows to a number per second etc. Naturally, all such filtering should be implemented for a duration possibly agreed between the edge nodes. The serving edge does not need to trust the destination edge blindly but in order not to lose its own reputation, its best option is to comply with the request towards the requesting node. When the serving edge gets a similar request from several destinations, it should assume that the host concerned is very likely a bot taking part in malicious activity.

The framework will apply the logic that if the remote edge refuses to comply with its request of restraining a misbehaving host, the serving edge itself will be blamed. This creates an incentive for the serving edge to do as the remote edge requests.

3.1 About the IDs

Collecting evidence of behavior is meaningful if the information collected today can be applied directly tomorrow even if the host appears with a different IP address or IP address, port pair. For this purpose, the ID against which the collection takes place should be the same across all destinations and it should stay the same over a longish time, at least for days. In the framework, each edge node that allocates addresses, address, port pairs and IDs, must be able to take responsibility for the served hosts towards the rest of the Internet and restrain the correct host when complaints about bad behavior of a host come in. Blaming a different host rather than the culprit should be impossible or very rare. In a simple case, the ID is unique for a host and stable from SW installation to SW installation or even longer and even if the host attaches through a different edge node. The proposed architecture however takes a little bit less strict attitude, the white paper just says that the ID is stable for the purpose of reputation information collection. How to best implement this taking into account all relevant constraints is for further study.

An example of a customer network is a cellular mobile network, e.g. 5G. Mobile devices have an equipment ID and the mobile subscribers SIM cards have an International Mobile Subscriber Identifier (IMSI). Would these work as suitable IDs for the purposes of reputation processing? In my opinion we could technically use any of these IDs but we do not recommend it. E.g. the mobile networks go to lots of trouble to minimize the use of IMSI in signaling because once an attacker or eavesdropper knows the IMSI of a mobile user, an attempt at breaching confidentiality or man-in-the middle attack can be attempted. Therefore, in the signaling over the air interface, IMSI is replaced by some other temporary ID. As we move from 4G to 5G this hiding is becoming even more elaborate than before. For reputation information collection, the serving edge node must give the host ID to the remote destination edge on request. Giving the host IMSI to the remote destination would defeat the efforts that have been made elsewhere to minimize the use of IMSI in signaling.

The normal pattern of communication over the Internet is that first the client issues a DNS query, most often this is the A-query for getting the IP of the destination as a response to the domain name that the client knows. So, the client already uses a domain name to identify the destination. A straightforward step would be that also the client has a domain name for identification and that the client's edge node will give it to the destination edge node on request. To stay on top of its responsibility for the host behavior, the edge node will have to apply the following rules:

- It must allocate both the private addresses and Rlocs for the host

- It must have a registry of the IDs and their mapping to the addresses
- It must limit the frequency of giving the host a new ID so that the ID can be called "stable" or it must be able to chain the new ID with the old one.

An RFC [5] argues that giving a clear text Domain name to just anyone is harmful for the privacy of the host. If we accept this argument, for reputation processing, the ID can be a derivative of the host domain name and the edge node is the only party that knows the translation from the derivative to the real Domain name. Also, from the remote edge point of view, it does not need to know the plain text name of the host that it wishes the remote edge to constrain in some way. It is enough that it can use some ID that the serving edge understands and based on this ID the serving edge can restrain the suspicious or harmful host in the required way. Nevertheless, the reputation-based methods will work only if the ID is "stable", i.e. stays the same at least for some reasonable time e.g. for a few days.

3.2 Comparison with the current communication pattern

In this section we compare what the Framework proposes with the current communication pattern over the Internet and point out enhancements or particular variations of the pattern that are needed to comply with the framework. In 6G, this does not need to be the way the framework is implemented, another protocol and communications architecture could be chosen. The purpose of the comparison is to illustrate the differences how the current Internet works and how we believe 6G needs to work.

