White Paper
Cooperative Security for 5G and the Internet

Raimo Kantola, Jesus Llorente, Hammad Kabir
Aalto University

Acknowledgements: This WP is based on work done in Aalto University partially with the support of MEVICO, SIGMONA Celtic Plus, the national BusinessFinland (TEKES) funded TAKE 5 projects and the 5G@II project funded by private foundations.

Abstract: *This White Paper advocates the idea of cooperative security for the Internet and 5G. The idea of cooperative security is that all good guys would cooperate automatically to mitigate all hacking over the Internet sharing evidence of misbehaviour, constraining detected infected hosts and deploying security patches as quickly as they become available. The paper outlines the solution called Customer Edge Switching as an implementation of cooperative security without going into much technical details. The paper discusses briefly some use cases under the constraints set by the network neutrality regulation. The technical details can be found in the academic journal and conference papers that are available in some form in www.re2ee.org and naturally e.g. in IEEE Explore. Most concepts presented here have been proven by Proof of Concept or running code level experimental implementation. Our current work is targeted to complementing the body of running code for further research and development. Code is published in GitHub/Aalto5G. The purpose of this paper is to facilitate executive level discussions and encourage additional work in this direction.*

1. Problem statement

The Internet is plagued by security breaches such as the recent problems caused by the *Wannacry* and *Petya* ransomware. In 2017, hackers demonstrated their competence by pooling the power of some 10M home appliances to produce a DDoS attack of 1Tbits/s using the *Mirai* malware. Most hackers are using known vulnerabilities for which cures or counter measures exist but have not been ubiquitously deployed. An example is the recent attack on suomi.fi, kela.fi that are Finnish e-Government services. It used the SYNFLOOD, an attack that uses a design weakness in TCP that is as old as TCP itself. Slow deployment of fixes is because each administration and user makes its own deployment decisions and may have to upgrade its applications at the same time as it deploys the security patch. Moreover, host security patch deployment may require upgrades in other systems. There are hosts connected to networks whose operating systems are not supported any more by the vendor, so it may be that no security patches are available. Also, vendors are allowed to sell to consumers goods that can and are meant to be connected to the Internet, but often have poor or non-existing security and possibly no automatic means of upgrading their software with new security patches when vulnerabilities are detected. This is because SW licences typically and completely legally take no liability for fit to purpose. Loose regulation speeds up the introduction of ever new innovations to consumers but is very bad for security. The commercial interests behind this state of affairs are so significant that it seems unlikely that the situation will improve any time soon.

Another hindrance to progress in securing the Internet is the *network neutrality* regulation that actively discourages ISPs and mobile operators (MNOs) to take a commercial interest in improving end system security. For example, in Europe, the regulation on “open Internet” says that all filtering ISPs apply to traffic *must be necessary* for protecting the network or the end systems. This discourages a proactive attitude to security and pushes the problem to end systems. Instead, the ISPs and MNOs limit their network services for improving end system security to what the regulator tells them to do. In [6] we argued that trust in the Internet is a problem that cannot be effectively solved by end systems alone. Instead, ISPs, MNOs and network admins should take an active role. Moreover, we showed that by implementing a system of indirect reciprocity, i.e. maintaining the reputation of all entities participating in communication, we could increase the welfare of all benevolent participants. Due to the ease of systems attacks against a reputation system consisting of end systems alone, we advocate that ISPs, MNOs and other network admins should participate and verify evidence of hacking indicated by end systems by trusted systems in their networks. However, even under the current network neutrality regulation it is possible to make progress towards cooperative security like we will discuss later in this paper.

Master hackers are using vulnerable hosts by taking them over and using them to distribute malware for more targeted attacks than just simple DDoS. By using several layers of other people’s computers, they can hide from being caught while causing havoc in network connected systems in hospitals, airports, banks, electricity grid etc. A whole ecosystem of grey economy around hacking and malware has emerged over the past two decades of the Internet era. The activity can be classified e.g. to amateurs, hacktivists with a political agenda, criminals who are after money and government agencies conducting espionage, information operations or cyber-attacks against foreign states.

The value of the network is reduced by the harm caused by these attacks. The attacks make it mandatory to invest in IT security software and appliances in all organisations that use the Internet to carry out their business. When these measures fail, cleaning after attack causes breaks in the normal business and work of the organizations. There are recent cases when an attack or a series of rather simple attacks caused disturbance to a commercial bank for almost a week. These examples show that reaching high reliability of services over the Internet is currently not feasible. Moreover, predicting the level of reliability is not feasible either: one cannot predict meaningfully the level or success rate of malicious activity which would be a crucial component in predicting the overall reliability. Even if we attempt to estimate the probability of succeeding in not being hacked, this does not help to improve our operations.

In the face of this situation, the prevailing attitude in IT security is still *everyone for himself*: organizations are reluctant to share information about being attacked and every organization makes its own deployment decisions separately. There is some cooperation between the security software vendors such as "Common Vulnerabilities and Exposures – CVE" or the CERT cooperation organized in many nation states. We recognise that companies have legitimate business reasons not to be very forthcoming in sharing the security incident information such as, publication may harm the public reputation and trust on the company or the reports are feedback to the attackers. At the same time, e.g. in Europe the regulation has widened the list of sectors of business where the organisations have a hacking incident reporting obligation. Mainly, this new regulation has been driven by the protection of consumer rights rather than being a part of a proactive and systematic counter measure to hacking.

2. Vision of cooperative security

To improve the situation and to make it possible to provide reliable services over the Internet, we must realise a new security principle: "*one for all and all for one*". This means that all good guys should cooperate in computer security against the brotherhood of hackers. The target is maximum automation of all counter measures against attacks up-to automatic security patch deployment. It should become

- practically infeasible to carry out successful attacks by trying to use known vulnerabilities because
- once a cure has been created, it should be automatically deployed in a centralized manner in the cloud if necessary to cover for weaknesses in host security[1] and
- once a connected computer has been detected to take part in an attack, it should be automatically isolated to a sandbox for security fixing while the rest of the hosts/networks should know the identity of the infected host/network and refuse to cooperate with it until the fixing has been carried out.
- On the upside for a host such as a smart phone or an IT device in Smart Grid, it should receive only traffic that it expects while all other traffic should be blocked by its agent at the network edge.

