SOMETHING ELSE: BEREC, NET NEUTRALITY AND BCP-38.

In the backwater of all discussions regarding DDOS in Sweden (And Europe) this spring i think it's important to not forget about the root cause of some of the problems. Not all, but most "mediaworthy" attacks recently has in some way been utilizing spoofed attacks. Most commonly is the reverse amplification attacks. To be able to produce such an attack you need three ingredients.

A botnet of computers or peripherals sitting on a network that allows spoofing.

A list of computers or peripherals that runs unregulated open services based on UDP, commonly DNS, NTP, Chargen and SSDP.

The IP-address of what you want to attack.

Take any of these out of the equation and these types attack will not be successful. To solve point #2 there is plenty of scanners around the world that scans the whole Internet or scans parts of the Internet to find vulnerable servers and then automatically notices that netblocks abuse-adress (if there is any). While some organisations takes responsibility for their abuse and actively follows them up and make sure that the abuse reaches the actual endpoint (SUNET recieves alot of abuse everyday, but most if not all is actually aimed for the connected institutions and not SUNET directly) but some has a redirect of abuse@ to /dev/null. Im not sure there is empirical data on this but it feels like whenever someone fix a vulnerable system there is two more popping up in some other part of the world.

To solve point 3 (which is quite common nowadays) is to try to obfuscate where your stuff actually is on the Internet. You can use DNS-tricks in combination with a cloud-based service to hide your valuable backends. Example of such services could be from Cloudflare, Akamai, AWS, Google Cloud and many others. Essentially you rely on someone else's infrastructure to "front" your services (or put all of your services there) . Going through all alternatives here requires a few hundred pages so lets skip that. Lets focus on #1.

How these attacks works is quite simple, you acquire a computer in a network that allows spoofing, this computer makes legitimate/optimized common requests to a open resolver, but instead of putting itself as the source of the attack the source will be rewritten to the DDOS-destination address, lets say sunet.se. The server on the other end will then be a good server as it should and reply with the requested data to the source-address, it has no idea whether or not it's spoofed since there is no handshake (this is UDP remember). The DDOS-target will then receive tons and tons of reply which it did not ask for but till has to process, then you multiply this method with 1000 bots making 100 requests per second to 100 different servers each generating hundred of thousands of unwanted replies back to the target.

To solve the spoofing-part there has been a lot of efforts throughout the year to get here (this is not a new problem, we had spoofing-problems since the dawn of Internet essentially). The most commonly referred to framework is BCP-38 (best common practice 38) from IETF. This is a set of good examples and guidelines on how to build networks that does not allow spoofing. This is a must-read for everyone that runs ISP-type of networks but unfortunaly its not a must-comply (yet). If all autonomous systems in the world (or atleast a major part) of them would stop spoofed attacks then these attacks would not be a huge issue and very hard to create.

The simplest way of stopping these attacks is to put a access-list on a suitable interface that only allows sources from a specific list, i.e that customer/server/something ip-address(es) and nothing else. This isnt very scalable and hard to keep track off without good automation. This is where RPF-checks, RFC3704 (Reverse Path Forwarding) comes into place. To put it simple it does exactly what the access-list with only the endpoints source-addresses does, but it does it dynamically. uRPF will look into the FIB and see if the source-address has a viable next-hop through the interface it was sourced from. This is commonly referred to as strict uRPF.

