Through the Looking-Glass, and What Eve Found There

Luca Bruno, Mariano Graziano, Davide Balzarotti, Aurélien Francillon
EURECOM
{*firstname.lastname*}*@eurecom.fr*

Abstract

Looking-glasses are web applications commonly deployed by Autonomous Systems to offer restricted web access to their routing infrastructure, in order to ease remote debugging of connectivity issues. In our study, we looked at existing deployments and open-source code to assess the security of this critical software. As a result, we found several flaws and misconfigurations that can be exploited to escalate from a web attack to a remote command execution on backbone routers.

This paper summarises the results of our study, and shows how even an attacker with very limited resources can exploit such flaws in operators' networks and gain access to core Internet infrastructure. Depending on systems configuration, these attacks may result in traffic disruption and global BGP routes injection, with severe implications for the security of the Internet.

1 Introduction

The Internet is composed by a large number of Autonoumous Systems (AS) which cooperate to exchange and carry data across their links. Several intra- and extra-AS routing protocols running on backbone routers are responsible for distributing routes in the control plane, across the world. Some of those protocols, however, have not been designed with security in mind and are not specifically resilient against malicious agents [1].

For example, the Border Gateway Protocol (BGP) [2] takes care of extra-AS routes distribution, but any malicious or wrongly configured AS can hijack and re-route prefixes owned by other ASes. Therefore, most of Internet routing relies on the assumption that no malicious BGP routers are ever allowed to announce bogus routes, and that the existing routers are benign and properly secured.

The aim of our study is to show how these assumptions do not hold true in the real-world, by focusing on a series of software flaws and widespread misconfigurations in "looking-glass" software that offers limited web-access to backbone routers.

The paper is organized as follow. In Section 1 we introduce the concept of "looking-glass" software as a public-access network debug tool, and its typical code architecture. Then, in Section 3 we outline a possible threat model, along with some of the most severe menaces. Furthermore, in Section 4 and 5 we present the results of the software review we did, and we describe the indirect experiments we performed to confirm our findings. The most relevant statistics and results of our experiments are shown in Section 6, along with an empirical rough estimation of BGP injection feasibility, based on historical records. Finally, Section 8 summarises our findings and give some insights on the current state of the Internet infrastructure.

2 Background

An AS infrastructure is composed of several network services, each handled by different systems and devices.

For the purposes of this paper, we will limit our focus to just two categories of systems that are strictly related to Internet routing: backbone BGP routers and Linux-based route servers.

- *Backbone routers*
  The worldwide Internet backbone is run on top of dedicated network devices capable of accelerated packet routing in the data-plane, using custom ASICs and dedicated hardware.

  These devices run a custom OS and control-plane stack which is responsible for computing the routing topology, e.g., by participating in BGP sessions with neighbors. In addition, all these devices have one or more interfaces for remote and out-of-band (OOB) administration, like a telnet service, a SSH service, or a remote serial port. The access to these

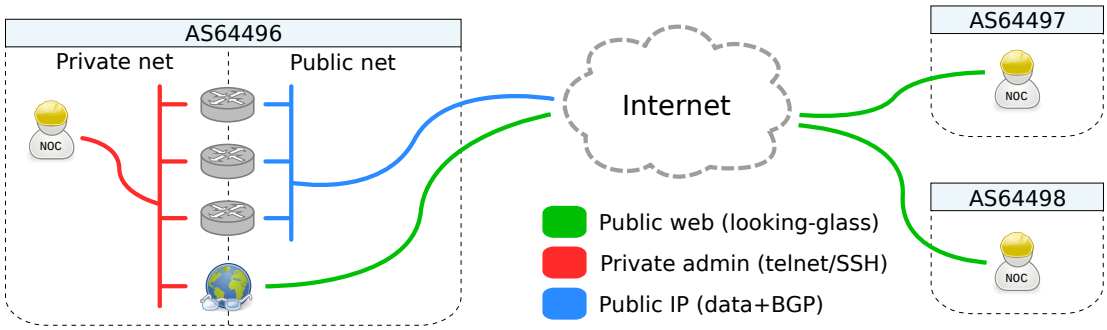

Figure 1: Looking glass architecture.

interfaces should be strictly restricted to Network Operations Center (NOC) operators and authorized AS personnels.

- *Route servers*
  Routing software exists for traditional Linux-based servers to establish BGP sessions with other routers and servers. Two prominent examples are Quagga [3] and Xorp [4], which are used by several operators and are under active development.

  The usages of route servers are multiple, from providing a read-only copy of the global BGP table to allowing scripting of BGP rules (e.g., by using traditional UNIX utilities). Also these servers can be accessed out-of-band by AS personnel, via telnet or SSH.

  Some public services exist, like the Route Views project [5], which provide unrestricted telnet access to their route servers to expose a read-only copy of the BGP table to analysts and researchers. However, those services are purposefully meant for public access, and are therefore outside the scope of this paper.

When debugging BGP routing problems, NOC operators are often facing issues affecting only a few ASes. Such problems are harder to debug due to the lack of a view on the remote routing table.