The realization of this vision requires a *comprehensive security and trust framework for the Internet*. This framework will have clear roles and places for malware and attack detection, for admitting traffic flows into/from customer networks, for restraining too aggressive infected hosts and for digital trust. It should be possible to flexibly manage the level of trust assurance that is applied to any communication. The expectations of the hosts and network entities are described as the *communications security policy* of that entity. It should be possible to pinpoint all resources the attack is using with accuracy and it should be clear what is the level of liability of the customer network serving an infected host.

Equally, it should be possible to implement the framework in different networks applying different business roles to players such as mobile application stores/vendors, security software vendors, insurance companies, ISPs, customer networks, equipment/systems vendors etc.

Ideally, the network neutrality regulation should be reviewed in the area of security exceptions, making it feasible for operators to participate in improving Internet wide trust from their commercial interests without defeating the purpose of the "open Internet" regulation. This calls for striking a new balance between the interests of the sender and the receiver, and protecting the benevolent users while counter acting in cooperation against the malicious actors.

[1] Imagine a TV set that has no automatic software upgrade capability but it has vulnerabilities. A policy in a CES node could be created to whitelist all connections related to TV services and blacklist all other possible connections to safeguard the TV from being used by hackers to attack other network connected devices. In this case waiting for the TV to be taken over by hacker before doing anything is not a wise approach because we expect that infecting the TV with a virus would take minutes rather than days or weeks. From then on, it would be a matter of hackers fighting over the ownership of the TV among themselves.

A constraint for the implementation of the vision is that administrations should be able to make deployment decisions one network at a time.

3. Why in 5G

5G sets the target of being able to provide *ultra-reliable services*. Such services will mainly be needed for Industrial Internet of Things (IIoT) using mainly machine to machine communication. We believe that since 5G is part of the Internet, this is possible only if at least selectively 5G will include a comprehensive system of cooperative (end system) security. 5G introduces the concept of network slice. A slice is a generalization of the concept of virtual private network in the sense that a slice may include features on all OSI layers and in addition to forwarding elements may also include compute elements and slice specific software. The control software may also be shared between several slices. This opens an opportunity to new types of network functions to meet particular user needs.

Beyond what is said by 3GPP, TAKE-5 introduces cooperative firewalling into the core network slices that strive to provide ultra-reliable service. The purpose is to eliminate source address spoofing, DDoS that uses currently easily available methods and control all flows by personalized communications security policies. In addition, cooperative firewalling introduces an interworking function with the legacy Internet such that flows can be initiated to 5G devices also from legacy Internet sources under policy control. This internetworking function is called Realm Gateway (RGW). An implementation of the RGW has been deployed in the TAKE 5 test network already in Otaniemi, Finland. Due to RGW, one network at a time deployment of our Cooperative Security framework becomes feasible. An implementation of RGW is available in GitHub/Aalto5G.

4. Solution

4.1 Cooperative Firewalling

Initially in a slice that seeks to provide ultra-reliable service, access to the Internet is controlled and secured by a cooperative firewall. TAKE5 solution for this purpose is called Customer Edge Switching.

The solution has 3 components: (1) interworking using Realm Gateway (RGW) and SYNPROXY; (2) CES-to-CES communication; and (3) communications security policy management. This is also the natural order of deployment of the functionality.

4.1.1 Phase 1: Realm Gateway

The RGW is both the client-side normal source network address translator (SNAT) and the server-side destination network address translator (DNAT). In both cases NAT binding is established dynamically making it possible to initiate flows either from a private client to a public server (SNAT) or from the Internet to mobile devices served by the RGW (DNAT). The DNAT function is linked to DNS leaf node in the RGW making it possible to allocate a DNAT inbound IPv4 (or IPv6) address for a short period for use by the expected flow [3, 12, 13, 16]. To make sure that no spoofed packets can reserve or hijack such addresses, SYNPOXY sitting between the Internet and the RGW will block spoofed SYNs from entering the RGW. The SYNPROXY is integrated with the RGW with an interface that allows pushing some TCP parameters of the hosts to the SYNPROXY for high performance of the TCP flows. We have published two SYNPROXies in GitHub/Aalto5G: (1) a user space implementation based on NETMAPS that is rich in features and has been shown to deflect a SYN Flood of 40Gbit/s using just 3 CPU cores. At this rate, the implementation requires direct access to the NIC. (2) Another SYNPROXY implementation, relying on normal Linux TCP/IP stack, making use of Netfilter (iptables) is not quite as rich in features but works well even in a virtualized environment without direct access to NIC.

The above approach of managing inbound RGW addresses is called Circular Pool of Public Addresses (CPPA) and it is effective for most applications that need just a single flow (defined by a 5-tuple of IP addresses, ports and the protocol). HTTP is different; browsers may establish a set of HTTP flows for rendering a single page. For HTTP, RGW includes a reverse HTTP proxy thus making it possible to run a www server on a device that only has a private address. In the future the reverse proxy will be replaced by an Application Layer NAT that has been demonstrated in the ldpAirwall in GitHub/Aalto5G improving scalability of the RGW.

Phase 1 has been deployed as depicted in Figure 1.

RGW applies local policies to admit flows. It implements a *Policy Based Resource Allocation* for new flows. It receives only non-spoofed SYN –packets and uses heuristics in Enhanced DNS (EDNS) query processing, it maintains reputation values for remote entities and decides on resource allocation using the reputation values of the remote entities. These entities include DNS servers/resolvers, remote hosts with a globally unique address and sets of remote hosts using one NAT outbound address. The solution can also accept external reputation data from Surricata Intrusion Detection system. In SoftFIRE competition in March/April 2018, this solution won the Silver Prize in the security track.

