Are BGP Routers Open to Attack? An Experiment

Ludovico Cavedon, Christopher Kruegel, and Giovanni Vigna

University of California,
Santa Barbara
{cavedon,chris,vigna}@cs.ucsb.edu

Abstract. The BGP protocol is at the core of the routing infrastructure of the Internet. Across years, BGP has proved to be very stable for its purpose. However, there have been some catastrophic incidents in the past, due to relatively simple router misconfigurations. In addition, unused network addresses are being silently stolen for spamming purposes. A relevant corpus of literature investigated threats in which a trusted BGP router injects malicious or wrong routes and some security improvement to the BGP protocol have also being proposed to make these attacks more difficult to perform. In this work, we perform a large-scale study to explore the validity of the hypothesis that it is possible to mount attacks against the BGP infrastructure without already having the control of a "trusted" BGP router. Even though we found no real immediate threat, we observed a large number of BGP routers that are available to engage in BGP communication, exposing themselves to potential Denial-of-Service attacks.

1 Introduction

The *Border Gateway Protocol* (BGP, [19]) is the routing protocol at the core of the Internet. BGP is employed to perform routing decision between *Autonomous Systems* (ASes), which are separate administrative domains. Directly connected Autonomous Systems establish *peering* relationships. A peering relationship implies full trust: one router will accept and use for routing any network prefix advertised by its peer routers (unless administratively forbidden).

This full trust between peers is one of the weaknesses of the protocol. In fact, unconditional acceptance and propagation of routing information coming from other peers might render the whole Internet routing stability vulnerable to malicious, compromised, or just misconfigured BGP routers. According to Mahajan et al. [13], BGP misconfiguration is quite common (up to 1% of the global BGP table entries), although only 4% of these misconfigured announcements result in disrupted connectivity. Nevertheless, wrongly advertised prefixes sometimes gave place to well-known catastrophic routing incidents. For example, Pakistan Telecom disrupted worldwide routing to YouTube's web site in an attempt to prevent access to that web site in Pakistan [20]. Another example is the AS7007 incident [2], where the wrong propagation of routes caused an Internet-wide

J. Camenisch, V. Kisimov, and M. Dubovitskaya (Eds.): iNetSec 2010, LNCS 6555, pp. 88–103, 2011.

blackout. In addition, an AS might generate malicious BGP prefix advertisements in order to hijack some IP addresses and use them as not-yet-blacklisted sources of spam, as reported by Ramachandran and Feamster [18] and analyzed by McArthur and Guirguis [14].

Previous literature has been investigating possible areas of attack against the BGP protocol (see, for example, [15] and [3]). Pilosov and Kapela [17] also gave a live demonstration about the feasibility of an unnoticed *Man-In-The-Middle* (MITM) attack performed by means of BGP prefix hijacking.

Much work has focused on finding solutions to the above-mentioned attacks. For example *S-BGP* ([11]), *soBGP* ([10]), and *IRV* ([8]) are BGP extensions aimed at authenticating prefix origins and updates. However, most of the proposed solutions imply a heavier load on the CPU and memory of routers in order to perform cryptographic operations. Moreover, changes to the protocol need to be backward compatible, as it not possible to replace all the routers software at once. For these reasons, adoption of the defense techniques proposed so far has been extremely slow.

Almost all of the successful attacks against BGP considered in the literature require the attacker to have control of a BGP router. This condition, however, is not easy to achieve and maintain, while having a malicious behavior. In fact, the attacker has to demonstrate the need to become an AS (i.e., at least being a multi-homed network), or must be supported by the complicity of an ISP (which is willing to accept the risk of damaging its own reputation), or needs to compromise an existing BGP speaker (but router exploits are rather rare), or requires to hijack an existing TCP connection between two BGP routers (which is very hard to carry out).

However, there is an unanswered question: Is it possible to mount attacks in order to disrupt inter-domain routing without already having the control of a "legitimate" BGP router? In this work, we perform a large-scale study to explore the validity of this hypothesis.

First, we tried to identify how many BGP speakers were reachable on the Internet, by performing a SYN scan for TCP port 179 over a very large part of the Internet address space (about 73%). The scan was performed between December 2008 and January 2009. 2.2 million hosts (0.8‰) answered to our SYN packets, identifying an equal number of processes listening on that TCP port. Clearly, there is no implication that those hosts were BGP speaker. Restricted to this subset of IP addresses, two additional scans (performed in February 2009 and February 2010) went further in the connection negotiation, aiming at detecting whether the counterpart was willing to continue after a 3-way handshake and even establish a BGP session.

In the rest of this paper, we present our findings and we describe the issues we encountered with such a large-scale scanning experiment.