For this reason, a new category of web-applications emerged in the '90s to permit a restricted set of operations on AS routers and route servers by the large public, over the web. This kind of software is usually referred as "looking-glass", as it offers a local observation point to remote network engineers.

Looking-glasses are web scripts, usually implemented in Perl or PHP and directly connected to routers admin interfaces (i.e., telnet or SSH). These scripts are designed to relay textual commands from the web to the router and print back the router' replies. They run on top of common Linux/Apache stacks, and sometimes provide additional utilities for latency and traceroute measurements. Figure 1 briefly shows their typical architecture and deployment.

Looking glass	*Language*	*Release (date)*
Cougar-LG [6]	Perl	1.9 (2004-11-25)
Cistron-LG [7]	Perl	1.01 (1997-10-21)
MRLG [8]	Perl	5.4.1 (2007-08-30)
MRLG4PHP [9]	PHP	1.0.7 (2007-10-11)
Telephone [10]	PHP	1.2.0 (2012-10-01)

Table 1: Open-source looking-glass software.

We decided to focus our attention on looking-glass software, as most of them are small and old web-applications that have been last updated in early 2000s.

In this paper we analyse what we found to be the most commonly used open-source software looking-glasses, as listed in Table 1.

3 Threat model

A looking-glass is an often overlooked critical part of an operator infrastructure, as it sits at the border between the public web and restricted admin consoles. As such, the threat model encompasses both the typical web security scenario and some more custom networking threats.

We categorized some of the most relevant issues as follow:

- *Reverse Cross-Channel Scripting (Reverse-XCS)*
  Reverse-XCS are defined by Bojinov et al. [11] as "*the use of web interface to eventually attack a non-web channel*".

  In our scenario, this translates to two relevant cases:

  - *Malicious command injection*
    Bypassing a weak or non-existent commands

sanitization, an attacker may trick the looking-glass into sending malformed commands to a router console, e.g., to trigger a DoS on the control-plane.

  - *Routing Information Base (RIB) manipulation*
    By exploiting flaws in the looking-glass, an attacker may inject arbitrary commands to manipulate the RIB on the router, e.g., by changing the BGP configuration. If the attacked router relays its topology to neighboring ASs, this may also affect remote networks.

- *Web flaws*
  Typical web threats applies here too. In particular, we highlight two of the most relevant cases:

  - *Exposed routers credentials*
    The configuration files of a looking glass contain IP addresses, usernames, and passwords in cleartext. If not properly secured, an attacker may be able to gather credentials by guessing the URLs of configuration files.
  - *Cookies stealing via XSS*
    Even though looking-glass applications are usually unauthenticated, an attacker may exploit XSS flaws in them to gather admin cookies for other administration web-applications served under the same-origin domain.

4 Threat Analysis

4.1 Misconfigurations

Misconfiguration or improper access control of resources are two of the most basic, yet important, issues for web security.

First of all, if server modules are not properly configured, it may be possible to get a listing of supposedly private files – including source codes and configuration files. In this case, search engine bots are capable of crawling and inspecting the files content, thus making login credentials easy to gather by searching for ad-hoc Google-dorks [12].

Another possible issue comes from temporary files. In this case, source code and configurations could be recovered by looking for temporary editor files (e.g., ending in $\sim$) on the web server. This can as well help an attacker confirm which version of the software is run by a targeted operator, and may pose a bigger problem for proprietary/custom looking-glasses.

Moreover, configurations and login credentials for routers may be stored as plain text files in the same web server directory. By default, web-servers will serve them as plain-text to anyone querying the proper URL. By inspecting source code and looking at default file names, it is possible to guess the URL of configuration files and, if not properly protected, retrieve them.

Finally, some software allows advanced authentication methods, for example by using SSH public/private keypairs instead of cleartext passwords. In this case, the path to the SSH key is stored in the configuration files, instead of the full passwords. However, the SSH keys themselves could be stored by mistake in the same path, openly readable to the entire world.

4.2 Poor network policies

Backbone routers are parts of a critical infrastructure and as such their admin interfaces have to be properly secured. Cisco's own best practices [13], for example, recommend exposing consoles only over out-of-band loopback interfaces, unreachable from the public Internet (e.g., by using private addresses and placing them in dedicated admin VLANs).

However, some operators may decide to put loopback interfaces on publicly routable networks, or to expose administration services on all router's interfaces. In such cases, a remote attacker may be able to directly login onto them, e.g., by using stolen credentials.

4.3 Web security

It is usually recommended that actionable web applications make use of captcha or other automation blocker to avoid scripted attacks [14]. While this may not look like a big issue for looking-glasses, the lack of this countermeasure can effectively help an attacker to automatically map resources in an AS infrastructure and scan them for information gathering. This may also result in automated bot attacks aimed at DoSing connected devices or other parts of the network.

Moreover, an attacker may be able to exploit insufficient input parameters sanitization to perform XSS and inject HTML/JS code in web responses. While looking-glasses are usually unauthenticated, this flaw can be used to steal admin cookies for other panels hosted under the same-origin domain.