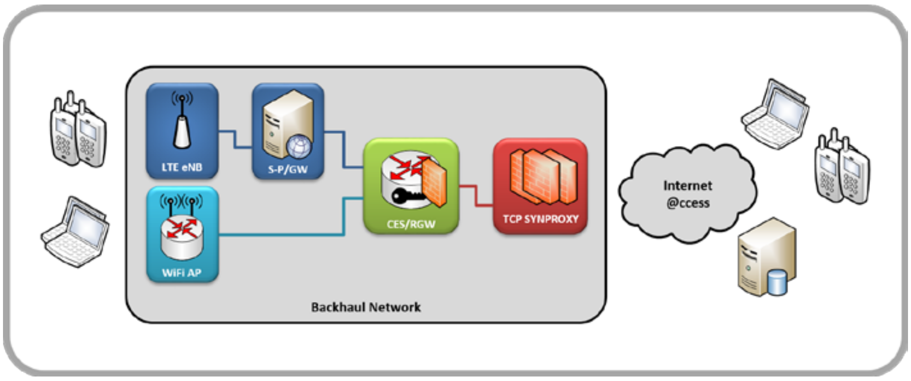

Figure 1 Realm Gateway and Synproxy in TAKE 5

4.1.2 Phase 2a: CES to CES communication

The Customer Edge Traversal Protocol (CETP) is responsible for the policy exchange between the cooperative edges we call CES nodes. The hosts behind the CES nodes can communicate with each other after a successful policy negotiation where both the CES and the host policies are evaluated against their counterparts on the remote site. The CETP separates CES-based and host-based policy negotiations. Any host level policy negotiation requires an underlying CES-to-CES connection that guarantees the required level of trust and security, with the network of the remote host.

It is the responsibility of a serving CES node to establish and maintain stable identities of the served hosts. As a result, we are able to establish a chain of trust "*host–CES-node–CES-node–host*". In the current implementation, we use Fully Qualified Domain Names (FQDN) as identities for hosts and CES nodes.

The CETP protocol, presented in Figure 2, has 3 layers: the bottom layer is for signaling transport and can use several connections for reliability; the middle layer manages the CES-to-CES relation and finally the Host to host layer negotiates host to host policies. All aspects of CETP are policy controlled making it possible to control the exact level of needed trust assurance for the communication freely.

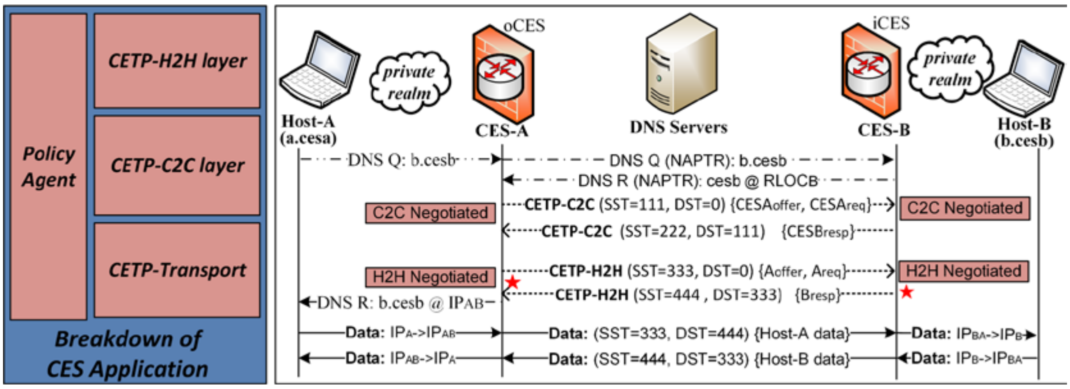

Figure 2. CES-to-CES Control Plane

CES seeks to optimize the number of roundtrips needed for host to host negotiation by separately controlling each of the CETP layers. For a DNS query initiated by a served host, a CES node always generates a NAPTR query to find out if the remote end is behind a CES node. If not, the serving CES node falls back to RGW, which depicts SNAT behaviour in this case.

A CES node acts as an authoratative DNS server for the served hosts. If the remote host is served by a CES node, upon the DNS response (NAPTR record) from the remote edge stating that CETP service is available to reach the remote host, the local CES node triggers CETP negotiation. If not already established, first the signaling transport is negotiated, then CES-to-CES policies are matched and finally the host to host policy negotiation takes place. Only then will the DNS response (A-record) be returned to the initiating host telling that the destination is reachable at a *local private proxy address*. All CETP negotiation is transparent to the hosts. Normally, the DNS query timeout (2s) will not expire at the host during the edge to edge negotiation and even if it does, the re-attempts of the original query will be absorbed by the serving CES node. By default, Linux and Windows hosts are configured to re-attempt the DNS A-query 4 times with increasing timeouts. Upon negotiation success, the CES data plane node has a binding state that maps the private and global addresses at the edge.

Host data is tunnelled between the CES nodes. Several tunnelling techniques are currently supported, such as GRE, VXLAN, or GENEVE. In addition, a CES node can support encryption of arbitrary host to host communications. The data plane of CES is realized with Linux; where we leverage OpenvSwitch (OpenFlow enabled) for tunnel management and StrongSwan (IPSec) and Netfilter for secure data communications.

One of the functions of cooperative firewalling is that a CES A being in communication with a CES B can tell the CES B to slow a particular host-b down or restrain it when CES A sees host-b as too aggressive in any way. This makes it possible to push the brakes to attackers to the attacker's own network. Naturally, if the CES-B does not comply, CES A will lower the reputation of CES B and allow it reduced resources locally. If there are several CES nodes in the same network with CES A, they can share the reputation information within the network. Wider sharing can naturally be accommodated.