2 BGP Basics

The *Border Gateway Protocol* (BGP) is specified in RFC 4271 [19]. BGP is a path-vector routing protocol used across *Autonomous Systems* (ASes) on the

Internet. BGP routers exchange information about reachable destinations (under the form of CIDR IP prefixes) with an associated path to traverse in order to reach them. The information collected by the BGP process is employed to perform routing decisions between ASes. In most setups, the BGP routers of an AS exchange prefix vectors only with the BGP routers of neighbouring ASes (i.e., directly connected ASes). In such cases, the two BGP routers are said to be in a *peering* relationship. Peering relationships between routers of two different AS are also called *eBGP* (*external* BGP) peerings, in order to differentiate them from connections between BGP routers within the same AS, called *iBGP* (*internal* BGP) peerings. A peering relationship implies full trust: one router will accept and use for routing any network prefix advertised by the other router (unless administratively forbidden).

Peering relationships are carried out using a TCP connection on port 179. Upon establishment of such connection, the two BGP routers exchange an *OPEN* message, which contains identification parameters like the AS number and the BGP identifier (usually the IP address assigned to one of the network interfaces) of the sender.

In case a BGP speaker accepts the peering request (on the basis of the remote IP address of the TCP connections and the information contained in the *OPEN* message), it sends to the remote party a series of *UPDATE* messages, containing the description of the prefixes that the router is configured to advertise. At the same time, the router listens for route updates form the other peer, and stores them in its *Routing Information Base* (RIB) according to the configured policy.

The lifetime of the peering relationship is bound by the TCP connection: if such connection is closed or lost, all the routing information learned via that connection is removed from the RIB. The connection is checked for liveliness with periodic *KEEPALIVE* messages.

Any error condition, for example during the *OPEN* negotiation or in *UPDATE* messages, is notified by sending a *NOTIFICATION* message, and the TCP connection is subsequently closed. Instability of a BGP connection might cause a potentially big set of routes to be cyclically added and removed from the RIB. This would have the effect of increasing the load on the router and on the network (as route *UPDATE*s would be consequently propagated). In order to overcome this problem, *Route Flap Dampening* (RFD, [22]) has been defined, which temporarily suppresses unstable routes. As it will be discussed in the next section, an attacker can exploit this functionality in order to disrupt connectivity.

In some cases, it may be required to establish an eBGP peering between two routers that are not directly connected, creating so called *multi-hop* BGP sessions. This may be the case, for example, when the two peers communicate via a device that does not support BGP or via links for which static routes have been set.

The main authentication mechanism for BGP is via the remote IP address, the router identifier and the AS number. Clearly this information is not enough to defend against spoofing attacks. BGP also provides an *Authentication* option in the *OPEN* message, which would allow for remote peer authentication, but would not defend against TCP-level attacks. Moreover, this option has never

found widespread usage. In fact, in the latest BGP specifications the *Authentication* options has been deprecated in favor of TCP MD5 signatures [9]. When two peers are communicating using TCP MD5 signatures, every TCP packet they exchange must carry a specific TCP option containing an MD5 hash of the packet concatenated with a shared secret. The value of the shared secret must be manually specified in the configuration of both routers. In this way, an attacker who does not know the secret cannot establish a spoofed peering session or hijack/reset an existing TCP connection, as non-signed packets are discarded.

3 Attacks against BGP

The biggest weakness of BGP is the trust that a router has toward its peers (see Fig. 1). Any route that is advertised by a neighboring BGP speaker is merged in the routing database and is propagated to all the other BGP peers. Configuration mechanisms for filtering incoming and outgoing prefix advertisements are available, but clearly having a system administrator manually specify acceptable prefixes mostly defeats the purpose of having a routing protocol like BGP. Moreover, the number of prefixes a BGP router is handling might be extremely large: the current the size of the *Forwarding Information Base* (FIB) of a BGP router is above 300,000 entries[1].

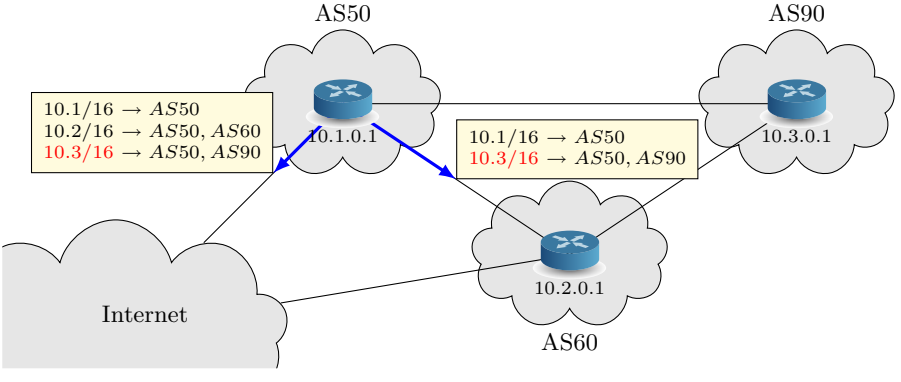

Fig. 1. Example of BGP prefix advertisement propagation. Two prefix advertisement messages sent from the router of Autonomous System AS50 are represented in the picture. AS50 is claiming to be able to reach the network 10.3.0.0/16 by being directly connected to AS90. AS60 and the other ASes on the Internet need to trust this piece of information from AS50.