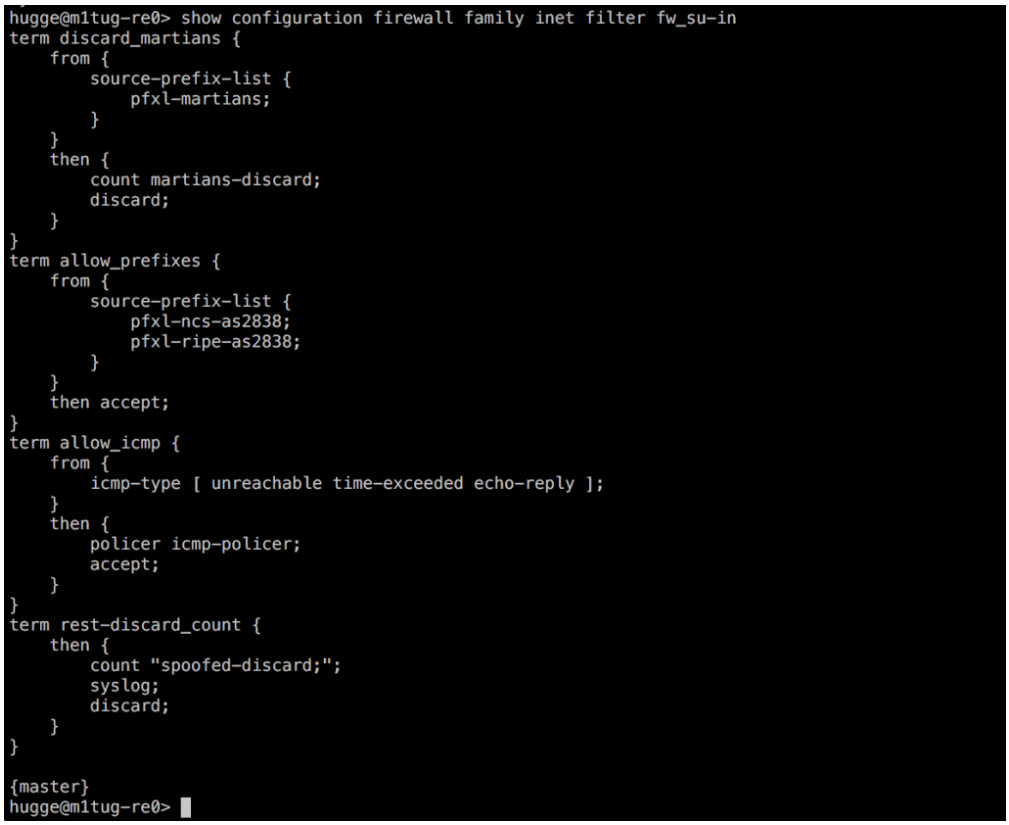
This can naturally pose a problem to multihomed entities where there might not always be a feasible entry in the FIB due to routing-policies taking place. This is when uRPF Feasible comes into play, uRPF Feasible does not only look into the FIB that there is a route pointing on the sourcing interface but it also looks into the RIB for a feasible path whether it's active or not. This solves most of the problems for asymmetric routing but can still pose a problem if the route-announcements is not consistent through all peers. This is hard for the upstream to control.

In SUNET we do feasible paths in all routers, and to solve the possibility of not having consisting route-announcements we utilize fail-filters. Lets take a look at some configuration.

```
hugge@m1tug-re0> show configuration interfaces xe-9/2/0
description "Link to SU Red, m1tug.su-br2";
vlan-tagging;
mtu 9192;
gigether-options {
    ignore-l3-incompletes;
}
unit 1 {
    description "link to su-br2";
    vlan-id 1;
    family inet {
        filter {
            input sample;
        }
        address 130.242.85.197/30;
    }
    family iso;
    family inet6 {
        address 2001:6b0:1e:37::2/127;
    }
    family mpls;
}
unit 10 {
    description "Link to LS SU su-br2, sunet-su-3";
    vlan-id 10;
    family inet {
        rpf-check fail-filter fw_su-in;
        no-redirects;
        filter {
            input sample;
        }
        address 193.11.0.41/30;
    }
    family inet6 {
        rpf-check fail-filter fw_su-in-v6;
        address 2001:6b0:dead:beef:2::0229/126;
    }
}

{master}
hugge@m1tug-re0>
```

So this is configuration for ”Stockholms Universitet” from old Optosunet (it will look the same in Sunet-C). Vlan 10 here is the uplink towards SU (the other is our vlan for our logical system) so it’s directly facing SU and is also the VLAN we run the BGP-peering in. Here we can see rpf-checks being configured with a fail-filter ”fw_su-in”



```
hugge@m1tug-re0> show configuration firewall family inet filter fw_su-in
term discard_martians {
    from {
        source-prefix-list {
            pfxl-martians;
        }
    }
    then {
        count martians-discard;
        discard;
    }
}
term allow_prefixes {
    from {
        source-prefix-list {
            pfxl-ncs-as2838;
            pfxl-ripe-as2838;
        }
    }
    then accept;
}
term allow_icmp {
    from {
        icmp-type [ unreachable time-exceeded echo-reply ];
    }
    then {
        policer icmp-policer;
        accept;
    }
}
term rest-discard_count {
    then {
        count "spoofed-discard;";
        syslog;
        discard;
    }
}

{master}
hugge@m1tug-re0>
```

In this filter we do a few things. Firstly we discard martians for obvious reasons, then we add a static allow-rule based upon the automatic BGP prefix-lists, which is generated every night from IRR-data. So the uRPF fail-filter follows the same scheme as the BGP-peer here, if we were to accept it through BGP then a RPF-check would never drop the traffic. There is also another source-prefix list entry here which is ”pfxl-ncs-as2838” and that is if we ever have the need to manually provision something in the prefix-list which isn’t in the route-object for any reasons (almost never used).