4.1.3 Phase 2b: Communications Security Policy management

The purpose of the SPM is to ensure that hosts will receive only flows that they expect and that all other traffic is blocked at the edge node where the cooperative firewall acts as the host's trusted agent. A factor of the trust decision is the identity of the remote party that a CES node must be able to verify. Therefore, the Firewall acts cooperatively, can ask and will respond to questions before the admit decision are made. The security policy management system (SPM) is modelled on the 3GPP policy management architecture (that deals with QoS rather than security). The SPM is the component that allows adapting cooperative firewalling to different use cases such as Mobile Broadband or Industrial Internet of Things while the security engine is generic.

An abstract blueprint of the SPM is presented in Figure 3.

A utility App on Android or Linux gleans the list of active Apps on the device with the port and other networking resources in use and the App ID and sends them to the User Policy System that after verification stores the information into the intermediate database. Here, the policies can be edited by the user using a web interface. Another tool is used by the admin to insert its policies into the database. The SPM client pushes the policies to the active MySQL database used directly by the Firewalls to retrieve the policies for their served hosts. The policy format adopted in the experimental system is JSON.

Policies are hierarchical: the CES admin decides the policies for CES-to-CES relations and CES packet transport. The network operator may wish to group its users into policy groups. Finally, it is feasible that there are host to host level policies that are based on full knowledge of which applications run on which device and how the end user wishes to use those applications. The idea is, that under policy control, the device (and the air interface) will see only the traffic that it expects in any given state that reflects the active SW on the device and the user's wishes. In different use cases, the sourcing of policies may differ.

This level of policy control does not, unfortunately, seem feasible at least in Europe for consumer customers of MNOs. But it seems feasible for corporate customers when not seen to replace "open Internet" access as well as for "special services" such as machine to machine.

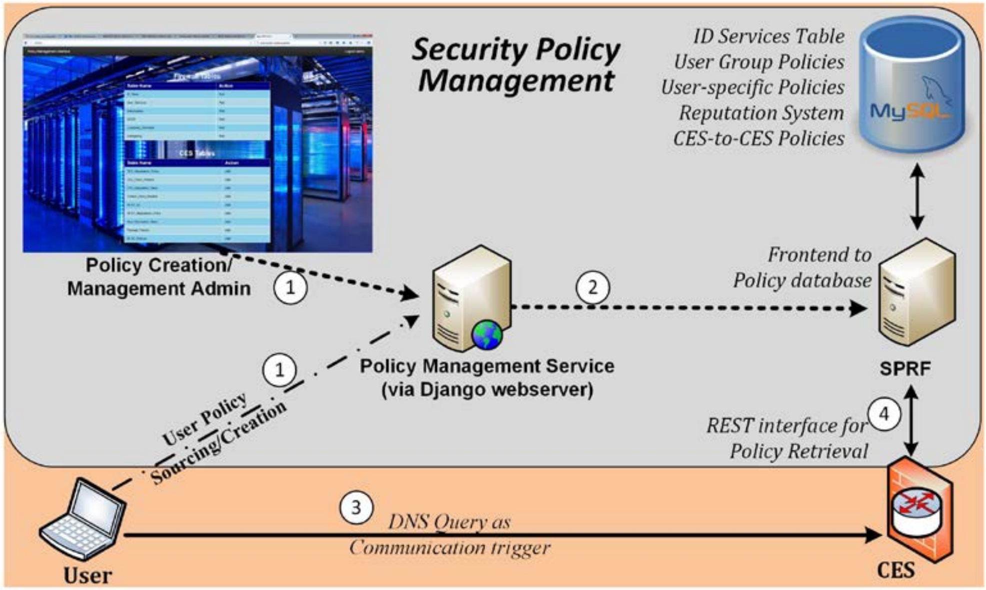
Security Policy Management
ID Services Table
User Group Policies
User-specific Policies
Reputation System
CES-to-CES Policies
MySQL
Policy Creation/ Management Admin
1
2
Frontend to Policy database
Policy Management Service (via Django webserver)
SPRF
1
User Policy Sourcing/Creation
REST interface for Policy Retrieval
4
3
DNS Query as Communication trigger
User
CES

Figure 3. Communications Security Policy Management for Mobile Broadband

In the Policy Database, the key to policy is the *Service Fully Qualified Domain Name* (sFQDN) of the service on the particular device. When an executable policy is created by the SPM for a CES node to execute, the different levels of policy hierarchy are consulted and the executable policy is bound to actual IP addresses, ports and protocols that will appear in the packets traversing the Firewall.

In different use cases, some additional components may be needed for a suitable SPM. For example, for the Industrial Internet we expect that the services outsourcing and data sharing contracts should be established and stored in an electronic contracting service with a contract database. If access to sensors and actuators and their data needs to be under access control in any way (which we assume to be the normal case), a contract implies a communications security policy that is recorded in the contract database as a security policy template. When this template is merged with knowledge of what software is available on the sensors and actuators, we can produce the Firewall policies.

4.2 High level solution logic

This section outlines the reasoning behind the proposed solutions of Customer Edge Switching.

4.2.1 Communicating with strangers and friends

The classical theory of cooperation – the foundations can be studies e.g. in the Book by Robert Axelrod "The Evolution of Cooperation, Basic Books", shows by using Game Theory that for maximizing your payoffs in an interaction it is best to cooperate with parties that you will meet again; while it is often best to cheat strangers that you would not meet again. Correspondingly, in a once-off Prisoner's Dilemma game *defecting* is the stable strategy that leads to higher individual gains for defecting entity, but it yields less than optimal overall sum of payoffs (for the whole population). This is a cooperation/social failure, because if both parties would choose to cooperate the sum of payoffs would be higher.