3.1 Attacks from a Trusted Peer

Exploiting a trust relationship, a malicious or misconfigured router can claim ownership or reachability of network prefixes it is not entitled to.

[1] `http://bgp.potaroo.net/as1221/bgp-active.html`, as of February 2010.

Such advertised prefixes will then be spread potentially to all BGP routers and used to build routing tables, having the effect of directing IP packets towards the misbehaving router rather than the legitimate AS.

Hijacking traffic can be performed for multiple reasons. For example, an entity or malicious AS may want to make a subnet unreachable from other Internet hosts, or "blackhole" it. In this case, such AS can start advertising the prefix for the target network and then discard all the traffic it receives that is intended for that network. This is what happened in February 2008 in Pakistan [20], where the government decided to block access to the YouTube website, by having the main Pakistani telecom provider hijack YouTube network prefixes. However, a BGP misconfiguration caused the hijacked prefixes announcements to be propagated outside Pakistan boundaries, therefore blackholing YouTube's website across the whole Internet.

Attacks to the BGP infrastructure can be more sophisticated than blackholing. The ability of diverting traffic toward one's own AS allows for mounting *Man-In-The-Middle* (MITM) attacks, where traffic is sniffed or even modified. Pilosov and Kapela [17], in fact, gave a live demonstration about the feasibility of attracting most of the traffic destined to a target AS and than route it back to the intended AS, without users noticing it, and hiding the presence of the attacker from the output generated by routing analysis tools, such as Traceroute.

Probably, the most common reason for willingly stealing a network IP prefix is for spam purposes. Ramachandran and Feamster [18] observed that a percentage of spam emails (growing up to 10% from time to time) are sent from IP addresses belonging to short-lived prefixes (i.e., prefixes that are withdrawn within a day after being firstly advertised). In this way, spammers can obtain non-blacklisted new IP address to use; moreover keeping the duration of the prefix hijacking short does not allow the affected ASes to perform effective countermeasures.

Effectively stealing prefixes. If two ASes advertise the same prefix, the ASes across the Internet will be likely to be partitioned into two groups: ASes which are closer to the legitimate one and ASes which are closer to the malicious or misconfigured one. Therefore, not all the traffic will be diverted to the misbehaving router.

In order to increase the effectiveness of an IP prefix hijacking attack, a commonly used technique is to advertise subnets with a longer netmask than the original netmask of the prefix to be hijacked. As BGP will build forwarding rules using the longest-prefix matching criteria, the newly advertised rules will override the original one.

On the other side, the longer prefix matching functionality can also be exploited by advertising shorter prefixes: in this way the advertiser will steal unused subnets without disrupting the routing of actively-used prefixes.

3.2 Attack Which Do Not Need Control of a Trusted BGP Router

If an attacker does not control a trusted peer, some other attacks are possible at the TCP level. First of all, a malicious node able to sniff traffic between

two BGP peers could be able to hijack the TCP connection and inject its own *UPDATE* messages. Achieving favorable conditions for this attack, however, is difficult to achieve, given that usually eBGP routers are connected via dedicated point-to-point links.

BGP is also very sensitive to *Denial-of-Service* (DoS) attacks, much more than other TCP-based protocols. This is due to the fact that, by specification, all the routes learned from a peering session are withdrawn as soon as the associated TCP connection is lost. Therefore, the ability for an attacker to reset a TCP connection or overload/reboot a router can have a significant effect. Moreover, if the connection is being reset multiple times, the *Route Flap Dampening* (RFD) protection mechanism of BGP will suppress the flapping routes for a prolonged length of time after the end of the attack.

A TCP Reset attack consists in sending a TCP packet with the *RST* flag set to one of the connection endpoints. Successfully performing this attack requires guessing the source port number of the TCP connection (which is generally non-randomly chosen), but does not require an exact guess of the sequence number. In fact, as shown by Watson [23], with an outgoing bandwidth of 1.4 Mbps it is possible to perform a successful TCP connection reset in about 10 seconds against a target with a 63 KB TCP window size.

TCP MD5 signatures [9] were introduced to protect against TCP hijacks and resets, and started being adopted by some ASes in the years 2003-2004. Still, as discovered by Convery and Franz [6], some implementations were performing the MD5 signature validation before checking for the correctness of other TCP fields (e.g., sequence number and source IP address), therefore significantly affecting the CPU usage in case of TCP DoS flooding attack. Moreover, in the same study, they showed that an attacker with the ability to sniff MD5 signed TCP packets could recover a 6 character password in 3.5 hours.