There is no reason really to not do this sort of filtering, the problem for multihoming has always been a popular excuse to not implement it, because it’s so cumbersome and there is a good potential to block legitimate traffic, but if carefully implemented its not a problem.

A good ”package” of how to be a responsible ISP is collected over at RoutingManifesto. It does not only cover anti-spoofing but also takes up a lot of very important things about routing-security. Incorrect route-advertisement and hijacks is brought up in the MANRS-document (Mutually Agreed Norms for Routing Security). I strongly encourage any potential customer to an ISP to follow-up if that ISP is validated towards MANRS. If all of the active operators on the Internet would adhere to the MANRS-framework we would see a much healthier Internet overall.

If there is interest – ill gladly go through in detail how we do to be able to validate towards MANRS Action point #1

”Action 1. Prevent propagation of incorrect routing information.”

But that’s for another day and another post. This is when the next topic is cued in, net neutrality.

Net-Neutrality is a hot topic as of late, especially in Sweden with a few ISPs testing the boundaries of what is to be considered ”neutral and free Internet”. Telia and Tre has been in the press lately when offering zero-rating on streaming of certain music-services and zero-rating on certain social-media. We’ll see what PTS has to say about that.

That’s not what i wanted to discuss though, BEREC (Body of European Regulators of Electronic Communications) has made some good progress on providing draft Guidelines for European ruling regards Net-neutrality. Most of the work is surprisingly good and adhere to the side of the ones that likes a free and open Internet (which we do). There is one point that is about overlooked though, and that is that not all filtering is bad. Some filtering is necessary to keep a healthy Internet, and one of them is anti-spoofing filtering (BCP38).

Below is an open letter as response to the draft that BEREC has produced regarding new ruling for net neutrality in Europe. This letter is initially originated from our friends over at JISC and it will be sent over to BEREC to consider changing part of the draft guidelines regarding net neutrality in Europe. Feel free to copy it yourself and send it over to BEREC before the 18th of July.

SUMMARY

We welcome BEREC’s recognition of the importance of filtering to protect the security of internet services and users. However, while most of types of filtering identified in paragraph 80 of the draft guidelines can be implemented in response ***to*** *a particular threat to a network, this is not true of filtering to protect other networks* ***from*** *threats created by the filtering network’s own users.*

BEREC’s draft guidelines identify one such class of filtering – spoofed addresses. This type of filtering can, as discussed below, only be done by the networks that originate traffic. The Internet Engineering Task Force has long considered it Best Current Practice against denial of service attacks, as documented in their BCP-38. This recommends that all networks be permanently configured to detect and block packets with spoofed source addresses, before they leave the originating network. This recommendation is promoted by network operators (for example the FENIX group in the Czech Republic) and regulators (for example FICORA in Finland). We are concerned that the draft regulations, by stating that permanent filtering should be considered a breach of network neutrality, would seriously harm these efforts to protect the security and stability of networks and services.

Permanent filtering of spoofed addresses is not only an effective way to reduce the opportunity to conduct denial of service attacks, it also distinguishes very precisely between legitimate and non-legitimate traffic. Unlike other types of security filtering it should not, therefore, affect network neutrality in any way. The only packets that will be blocked are those that, either accidentally or deliberately, do not conform to the fundamental Internet Protocol standard. Computers sending these packets would not, even on an unfiltered network, receive any internet service, since the response packets would never reach them. Filtering spoofed packets will have no effect on the computers sending those packets and only beneficial effects on the rest of the network.

We therefore encourage BEREC to recognise this type of filtering as not constituting a breach of network neutrality.

DISCUSSION

Most Internet denial of service attacks use a technique known as amplification (see Arbor Networks, "Worldwide Infrastructure Security Report, Volume XI (2015)", page 24). This has been compared to the attacker asking for a mail order catalogue to be sent to the victim: by sending small postcards to a legitimate third party the attacker can create a much greater load on the victim's mail delivery service.