When the same game is used to model interactions in a group of e.g. 1000 players that are capable of learning and adopting winning strategies, magically (under the assumption of similar learning patterns used by every player) a cooperative strategy emerges. Axelrod showed by computer games that this cooperative strategy is "tit-for-tat". Other strategies have been proposed later but among the fair cooperative strategies, "tit-for-tat" – defined as: never defect first, but when you get cheated, immediately do the same to the cheater when you can and then forget about it – still remains a good candidate for maximizing the long term social welfare in the community of players.

This classical theory can be applied to the context of the Internet [6]. Due to weak identification (dynamic addresses, NATted addresses, spoofing, DNS poisoning etc) in the Internet the prerequisites for cooperative strategies to become dominant are not in place. We argue [6], that an implication is that "*there is no end to end solution to the problem of trust in the Internet"* and therefore, some network-based functions are needed to improve the Internet and make it a better platform for people and organizations to cooperate.

With the current popularity of Block Chain, one might object that may be there is a distributed solution. We remain sceptical about this possibility due to several factors:

- Block chain does not scale to the number of nodes we have in the Internet
- Read/write operations in the block chain are slow
- Trust is not an on/off parameter
- Between the hacker and the victim, most times there is another victim host that was previously taken over by the hacker. So, the interaction between the hacker and the current victim is not direct.

We propose in Customer Edge Switching that the first network function needed is a new type of edge node that acts as an agent of the host and establishes a chain of trust from the sender to the receiver. Prior to two hosts communicating (sending any packets to each other) the edge nodes should establish the required level of trust for connecting the two customer networks and the two hosts. The role of the edge node includes:

(1) Establishing stable identities for the hosts, so that reputation information can be collected and a system of indirect reciprocity can be applied in the Internet,

(2) Restraining its host when told to do so by the remote edge, because the edge node takes the responsibility for the behavior of the hosts it is serving and to a given extent it is liable for the behavior.

Given these functions, we suggest that the required prerequisites, as defined in the "Evolution of Cooperation", for cooperative strategies to become dominant will be put in place in the Internet on the level of basic connectivity. It seems clear that neither of the above conditions could be implemented under the "end-to-end" principle by the hosts involved alone.

4.2.2 Addressing: private or globally unique?

Classically, the leading Internet principle has been "end-to-end". Its simplistic interpretation is that each host should have a globally unique IP address. With IPv6 this would be possible even for the Billions of mobile devices and the tens or hundreds of Billions of IoT devices. And indeed, under the assumption of mostly nice users connected to the Internet, this would be the best platform for innovation in services and network connected machines and appliances.

Weaknesses in the argumentation for globally unique addressing include the following long list:

- It is trivial to attack a device in a globally unique address,
- A trivial attack can e.g. use source address spoofing that is not eliminated by such an approach while the recommended approach of source address filtering misaligns gains and investments.
- Even if the device has the best protection against such attack on the device itself, but the device is battery powered like most devices are in the present and future Internet, the execution of the protection software will deplete the batteries leading to denial of service.
- If IPv6 devices are protected by a network firewall from attacks, equally well, the firewall can translate the addressing from public to private and private to public.
- Many organizations wish to hide the exact topology of their networks from potential competitors or malicious actors. Such hiding is not possible under the simplistic idea of "end-to-end" and ubiquitous globally unique addresses.
- Host addresses will then appear in the core routing tables (at least host address prefixes) meaning that the scalability of the core will remain a constant challenge as the tens of billions of new devices are connected to the network. This core routing scalability issue applies to tens of thousands of network nodes owned by a large number of corporations and ISPs.
- Since host addresses are in the core routing tables, mobility of end devices such that execution of applications on the device would not be disturbed by the mobility, requires an add-on solution. Since more and more of the devices are battery powered and mobile, this is not ideal.
- In real life networks, this model has been abandoned in favor of and ad-hoc solution that allows hiding hosts from the core network by NATting the host addresses. In this solution, many hosts have just a private

address and this address is dynamically bound to a NAT outbound address by the NAT device on the edge of the network on a flow level. A NAT is an IP-layer single sided switch. By this we mean that a NAT binding state is normally set up only on client initiative while a server on a private host is unreachable. The recommended methods (STUN, TURN, ICE) of NAT traversal are unsatisfactory for mobile devices: peer to peer session setup is slow and lots of code and effort must be spent on application layer for the mundane task of being continuously connected if that is an application requirement.

In addition, global addressing does not, in practise, reach its goal of any application on any device being able to communicate with any application on any other device due to the trend of running more and more application layer protocols over HTTPS and most of the servers using a "well-known" fixed port. E.g. all HTTPS Apps would reside in port 443 on the mobile device. If there were no intermediate nodes of the application in cloud, the App would be truly peer-to-peer, and the first message of the application protocol would arrive at the device, the device would not have a socket ready to receive nor would it know which App should handle the packet. Therefore, usually, the App has been designed to *register into a cloud based intermediate node* and the device-based App is always just a client that "simulates" being a server when a new session arrives. Addressing to the App on the device succeeds, because the intermediate servers of the Apps have different IP addresses in the cloud and as a result, each device-based App listens in a different TCP/IP socket.

In IPv4 networks, IETF has allocated 3 address ranges for private addressing. In most of the world private addressing is the only way to connect mobile devices (exceptions are possible but to just a few). A private address can be reused by any number of devices residing in different private address spaces or realms. In mobile networks the typical model is that each mobile device resides in its own private address space; communication with another mobile device or an Internet host is possible only through a gateway that will translate the addresses dynamically as needed.

The popularity of NAT is explained mainly by two factors: (1) a NAT allows serving thousands of hosts with a single globally unique address and (2) a host behind a NAT cannot be easily attacked from the Internet because inbound flows are typically not enabled unless the host behind a NAT initiates the connection.