Finally, the last two more generic channels for attack should be considered: if the router accepts BGP connections (even if it later declines the peering request), DoS by resource exhaustion could be an option for an attacker. Alternatively, malformed packets could be used to attempt to crash the router or even exploit it to gain some kind of control over it. Nevertheless, these types of vulnerabilities are rare, and they are very hard to exploit.

4 Scanning the Internet for BGP Routers

The goal of our experiment was to scan the whole Internet for open BGP ports (179) and gather information about how many BGP routers are reachable and how they react to our probes. Our question was: "Is it possible for an attacker who does not already have control over a legitimate, trusted BGP router to affect the BGP infrastructure?"

Scanning the whole Internet address space is an activity that poses some challenges by itself. The number of IP addresses to probe is about 3.7 billions, which means that in order to complete the scan in 2 weeks, with maximum 1 retry per IP address, we need a constant outgoing packet rate of 6,000 SYN packets per second.

Such activity is very noisy, both from the side of the scanner's ISP and the side of the ASes receiving the scan probes. In particular, the port 179 is notoriously a very low traffic port, and, independently from how slowly the scan is performed, a whole network sweep on that port is going to be easily detected. Moreover, scanning activity is usually identified as malicious, and sometimes finds a hostile response from network administrators.

We took a number of precautions in order to reduce the impact of our scanning activity on remote networks, and we tried to address in advance potential complaints about our scanning. First of all, we setup a web page served by an HTTP server reachable at the IP address originating the scan. This web page was explaining who we were, what we were doing, and making it clear that no malicious activity was to be performed by means of our scan.

Moreover, we offered the scanned party the opportunity to contact us and request to be excluded from the rest of our scans.

Second, we minimized the incoming packet rate seen from target networks by spreading the scan of the whole network across the whole duration of the experiment. In other words, we cycled over the IP address space by incrementing the most significant bytes (network-bytes) first.

During our scanning activity we received complaints from 10 different institutions, asking to be excluded from further scans. Only one other institution (AT&T) contacted us letting us know that they were aware of our activity (see Fig. 2) and that they were interested in hearing about the details of our experiment.

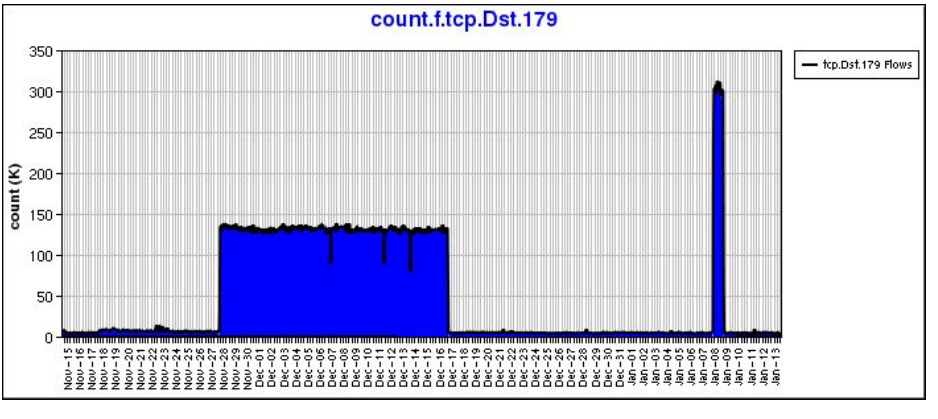

Fig. 2. Numeber of TCP flows per hour as detected by AT&T on their network (/8 prefix). The picture clearly marks two phases of our scanning activity at two different outgoing packet rates. It it noteworthy how scanning on port 179 is extremely more noisy that normal activity on that port.

One problem we incurred in was that, because of the restructuring of our lab network, we needed to change the source IP address used during the experiment. Such change was perceived by some network administrators as an evasion

attempt. These administrators had actually noticed our activity and silently added a firewall rule to block our original IP address. This issue caused our IT department to request the termination of the scanning activity. The data we collected up to that point was about 73% of the whole address space, from X.X.0.0 to X.X.186.249.

Given the noisiness of such an experiment and the perceived blurred boundary between benign and malicious scanning, such a large-scale scanning constitutes a one-shot experiment: it is not viable to repeat it, at least in the near future.

4.1 Scanning Tools

We first looked at the *Scapy* [1] tool, which allows great flexibility for manipulating network packets. However, such tool does not allow handling an outgoing packet rate as high as we needed (6,000 packets per second), because of its excessively slow data handling.

We therefore decided to use *Nmap* [12] for the Internet-wide scan of hosts. Nmap was able to handle the required rate on a quad-core Intel Xeon X3220 2.4 GHz with 4GB of RAM and 1 Gbps connection to the Internet. With Nmap, we generated a report of which hosts had the TCP port 179 open.

Even though we did not have major issues, with some hindsight, we now think that implementing an *ad hoc* tool for large-scale SYN scanning would have provided us with more accurate results. Nmap has an adaptive mechanism to adjust scanning parameters, which tries to learn network delays and congestion from previous probes. While sweeping a broad range of networks, however, adaptive learning does not work well, causing unexpected (and incorrect) timing estimation (such as network RTT and optimal inter-packet sending time).