The Internet version of the technique likewise involves an attacker sending a small message to a third party that causes that third party to send a much larger message to the victim. The Network Time Protocol (NTP) can generate responses 500 times larger than the request, many other services provide amplification factors of more than 100. Most attackers use compromised computers to send their request packets, obtaining a further level of amplification. A single command sent to a few hundred compromised computers, each of which generates amplification requests at the speed of a typical ADSL connection, can generate flows of tens or hundreds of gigabytes per second to the chosen victim. This is sufficient to fill the connection of almost any organisation, resulting in the victim's website and other services becoming inaccessible to legitimate visitors. Such attacks may be used, for example, for blackmail, activism, online gaming advantage, or to distract the victim from other hostile activity. By congesting other networks, their side-effects can cause instabilities across a wide area.

Amplification attacks are particularly hard for the victim to deal with, as the packets they receive are completely normal and come from legitimate sources. Any filtering that the victim or their network provider can implement in response to the attack will inevitably block legitimate traffic as well as the packets forming the attack. Similarly the services that are used for the amplification receive apparently normal requests, though perhaps at an increased rate, and respond in the normal way.

The only parties that can distinguish the packets involved in an attack are the networks that connect the compromised computers, controlled by the attacker, to the Internet. For the responses to be sent to the victim, the request packets must appear to come from the victim. Request packets are therefore sent with a "spoofed" source address – that of the intended victim – rather than the true addresses of the computers that generate them. This is the only point in the attack where abnormal packets are used, and where they can be accurately distinguished from legitimate traffic. Networks connecting users or organisations to the Internet should know which IP address ranges those users or organisations legitimately use: any packet that has a source address outside these ranges can be detected and blocked. Once packets reach transit providers the diverse connectivity of the Internet makes it practically impossible to distinguish those having a source address that does not match their network of origin.

Blocking spoofed packets was identified as Best Current Practice against denial of service attacks by the Internet Engineering Task Force in 2000. Wider adoption of this recommendation, known as BCP-38, has been encouraged by many global and national campaigns, including the Internet Society's Mutually Agreed Norms for Routing Security (2014) and Czech Internet Service Providers' FENIX project (2013). The Finnish telecommunications regulator, FICORA, makes BCP-38 implementation mandatory for ISPs in Finland.

Unlike filtering by the victim of a denial of service attack, address spoofing filters must be in place permanently. The volume of spoofed traffic from any individual computer is unlikely to be sufficient to trigger its Internet Service Provider's alarms; other networks or services suffering from the attack cannot notify the source ISPs because the address spoofing prevents them identifying the source of the packets. BEREC's requirement that filters be enabled only in response to a particular threat is therefore likely to reduce (or at best slow down) the adoption of this important protection technique.

This would be a particularly unfortunate outcome of regulation designed to protect network neutrality, as filtering spoofed addresses is the most neutral technique available to prevent denial of service attacks. As discussed above, any filtering by the victim will inevitably also block legitimate packets, so will interfere with some genuine use of the network. By contrast, packets with spoofed source addresses can never form part of genuine use, because the responses will never reach the computer that originated them. Spoofed packets can only be created accidentally, through a misconfiguration of the sending computer, or maliciously. In the former case the computer will not receive services from the Internet whether or not its packets are filtered by its ISP. Such filtering therefore makes no difference to users' Internet experience and has no impact on network neutrality.

SUNET

The Swedish University Computer Network (SUNET) is the National Research and Education Network in Sweden. Since the early 1980s SUNET interconnects Swedish universities and academic institutes all over the country in a nation-wide high-speed computer network. SUNET is governed by the Swedish Research Council (Vetenskapsrådet). Through partnership with the Nordic University Network (NORDUnet) SUNET users are provided with high-quality data communications to academic resources and research networks throughout Europe as well as to the general Internet. SUNET was founded in 1980, initially as a research project for Swedish computer scientists. A decade later SUNET paved the road for the establishment of the general Internet in Sweden.

Maria Häll, SUNET CEO
Per Nihlén, SUNET CTO

Skriven av

FREDRIK “HUGGE” KORSBÄCK

Network architect and chaosmonkey for AS1653 and AS2603. Fluent in BGP hugge@nordu.net