A problem with NATs is that not all application layer protocols (e.g. FTP or SIP) are NAT friendly. Such protocols require special methods of NAT traversal. Our claim is that because of the exhaustion of IPv4 address space, slow adoption of IPv6 and wide adoption of NATs, today it does not make sense to design new NAT-unfriendly application protocols and even more, it makes sense to avoid using old NAT-unfriendly protocols. Unfortunately, to reach NAT friendly status many communications protocols need to use some methods to reach hosts in private address space and the methods imply adding NAT traversal code to such applications. This conflicts with the age-old principle of layer independence in protocol and application design.

Advantages of NATting include:

- Scalability of the network to tens/hundreds of billions of connected devices
- Host addresses are not needed in core routing tables improving the scalability of the core network
- While the host moves within the private network, its movements are not visible in the Internet core improving scalability of the core.
- It is not trivial to attack a NATted host
- Hosts are hidden from direct attacks from an arbitrary Internet host.

This reasoning has led to CES addressing principle: *normally all hosts have just a private address, host level globally unique addresses are allowed to the extent they are available and needed for heavy duty servers.* Another key outcome of the reasoning is that ICE-like NAT traversal should be replaced by the possibility of establishing a dynamic DNAT binding at the receiver's network controlled by policy. For clients, a CES node naturally can act as a normal SNAT towards servers in a globally unique IP address. Therefore, CES can be seen as a generalization of NATting.

4.2.3 Routing and switching as forwarding methods

In our solution the edge is switched while the edges can be connected by a routed IP network. Routing that is based on globally unique addresses scales very well to short flows but does not allow elaborate control over the flow: all packets are treated in the same way. This even applies to malicious packets. Connection or flow level switching introduces binding state at the edge. The benefit is that any security or filtering algorithm that runs in about 0,1…2s can be added prior to admitting the flow to the receiver. Considering that (1) it is legal to sell network connected appliances with poor security to consumers, (2) more and more of the connected devices are battery powered and (3) lightweight so that they cannot run a bulky security software package that needs to be upgraded often, the binding state at the edge with the required security is what we need. Once we accept this premise, mangling packets between the private and global realms will be naturally done at the edge. The result is that IPv4, IPv6 etc become just forwarding protocols to be chosen for each network independently.

4.2.4 Security of DNAT

Strong security heuristics are needed to secure the private hosts. An alternative in case of DNAT would be to let any flow form a dynamic NAT binding to the receiver in a private address. This would eliminate host hiding (by NAT) that is normally expected in a private network lowering the level of security. We believe that this does not meet user needs. Therefore, effective security methods are needed. These include SYNPROXY that eliminates all spoofed SYN -packets from competing for the DNAT dynamic binding states. Given non-spoofed source addresses and by counting success and failure events, the function can maintain DNS resolver and host reputation and allocate binding states under high load based on reputation. This is called *Policy Based Resource Allocation* (PBRA). Even more restrictive policies are possible. The RGW uses PBRA to remote parties and leverages Enhanced DNS (EDNS) to achieve fine grained control.

4.2.5 Why policy control

The CES node always establishes binding state prior to letting two hosts communicate. For the binding states, memory is always finite. An attacker could try to exhaust the memory by initiating a large number of connection attempts. This is why the receiver side needs to be protected from all such attacks. Many different methods can be used to achieve good enough security. The usefulness of the methods may depend on the network environment. Possible methods include: enhanced DNS (EDNS(0)) carrying the sender ID in the DNS query or using TCP instead of UDP for DNS queries, spoofing elimination between the edges, checking credentials of the initiator or both edge nodes prior to CES-to-CES relation establishment, reputation of DNS resolvers, DNS servers, edge nodes, hosts etc.

There is a cost in establishing a CES-to-CES relation and the control plane transport connection, negotiating the host to host relation and finally establishing the host to host tunnel between the edge nodes. The cost can be lowered by caching state information and by controlling the level of assurance required for the connection by policy. If the destination is just a web server of a best effort kind, probably quite loose policy is enough under normal conditions for the sake of a responsive service. Under attack, the policy can be dynamically tightened. In case the www site is an e-commerce site, it may be better to use a somewhat stricter default policy.

In case the destination is an industry site where an Industrial Internet is deployed, access to the site should be strictly controlled based on permissions that mostly follow from the outsourcing contracts. This case could be alternatively implemented by a set of VPNs. Such VPNs are supported e.g. by BGP over the wide area. This approach burdens the BGP infrastructure with the needs of private networking. We believe that a more scalable approach is needed.

Two complementary methods can be used: (1) using SDN to quickly set up VPNs over the wide area – for this the currently non-existent east west interfaces for the SDN controllers are needed and (2) using policy control at the edge nodes. The latter allows using private links to carry and protect the more valuable traffic. This approach requires a policy controlled Firewall at the edge and scales well to a large number of services and devices with controlled access.

Based on this reasoning, we have chosen to make the edge node fully policy controlled and use the security policy management system (SPM) with its policy hierarchy to tailor the cooperative firewalling approach to any use case. This allows to build a generic security engine for a large number of various use cases and do the tailoring in a declarative manner in the SPM. This approach also allows for innovation in the way applications and devices are used because many new needs can be handled by changing the policy without touching the security engines at the edge nodes.

4.2.6 How to secure policy creation

Policy creation must be as automatic as possible. Tools such as we describe in section 4.1.3 will help, but a user-friendly policy editing interface for the users to modify policies for their applications is necessary. Under the assumption that device of a user cannot be fully trusted, the policy creation system must do all it can to prevent malicious policies from entering the active firewall policy database, or malicious Apps using the network resources. However, banks trust the mobile devices to be used by people to manage their money and investments. We believe that the policies are less critical than the money people have, so we do not see using the end device for policy sourcing as infeasible in terms of security.

When a user wants to download a new application, our policy management system can compare the App to a possible list of "safe or approved" applications and to a list of known malicious applications. For company owned smart phones, this might be enough. Different approaches are possible for the Apps that are not on any of the lists mentioned above.