In addition, the parallel scanning of hosts performed by Nmap was not optimal. The ideal technique would have been sending out SYN packets at a chosen rate while recording possible answers. Allowing a timeout of maximum 1.5 seconds, with at most 1 retry requires one to keep in memory the state for up to 9,000 IP addresses at the same time. Unfortunately Nmap did not allow keeping a sliding window running over all the range of IPs, and, instead, it required scanning the hosts in batches. Therefore, we set to the batch size to 15,000 IP addresses, therefore alternating peaks of high scanning rate and moment of limited outgoing traffic.

5 Probing BGP Routers

Once we collected the list of IP addresses answering to our SYN packets, we tried to engage in a BGP exchange with them. Given the much smaller number of hosts to contact (about 2.2 millions), it was possible to write a Python tool for that purpose. We based out tool on the *Announcer* application by Colitti [5]. *Announcer* is an implementation of a basic BGP speaker written in Python, without any actual routing feature. We borrowed the code for generating and parsing BGP messages, while we wrote our own BGP state machine.

We took an approach that was as passive as possible. In more details, we did the following:

1. opened a TCP connection toward port 179;
2. waited for the remote peer to send any data (5 seconds timeout);
3. in case of no data received or *OPEN* request received, we sent a BGP *OPEN* request, using the UCSB AS number (AS131) and our IP address as BGP identifier;
4. waited for the remote peer to send any data (i.e., *OPEN* message, if not already sent, or *UPDATE* messages);
5. sent a *KEEPALIVE* message and waited for reply;
6. terminated the peering session.

At all times, we monitored and recorded any event that would cause an unexpected session termination:

- *timeout*, either during connection establishment or waiting for data from the remote peer (the timeout was set to 5 seconds);
- TCP *RST* packet, indicating a connection refusal or abrupt connection termination;
- TCP *FIN* packet, indicating the intention of the peer to close the connection;
- ICMP error, indicating unreachability of the remote host or port;
- BGP *NOTIFICATION* message, indicating an error during the establishment of the peering session;
- non-BGP data read from the socket, indicating the presence of a non-BGP speaker on that port.

6 Results

During our initial TCP SYN scan (carried on between December 2008 and January 2009) about 2.2 million hosts where reported to have the TCP port 179 open (i.e., 0.8‰ of the scanned IP addresses).

This pool of potential BGP routers was than probed with our Python BGP speaker in February 2009. Quite unexpectedly, 35% of these hosts no longer responded to our BGP connection requests. A reason for this behavior can be inferred by looking at the distribution of the IP addresses that were leading to timeout in the second scan: 90% of such IP addresses were belonging to less than 1% of /16 networks in the scanned address space. In fact, we found a surprisingly large number of subnets that contained a large number of hosts accepting TCP connection on port 179. The initial Nmap scan was hitting these networks at a relatively low rate, as the whole Internet address space was being covered breadth-first. When we later processed the IP addresses with the open port, these peculiar networks started receiving a much higher rate of incoming SYN packets (up to 50 packets per second). The likely explanation for the increased number of observed timeouts is the triggering of rate-limiting mechanisms in the target networks. Such networks are probably honeypots, employed to monitor malicious activity on the Internet.

The same BGP probes on the same list of IP addresses was also repeated one year later, in February 2010. As could be expected, the number of hosts timing out at our connection attempts increased by 34%. Among the IP address that resulted in a timeout in February 2009, 85% were also present in the latest scan.

Similarly to the variation in number of hosts timing out, we also observed a doubling of the hosts no longer reachable or refusing the TCP connection between the two BGP scans. Among the hosts concluding the 3-way handshake, though, the distribution of observed behaviors remained pretty much the same. Therefore, we will concentrate the results of the second scanning experiment (Table 1).

Table 1. Results of February 2010 scan. The first column indicates the last packet we sent during our probe, the second column indicates the last packet received from the remote host (or the firing of a timeout in case no data was received). The fourth column indicates the percentage over total probed IP addresses, while in the last column the percentage is computed over the hosts that completed the TCP 3-way handshake.