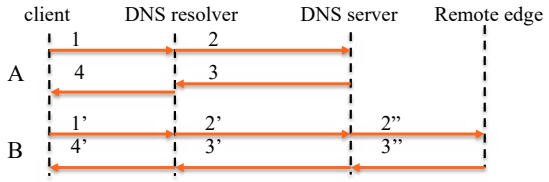

Figure 2: Name resolution

Figure 1 (A) shows the typical but simplified name resolution on the Internet. Using the domain name of the server, the client wishes to communicate with, it sends a DNS address query (1) to its DNS resolver (default DNS server) that will browse (2) the DNS hierarchy until it locates the DNS server that has been delegated authority over the destination domain name or until the destination address is found in some cache on the way. In (2) the address or the ID of the client is not given to the DNS server or any other server closer to the destination. It is not even compulsory that the source address of the client is correct in (1). When client can spoof its address in the query, this allows using DNS for reflection attacks. A-response (3) will contain just the address of the destination. If more information is needed, another query type must be used and the resolver needs to act as a proxy that swaps the A-query to something more elaborate. Finally, the resolver responds to the query (4). On the client, some applications, in addition to A-queries, support other types of queries and are able to use the rich features of DNS. However, this is not uniform and having millions of applications already implemented, this is not easy to change any time soon. For example, support for the SRV records that could return a destination port number in addition to the destination address is not uniform.

Having received the response (4) the application typically initiates a flow, e.g. a TCP flow, towards the destination. Some applications like the web -browsers may initiate many flows to the same destination one after the other. In pattern A, the remote edge has no way of knowing that the incoming (TCP) flow relates to the previous A-query. Like the DNS server, it has no idea of the source IP that will appear in the TCP SYN packet. Unlike the DNS server, it has not even seen the address query. This makes the remote edge vulnerable to a SYN flood attack. This can be mitigated by a SYNPROXY that however, is not part of the "Internet architecture" if there is one.

For implementing the framework, the trust gaps in pattern A, must be filled in pattern B. Changing how the applications behave is possible only in specialized networks with a limited set of applications. The consumer Internet, this is not realistic. So, let us assume that 1' = 1. However, the DNS resolver resides in the edge gateway to the wide area network and assigns the private and public outbound addresses and also the IDs, the edge node can and must exclude source address spoofing in the query. So, it will always send 4' to exactly the same host where it got 1'. If in 1, the source address was spoofed, the query is dropped in order to protect the resolver's reputation. To eliminate spoofing, the server and the remote edge can apply enhanced DNS (EDNS) methods either as they are or in some more fine-grained form on the message flow 2' and 2''. The remote edge can also eliminate spoofed DNS queries from not-so-well known DNS servers and resolvers, by demanding that the query is sent over TCP instead of UDP. Finally, DNSSEC can be used to ascertain the authenticity of the original authoritative DNS records.

The remote edge node will be the authoritative DNS server for the destination hosts residing in the destination customer network. The remote edge can apply reputation-based filtering to the incoming DNS query, the reputation can relate to DNS nodes on the way, in particular the resolver, or, due to EDNS, the initiating host on some level of granularity. If the remote edge holds a bad reputation of any of them, under high load or attack it may decide not to respond at all, tell that the destination is not available etc. Even in this case, the remote edge is in difficulty trying to relate the incoming TCP flow to the DNS query, some heuristics are, however, often possible based on proximity in time of the DNS query and the initiation of the flow. Still a SYNPROXY is either a part of the remote edge or in front of it towards the Internet.

3.3 Related work

The idea of ID/Locator split is not new. In [6, 7, 8] an architecture is proposed. The solution is similar to Host Identity Protocol [9] in the sense that the host protocol stack would have to be changed. We argue that this is not feasible simply because too

many applications have already been written to the IPv4 socket interface for the normal TCP/IP stack. In addition, we argue that for a change of this magnitude there must a really good reason. Beatifying the host protocol stack or scaling the architecture a little better are not good enough reasons to convince the millions of users and network admin on the necessity to change the host stack and updating all the millions of applications.

IETF has also created a new protocol called LISP [10] that separates host addresses from addresses used for host location in the larger network. Although the name of the protocol leads to believe that the protocol would introduce Identifiers that are never used for location, this is not correct. The EID or end-point identifier, the ID introduced by LISP, is actually an address in a local network. Similar to [6, 7], there is no explicit trust model in this proposal either.