4.2.7 Digital trust in wide area

Among primates, humans are super cooperators. If e.g. ape societies scale to a few tens of individuals, humans have developed effective ways how strangers, who do not even need to trust each other, can cooperate. Money is one such method, another is spoken and possibly recorded gossip. When communication takes place over the Internet, we often have a case of strangers trying to cooperate. Prisoner's dilemma tells us that often for maximum payoff it would be best to cheat or defect in such a case. But most people just do not wish to do that. However, for the few who have decided to cheat, Internet offers a great playing field.

Using money to manage the liability of communicating with a stranger is difficult because between the bad guy and the target, we have an unwitting (possibly careless) user who has lost its host to a hacker. Therefore, it makes sense to use the other well-known method, namely gossip, to make it harder to cheat and limit the gains from cheating. This requires a wide area reputation system.

Customer edge switching establishes the basis for a wide area reputation system by setting up a firm rule: a CES node will be responsible for the hosts that it is serving and will establish a stable ID for the hosts for the purpose of reputation management. This is important because in a packet we have just the source and destination IP addresses and they can be dynamically assigned or one globally unique address can in fact be representing many hundreds or even thousands of hosts.

Ideally, all good guys should be sharing their observations in order to form a reliable reputation for all entities including at least customer networks, DNS servers, hosts. There are legitimate reasons why this level of sharing is difficult to set up. At least 3 approaches are interesting topics for research:

(1) A CES A can tell the remote CES B with whom it is communicating to slow down or stop its host *b* and state some reason such as host *b* is generating too many flows towards hosts in CES A's network. If CES B complies, restraining host *b*, detecting conclusive evidence of malware on host *b* and cleaning host *b* from the malware is then the responsibility of the admin of CES B. If CES B does not comply, CES A can refuse to provide any service to hosts that are served by CES B for some duration T. If network A has many CES nodes, they can share their observations on host *b* and CES B allowing all the CES nodes to team up against all attacks from the CES B network.

(2) A Trust Alliance can be set up for several networks that agree to share their observations of remote networks and hosts and consequently act in concert against all attacks from the rest of the Internet. Willingness to share evidence of security incidents can be increased by letting corporate networks encrypt the evidence prior to sharing it with their ISP. Using *partially homomorphic encryption* that supports addition and multiplication by a constant, it is possible to aggregate the evidence and form a wide area reputation of the remote networks, DNS servers and hosts that can be used in all CES nodes in the Alliance. See our papers [5,8,11] where we showed that this is feasible. This approach requires a centralized trust operator in the Alliance. The system has 3 layers: (a) host/ customer network, (b) Internet Service Provider and (c) the Global trust operator (GTO). The ISP can hide the ID of the reporting network and host and only the GTO can decrypt the reports. For trust calculation it does not need to do so, because the aggregation takes place using the encrypted reports. Anonymization of the host ID helps to keep the ISP subscription relation confidential from 3rd parties as it should be for business reasons. When the

homomorphic calculation gives an actionable result, the result naturally is decrypted so that action against the guilty party can be taken.

(3) A distributed solution? The challenge is scaling any distributed solution to a large number of nodes. Let us recall that the distributed hash table requires *log N* hops to locate the needed information with one key value. In case more than one key are needed, the situation is even worse. Nevertheless, there may be cases, when a distributed approach is desired.

4.2.8 Future of NATs

Question is would it be feasible or desirable to remove NATs from the Internet like many pundits of the traditional Internet ideology seem to imply: lots of complaining about the difficulties caused by NATs can still be heard, while we are advocating that all such difficulties are due to bad protocol design that has obstinately ignored the fact that networks are full of NATs. One such example is the SIP protocol used for Voice over Internet that largely failed on the markets against a pragmatic SKYPE protocol that instead of complaining about NATs or advocating removing them found feasible methods of traversing them and became the first successful VOIP protocol.

Since the best use of NATs is for battery powered wireless devices, let's focus on this case. If such a device would have a globally unique IP address, it could be attacked by any host on the Internet. At the same time such attacks could cut the capacity of the air interface available to legitimate use. So, by pooling the resources of e.g. a million or ten million consumer gadgets with weak security, it would be feasible to completely block the mobile services provided by a mobile network operator for a time. This would make the operator look completely ridiculous and lose credibility on the market. For this reason, the mobile networks MUST be protected by a Firewall. Modern network firewalls are stateful. If so, the firewall can equally well translate between private and public addresses. This is because network node performance is limited in terms of the number of packets while the exact amount of processing per packet has a minor role.

Another de-facto complementary solution against such a possibly devastating attack is that communicating mobile Apps are designed to register into an intermediate server on the cloud before they become reachable. This consumes their battery and completely ignores the idea of end to end, but is accepted both for security and business reasons. Since mobiles in the hands of consumers are always clients, and only simulate being servers, they do not need globally unique addresses.

Finally, our research has shown that hosts in a private address can be made reachable to any Internet host using the RGW.

In the light of this logic, removing NATs does not serve any useful purpose. But the proposal does maintain the illusion that the Internet only has nice users or that the issue of trust can be effectively solved by hosts alone.

5. Use cases

In Europe, feasible use cases of cooperative security are limited by the network neutrality regulation. Any *filtering of traffic in a public network is strictly regulated* and the ISPs and MNOs are in a reactive rather than proactive mode in dealing with security. However, filtering in corporate networks for trusted services and in case if so called "special services" is feasible. For trusted corporate services, a mobile could be connected to a VPN where critical corporate services but also additional processing for Internet access would be allowed at least when this arrangement is not seen to replace "open Internet access".

Customer Edge Switching can be tailored to use cases by developing new policy tools for generating the required policies for the use case. Our original work considered enhanced Mobile Broadband in 5G as the use case. Under network neutrality regulation this would be needed for trusted corporate services.