After we sent	*event/response*	*number of IPs*	*% of probed*	*% of conn.*
SYN	timeout	1026798	47.40	-
	TCP RST	55223	2.55	-
	UNREACH	142257	6.57	-
SYN+ACK	TCP RST	24484	1.13	2.60
	TCP FIN	402381	18.57	42.71
	BGP NOTIF CEASE	8787	0.41	0.93
	subcode 0	8559		
	subcode 5	228		
	non-BGP data	297	0.01	0.03
BGP OPEN	timeout	441130	20.36	46.83
	TCP FIN	51631	2.38	5.48
	BGP NOTIF OPEN	7843	0.36	0.83
	subcode 2	973		
	subcode 3	29		
	subcode 4	1444		
	subcode 5	5391		
	subcode 7	6		
	BGP OPEN & UPDATE	5	<0.01	<0.01
	non-BGP data	4994	0.23	0.53

Of the host that completed the TCP 3-way handshake, 45% closed or reset the connection right away. This behavior is probably determined by the interaction between a user space process and the router kernel for the management of the TCP connection. In fact, if the BGP speaker code is running as a user process, the 3-way handshake is managed by the kernel, which is not aware of the IP addresses of the configured peers (unless some ad-hoc firewall rules have been setup). The actual decision on whether the host requesting a TCP connection has the right to do so is made in the user-space process, which is notified of the connection after the 3-way handshake. Only at this point the BGP process can decide to terminate the connection.

About 1% of the hosts immediately requested us to close the BGP connection with a BGP *CEASE NOTIFICATION*, while 0.5% started communicating with our tool sending non-BGP data. Interesting enough, about 4900 hosts answered with the signature of a popular Instant Messaging (IM) protocol. Similarly to the network behavior of honeypots, such IM servers where listening on all IP addresses belonging to multiple /16 subnets on all TCP ports. Other popular non-BGP speakers we found were SSH server or telnet-like services.

0.4% (7,904) of the hosts concluding the 3-way handshake parsed our *OPEN* message, but declined the BGP peering session with a *NOTIFICATION OPEN Error* message. Almost all these BGP speakers also sent us an *OPEN* message, therefore identifying themselves. They turned out to be actually only 1258 distinct routers (i.e., distinct BGP identifiers), belonging to 318 different ASes. Among them, 3497 routers declared themselves to be in AS numbers reserved for private use, lowering to 192 the number of public ASes for which BGP routers were discovered.

In addition, we found 5 routers (all of them from the same AS) who accepted our BGP peering request and started sending us *UPDATE* messages with BGP prefixes. At this point, theoretically, we should have been able to send prefix *UPDATE*s back to the BGP router. If malicious and accepted by the remote peer, such injected route information could be used to disrupt local connectivity or backhole a subnet, potentially world-wide (i.e., advertising a subnet with a non-routable address as *NEXT_HOP*). We decided not to attempt to send such an update, as that would have been an excessively aggressive experiment. We instead contacted the administrators of such routers, in order to report the potential security threat and to try to gather some information about those BGP speakers. It turned out that we were not actually dealing with some BGP routers, rather than with a *BGPDNS* [16] infrastructure. While those hosts were not propagating incoming BGP prefixes to other BGP routers, they were storing them in a database. The owner of those BGP speakers confirmed that their security had to be improved and sending them updates could have been a way to overload their database server.

For the sake of completeness, in our scan in February 2009, we found about other 400 peers that were apparently willing to initiate a BGP peering session with us. After some investigation, these hosts appeared to be all from the same network and were not actually BGP speakers. Rather, they were behaving as *reflectors*. More in detail, they were sending us back whatever data we were writing on the TCP connection. Similarly to honeypots, reflectors are used to capture and observe malicious behavior on the Internet.

6.1 Filtering Potential Honeypot Networks

As mentioned in the previous section, we found a surprisingly relevant number of networks with a high number of IP addresses answering to our SYN packets on port 179. Such behavior is an indication that most probably such TCP endpoints are not actually BGP speakers, rather they are likely honeypot networks, whose purpose is to monitor network activity on the Internet.

Based on this rationale, we tried to identify all the networks that had more than 32 hosts answering with SYN+ACK on port 179. 2.6% of all the networks[2] satisfied our criterion, covering 81% of all the probed IP addresses. We then excluded such IP addresses from the statistics on the scan of February 2010 and compared the results.

As we expected, the percentage of IP addresses being unreachable or leading to a timeout reduced to approximately half, while the percentage of hosts closing the connection right after the 3-way handshake almost tripled (59% of all non-filtered IP-addresses). These numbers seem to confirm the hypothesis we did at the beginning of the previous section about our BGP probes hitting some rate-limiting firewalls of these honeypot networks.

6.2 Areas of Attack

From the results of the experiments we described, we can identify some areas that can be the target of attacks from a malicious user who does not have control of a trusted BGP router. First, the router that engages in a BGP peering session with an arbitrary BGP speaker is the most exposed, possibly to route injection, but also to mishandling of malformed packets. In fact, Convery and Franz [6] were able to find 4 flaws on 3 different router vendors that allowed a BGP speaker peering with a target router to force a reset of the target router.

From a more general point of view, whenever unnecessary input is parsed or resources are allocated, there is some ground for attack. For example, all routers waiting and replying to *OPEN* messages might be vulnerable to some kind of malformed *OPEN* message, or to resource exhaustion attacks.