Our proposal differs from these earlier works in two aspects: (1) we assume that it is not feasible to change the protocol stack in hosts nor change the applications running on hosts, because there are too many of those already in use and (2) the sole purpose of the ID we will need is to act as an identifier against which trust related data can be collected and an identifier that can be used by the remote and serving edge nodes to talk about the behavior of the hosts served by the edge nodes and restrain the hosts in case of detecting malicious behavior without necessarily ever distributing the plain text domain name or any other plain text ID of the hosts. If there are any changes in the hosts, they should be optional. Some of the ID/Locator split proposals support heterogeneous forwarding formats. Since we assume an edge node with a flow state, this can be easily supported in our proposal as well. Many of the proposals have also other appealing features.

4. Conclusions

We argue that when the physical and digital worlds merge closely, Internet level security will not suffice because of the tight dependence of physical safety on information security. Several approaches can be proposed to overcome the security issues on the physical/digital boundary. One is to build so called specialized networks [3] that isolate the boundary from the Internet and leverage cellular mobile networks for remote access. In this approach the boundary nodes or none of the critical Internet of Things (IoT) elements are visible on the classical Internet at all. Therefore, no consumer Internet device can directly take part in an attack against the boundary or critical IoT nodes. The downside of this approach is that it may lead to segmentation or fragmentation of the Internet which means that the approach is working against the Metcalfe's law of network value. The answer to this worry is that all these specialized networks still run on the same infrastructure as the Internet and also the cellular networks have already been penetrated by Ethernet links and the IP protocol although they are not fully part of the Internet. Therefore, in this network of networks, consumer and specialized, we can always make any legitimate communication take place.

The second approach of serving the need to merge the physical and the digital worlds is embedding trust into the network itself making sure that no simple attack will succeed and additional security for low power devices can be added to cloud-based elements on the infrastructure at will.

In this approach, DDoS using any consumer devices will be detected and stopped immediately by all involved edge nodes. DDoS mitigation capacity is as big as the network capacity of the involved nodes. Monitoring for DDoS detection is always on and no edge node trusts any other edge node blindly. It will have means to collect evidence of behavior and act on it as needed. The edge nodes can also share their observations within a trust domain, for example, within an administration.

REFERENCES

[1] K. Leppänen, M. Latva-aho, ed., White Paper of 1st 6G Summit, 2019 (arxiv.org).

[2] Regulation (EU) 2015/2120 of the European Parliament and of the Council of 25 November 2015 laying down measures concerning open internet access and amending Directive 2002/22/EC on universal service and users' rights relating to electronic communications networks and services and Regulation (EU) No 531/2012 on roaming on public mobile communications networks within the Union *OJ L 310, 26.11.2015, p. 1–18* ELI: http://data.europa.eu/eli/reg/2015/2120/oj

[3] Body of European Regulators for Electronic Communications, BEREC Guidelines on the Implementation by National Regulators of European Net Neutrality Rules.

[4] R. Kantola, J. Llorente Santos, N. Beijar, Policy Based Communications for 5G Mobile with Customer Edge Switching, Wiley Security and Communication Networks, 05/2015; DOI:10.1002/sec.1253.

[5] C. Huitema, D. Thaler, R. Winter, Current Hostname Practice Considered Harmful, RFC 8117.

[6] V. P. Kafle, ID/locator split networks, Akari, NICT, 2009.

[7] V. P. Kafle, H. Otsuki, M. Inoue, ID/Locator Split Architecture for Future Networks, IEEE Communications Magazine, 2/2010.

[8] S. Kanemaru, F. Teraoka, ZNP: A Network Layer Protocol Based on ID/Locator Split Considering Practical Operation, IEEE ICC 2011.

[9] R. Moskowitz, T. H. Hirschmann, P. Jokela, T. Hendersson, Host Identity Protocol v2, RFC 7401, April 2015.

[10] D. Farinacci, V. Fuller, D. Meyer, D. Lewis, The Locator/ID Separation Protocol, RFC 6830, 1/2013.

ACKNOWLEDGMENTS

Sudhir Dixit helped to write the Networking Chapter of the White paper, Matti-Latva-aho and Kari Leppänen are the overall editors. Tarik Taleb moderated the discussions on networking.