A network of trusted services could be deployed for e-Health services in hospitals connecting medical devices, patient monitoring, doctors and other personnel. Remote services could be protected as well using a CES to CES scenario. Yet another use case is using 4G/5G networks for rescue services, the police and other government use: a high level of security of such services seems like a good idea.

A use case of "special services" might be Smart Grid with requirements of

- Managing the Grid with bi-directional energy flows under distributed electricity generation and storage using renewable methods, wind and solar.

- Protection from short circuits on distribution lines, where current measurement results at the endpoints are compared to detect the short and then circuit breakers are triggered to cut the connection in less than about 100ms from the cut taking place. Most of the time budget needs to be reserved to the circuit breakers.

We can use 5G (or Fiber) to exchange the measurement values; 5G terminals at the measurement points and a 5G network are needed to carry the packet flows on paths that give max a few *ms* of propagation delay. If this SDN style network is served by a CES firewall, it can also command e.g. a Ryu controller to set up the required flows. In such a network, only planned and intended flows are transmitted, the intention is defined in policies. If a hacker manages to get physical access to the network, it would be hard to fool the network to set up any new flow in the network. A much stricter security would be within reach as compared to just connecting the power meters into a regular IP network and patching security with IDS, and typical Firewalls.

The idea of 5G is to provide superior support for many new industrial use cases in Industry 4.0. There we are dealing with automated moving machines, the automation of production and logistics processes involving physical safety and very significant value. It is fair to expect that the security requirements of these applications are high, more like what we can today find in the finance sector rather than in normal offices. We propose to use CES enabled policy-based communications and possibly Software Defined Networking to achieve that level of security.

In 5G, the network is planned to support slicing. Security in different slices can be implemented differently without changing any of the current network neutrality regulation or causing any legal hassle. Some of the trusted corporate services could be supported by transport network slicing. It seems clear that it would be a bad idea to mix the machine to machine traffic with the normal human to human traffic in the same slice – this might limit the possible security architectures for the machine to machine traffic.

6. Summary

We have reported in a number of publications the experience of implementing Customer Edge Switching and the Realm Gateway on a level of proof of concept (PoC). The PoC was concentrated on verifying just the new algorithms while ignoring efficiency and many known attack methods. In Section 4.1 we outline an implementation that rethinks the role of SDN and Linux and extends the work to policy management. Its first parts are deployed in a test network and most of the running code is available for further experimentation in GitHub/Aalto5G. The plan is to publish the rest soon.

In this implementation, Netfilter in the Linux kernel is used to filter the packets and packet flows while OpenFlow is used to mangle the packets to the required packet formats. We have implemented the SYNPROXY both using Netfilter/iptables and in the Linux user space. We have also proposed and implemented several extensions to EDNS to mitigate the use of DNS for attacks. Naturally, other implementations of cooperative security are possible.

Using the running code in GitHub/Aalto5G, use case focused experiments are feasible with limited effort. We hope and expect that product level implementation and standardisation of the new protocol (CETP) and some other aspects of the solution will follow.

Resources

1. Y. Shen, Z. Yan, R. Kantola, Analysis on the Acceptance of Global Trust Management for Unwanted Traffic Control based on Game Theory, Computers &Security.
2. L. Chen, Z. Yan, W. Zhang, R.Kantola, TruSMS: A trustworthy SMS spam control system based on trust management, Future Generation Computer Systems, 7/2014.
3. R. Kantola, J. Llorente Santos, N. Beijar, Policy based communications for 5G mobile with customer edge switching, Wiley Security and Communication Networks, 5/2015.
4. Liyanage, M., Ahmed, I., Okwuibe, J., Ylianttila, M., Kabir, H., Santos, J. L., Kantola, R., Perez, O. L., Itzazelaia, M. U. & De Oca, E. M. Enhancing Security of Software Defined Mobile Networks, 2017 In : IEEE ACCESS. 5, p. 9422-9438 17 p.
5. L. Zhang, Z. Yan, R. Kantola, Privacy-preserving trust management for unwanted traffic control, in Future Generation Computer Systems, July 2016.
6. R. Kantola, H. Kabir, P. Loiseau, Cooperation and end-to-end in the Internet, International Journal of Communication Systems · February 2017.

7. Hammad Kabir, Raimo Kantola, and Jesus Llorente Santos, Customer Edge Switching – A Security Framework for 5G, Chapter 9 in M. Liynage, A Comprehensive Guide to 5G Security, Adobe Digital Edition.

8. Z. Yan, R. Kantola, Y. Shen, "Unwanted Traffic Control via Hybrid Trust Management", IEEE TrustCom 2012, Liverpool, UK, June. 2012.

9. J. Llorente Santos, R. Kantola, N. Beijar, P. Leppäaho, Implementing NAT Traversal with Private Realm Gateway, IEEE ICC 2013.

10. P. Leppäaho, N. Beijar, R. Kantola, J. Llorente Santos, Traversal of Customer Edge with NAT-Unfriendly Protocols, IEEE ICC 2013.

11. Y. Shen, Z. Yan, R. Kantola, "Game Theoretical Analysis of the Acceptance of Global Trust Management for Unwanted Traffic Control", in Proc. of IEEE HPCC 2013, Zhangjiajie, China, Nov. 2013.

12. J. Llorente Santos, R. Kantola, Transition to IPv6 with Realm Gateway 64, ICC 2015.

13. H. Kabir, J.Llorente Santos, R. Kantola , Securing the Private Realm Gateway, Proceedings of IFIP Networking, 2016

14. Amir, K. C., Goulart, A. & Kantola, R., Keyword-driven security test automation of Customer Edge Switching (CES) architecture, 21 Oct 2016 *Proceedings of 2016 8th International Workshop on Resilient Networks Design and Modeling, RNDM 2016.* IEEE, p. 216-223 8 p.

15. www.re2ee.org

16. GitHub/Aalto5G