6.3 Areas of Defense

BGP is not a public service. Every network administrator knows which ASes their routers will be talking to, and the configuration of such peers is done manually. For this reason, protection against unwanted peers is quite simple. Denial of TCP connections should be performed as early as possible, during the 3-way handshake, with cooperation between the BGP process and the kernel and/or firewall.

Moreover, the standard mechanisms for BGP protection at the TCP level should be employed, i.e., TCP MD5 signatures [9] and the TTL Security Mechanism [7]. These features should be a requirement for BGP peers rather than an option.

7 Related Work

The security of BGP and Internet routing has been addressed by multiple researchers in previous literature.

[2] We used the "network" separation as provided by the IP2Location (`http://www.ip2location.com`) database. The actual number of networks (4510) satisfying our creterion is not relevant as an absolute number, as single big networks are sometimes split into smaller ones.

A survey of possible attacks against BGP has been made in 2004 by Nordström and Dovrolis [15], in order to raise awareness of potential security risks, and to observe how existing and proposed countermeasures would have not been actually deployable or would have not been effective enough.

In 2008, Butler et al. [3] provided an updated and extensive snapshot of BGP security issues and possible solutions. They observed how BGP has been so far successful in providing a good stability of Internet routing. Nevertheless, they underlined some areas where BGP is still believed to be in need of security improvements.

Most of the literature has focused in proposing solution to the attacks where the malicious user owned a trusted BGP router. One of these proposals is *S-BGP* ([11]), which adds multiple layers of security to the current BGP specifications. S-BGP allows a BGP router to validate the authenticity and integrity of each path received from a peer. S-BGP, in fact, requires BGP speakers to sign every *UPDATE* message. The ownership of a prefix by an AS, instead, is signed by the address assignment authorities. TCP/IP level security is guaranteed by requiring the use of IPSec. Adoption of S-BGP does not appear to be feasible, as it does not allow for incremental deployment over exiting BGP, and requires many other infrastructure modifications, like the introduction of a *Public-Key Infrastructure* (PKI) and IPSec support.

Another mechanism for route validation has been proposed by Goodell et al. [8] under the name of *Interdomain Route Validation* (IRV). This proposal, rather than a modification of BGP, is an external service that allows a BGP speaker to query other ASes and the address assignment authorities in order to validate the received paths. This approach requires a much lighter deployment effort than S-BGP, but poses some concerns of scalability and vulnerability to DoS attacks.

James ([10], *soBGP*) proposed to increase the security of BGP by adding an additional *SECURITY* BGP message which carries signed information about prefixes ownership and ASes connectivity, allowing for the creation of a distributed database which routers can use to establish validity of paths. Even though soBGP allows for incremental deployment, it is not being adopted because of the necessary changes in the router software and the increased requirements in terms of router memory and CPU time.

Chan et al. [4] developed an interesting study about the adoptability of secure BGP enhancements.

In their study about behavior of spammers with respect to network resources usage, Ramachandran and Feamster [18] observed how some spammers make use of BGP prefix stealing in order to acquire control of not-yet-blacklisted IP addresses. Unused IP addresses are stolen for a short amount of time (often less than 24 hours), in order to prevent the execution of effective countermeasures. Further analysis on prefix hijacking has been carried by McArthur and Guirguis [14].

As already described, the main concerns for BGP from the DoS attack point of view, are the consequences of BGP peering session teardown. Sriram et al. [21]

conducted a simulation study on the impact of such attacks on routing performance. They showed how the BGP routing infrastructure is non-negligibly affected by this kind of attacks, even while having a relatively low success ratio for BGP peering session attacks.

A very interesting study on BGP security has been performed by Convery and Franz [6]. Their work started from an analysis and categorization of all possible vectors of attack against the BGP protocol. As the second step, they tried to discover potential vulnerability in routers by means of protocol fuzzing and resource exhaustion attacks. In the end, they performed a scanning study very similar to the one we performed. They ran a traceroute towards each prefix advertised in the Internet routing tables in order to gather IP addresses of potential BGP routers. Afterwards, they tried to connect to all of these hosts on port 179, and they sent an *OPEN* message and listened for data. Our approach for collecting IP addresses of routers was performed on a much larger scale. In fact, we were able to conclude the 3-way handshake with 2.2 million hosts on port 179 (vs. 4,602). Clearly our number is inflated by the fact that we hit not only routers, but also some honeypot/reflector networks. However we were able to reach a broader number of BGP speakers, not only those we could encounter on a path from our network towards all ASes on the Internet. In fact, in February 2009, about 16,000 hosts answered our OPEN request (vs. 1,750), and we found that 7 of them were willing to establish a BGP peering relationship with us.

8 Conclusions

With our experiment we took a snapshot of a very large subset of the Internet, looking at how BGP routers answer to a casual inquirer.

We described the challenges we faced for performing such large-scale scanning experiment, and what we learned. Such activity revealed itself to be very noisy and perceived as extremely hostile. Therefore, one performing a large-scale scan must be able to handle complaints, no-scan requests, and be aware that she may not be able to repeat the experiment.

Based on how the BGP protocol works and on related work on the subject, we identified some possible areas of attack to the BGP infrastructure. We then tried to match them to the results of our scan.

We could not find any real immediate threat. Nevertheless, we found a large number of BGP speakers that would establish a connection with us and possibly accept some input from us. Such routers were not required to engage in such communication, and, in doing so, they exposed themselves to a potential menace, as BGP is extremely sensitive to Denial-of-Service attacks. Moreover, given the amount of trust given to the other peers, a compromised router could seriously affect the whole Internet routing infrastructure. For this reason, it is very important that some equivalent of the principle of *least privilege* is applied also to the acceptance of BGP connections, shielding the routers as much as possible from potentially malicious traffic.

Acknowledgments. This work has been partially supported by the National Science Foundation through grants 0905537 and 0716095, and by the Office of Naval Research through grant ONR N000140911042. We would also like to thank AT&T for their feedback on our scanning activity.

References

1. Biondi, P.: Scapy (2009), http://www.secdev.org/projects/scapy/
2. Bono, V.J.: 7007 Explanation and Apology (April 1997), http://www.merit.edu/mail.archives/nanog/1997-04/msg00444.html
3. Butler, K., Farley, T., McDaniel, P., Rexford, J.: A survey of BGP security issues and solutions. AT&T Labs Research (2008)
4. Chan, H., Dash, D., Perrig, A., Zhang, H.: Modeling adoptability of secure BGP protocol. ACM SIGCOMM Computer Communication Review 36(4), 290 (2006)
5. Colitti, L.: Active BGP Probing (2009), http://www.dia.uniroma3.it/~compunet/bgp-probing/
6. Convery, S., Franz, M.: BGP Vulnerability Testing: Separating Fact from FUD. In: Black Hat US 2003 / NANOG28 Meeting (2003)
7. Gill, V., Heasley, J., Meyer, D.: The Generalized TTL Security Mechanism (GTSM). RFC 3682 (Experimental) (February 2004), http://www.ietf.org/rfc/rfc3682.txt; obsoleted by RFC 5082
8. Goodell, G., Aiello, W., Griffin, T., Ioannidis, J., McDaniel, P., Rubin, A.: Working around BGP: An incremental approach to improving security and accuracy of interdomain routing. In: Proc. NDSS, vol. 3 (2003)
9. Heffernan, A.: Protection of BGP Sessions via the TCP MD5 Signature Option. RFC 2385 (Proposed Standard) (August 1998), http://www.ietf.org/rfc/rfc2385.txt
10. James, N.: Extensions to BGP to support secure origin BGP (sobgp). Network Working Group, Cisco Systems (2002)
11. Kent, S., Lynn, C., Seo, K.: Design and analysis of the secure border gateway protocol (S-BGP). In: Proc. of DISCEX 2000 (2000)
12. Lyon, G.: Nmap – Free Security Scanner For Network Exploration & Security Audits (2009), http://www.nmap.org
13. Mahajan, R., Wetherall, D., Anderson, T.: Understanding BGP misconfiguration. In: Proceedings of the 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications, pp. 3–16. ACM, New York (2002)
14. McArthur, C., Guirguis, M.: Stealthy IP Prefix Hijacking: Dont Bite Off More Than You Can Chew. In: Proc. ACM SIGCOMM (2008)
15. Nordström, O., Dovrolis, C.: Beware of BGP attacks. ACM SIGCOMM Computer Communication Review 34(2), 1–8 (2004)
16. Oppermann, A., Jeker, C.: BGPDNS, Using BGP topology information for DNS RR sorting a scalable way of multi-homing. RIPE 41 Meeting (2002)
17. Pilosov, A., Kapela, T.: Stealing The Internet. DefCon 16 (2009)
18. Ramachandran, A., Feamster, N.: Understanding the network-level behavior of spammers. ACM SIGCOMM Computer Communication Review 36(4), 302 (2006)
19. Rekhter, Y., Li, T., Hares, S.: A Border Gateway Protocol 4 (BGP-4). RFC 4271 (Draft Standard) (January 2006), http://www.ietf.org/rfc/rfc4271.txt

20. RIPE NCC: YouTube Hijacking: A RIPE NCC RIS case study (2008), http://www.ripe.net/news/study-youtube-hijacking.html
21. Sriram, K., Montgomery, D., Borchert, O., Kim, O., Kuhn, D., et al.: Study of BGP Peering Session Attacks and Their Impacts on Routing Performance. IEEE Journal on Selected Areas in Communications 24(10), 1901 (2006)
22. Villamizar, C., Chandra, R., Govindan, R.: BGP Route Flap Damping. RFC 2439 (Proposed Standard) (November 1998), http://www.ietf.org/rfc/rfc2439.txt
23. Watson, P.: Slipping in the Window: TCP Reset attacks (2004)