Similarly, it could be possible to perform a Reverse-XCS against the network infrastructure. If web input is not properly sanitized, an attacker may forge input parameters to inject custom commands into the router console. This directly means an escalation from a web attack to an Internet routing attack if the attacker manages to modify the router configuration.

Finally, some software may come with additional tools to be deployed directly on the web-server to perform advanced measurements (e.g., high granularity latency

measurements). Such tools may as well include vulnerabilities, leading to remote code execution on the looking-glass server itself.

4.4 Impact

Most of the threats described so far can result in an attacker getting unauthorised access to the administration interface of a router, under the same access level of the looking-glass software.

In a typical network topology designed as in Figure 1, this results into having an observation point into the private part of an AS infrastructure, and possibly manipulating portions of it. Moreover, an attacker with restricted console access to a router could easily elevate his privileges by cracking weakly-hashed secrets [15] (e.g., Cisco's ciphers Type-0, Type-5 and Type-4 [16]) or by abusing known authorization bypass vulnerabilities that affected several top vendors [17, 18].

The impact of a malicious attacker with elevated privileges on a backbone routers are manifold with respect to both the local AS infrastructure and the global Internet. For example, by altering internal or external routing configurations, an attacker may be able to blackhole or disrupt specific subnetworks, or set up traffic mirroring or re-routing scenarios for further attacks.

We consider the post-exploitation analysis of such a scenario to be complex and quite specific to single vendors, devices, and network setup. As a result, we will not cover post-exploitation details in the rest of this paper. However, we would like to stress the fact that backbone routers are usually capable of announcing routes both internally (e.g., into an OSPF domain) and to the whole Internet (i.e., to peering ASes via BGP). As such, in this threat analysis we have highlighted one possible path for a remote attacker with modest resources to escalate from a web attack, to a remote command injection into multiple backbone routers, to injecting malicious announces into the Internet BGP table.

5 Experiments

Given the theoretical attack surface presented in Section 3, we tried to assess how many ASes worldwide were actually vulnerable to remote attackers. Our goal was complicated by the fact that, due to ethical and legal considerations, we could not perform direct experiments, e.g., by injecting commands on remote routers or by just trying to login into them.

5.1 Ethical concerns

Unauthorized access to computers and network devices is prosecuted by several national and international laws. Since in this paper we conducted a comprehensive study, encompassing 26 countries with different legislations, we took ethical considerations as a top priority. For this reason we avoided any direct connections to routers and other devices we found online, even when credentials where publicly exposed. We performed our analysis and code review of the looking-glass software in a local setting, performing the injection experiments in a controlled environment, as described further below.

During our research we found several vulnerabilities and misconfigurations. Unfortunately, in most of the cases these flaws were trivial to detect and to exploit, significantly increasing the relevance of our study. For this reason, we decided to responsible disclose the vulnerabilities and misconfigurations by contacting the CERT/CC in order to coordinate all the entities involved. In addition, we contacted and reported our findings to the software developers, who acknowledged the problems and are working on fixes.

5.2 Methodology

Not all ASes provide public looking-glasses, and there is not a single central list containing all of them. As such, we compiled a list of known ones on a best-effort basis by collecting URL from operators-related resources.

Such resources are not updated frequently and may contain unreachable or inactive services, while missing recent ones. We started filtering out the inactive entries by connecting to them and looking for network or web failures. For the remaining ones, we performed a simple HTTP request, matching the returned page with a set of HTML signatures we developed for each open-source looking glass we were able to download. At the end of this fingerprinting phase, we obtained the list of the most popular open-source software, and which ASes are using them.

We then proceed to perform a security review of their codebase, especially looking for the kind of issues we described in Section 4. At the same time, we also collected a list of default paths for configuration files, sources, and keys. This knowledge base was subsequently used to scan web servers and search engine indexes for publicly exposed configurations and key files.

At this point, we focused our tests on the subset of ASes for which we identified an existing security problem. In our experiment, we first tried to enumerate at least one public IP for all routers connected to a looking-glass, by requesting an ICMP echo request to one public IP address under our control. We then looked for IP addresses publicly exposing an admin service (telnet or SSH). We performed this test by checking for publicly routable loopbacks or services listening on routing interfaces. These actions were easy to automate, as

Looking glass	*Number of ASes*
Custom/Unknown	515
Unreachable	184
Cougar-LG	175
Cistron-LG	15
MRLG4PHP	12
MRLG	11
Telephone	7

Table 2: Looking-glass software deployments.

none of the open-source looking-glass software employ CAPTCHAs.

At the end of our experiments, we identified a subset of vulnerable ASes, for which an attacker could be capable of directly injecting commands on the router or could be able to recover the credentials required to remotely log in in a publicly accessible interface.

We then correlated this list of ASes with historical records of BGP announces collected by RIPE RIS through several probes and peers, all over the world. The results suggest that by compromising some of these ASes it would probably be possible for an attacker to announce routes that would not be properly filtered by neighbouring ASes.

We acknowledge that this methodology may result in several false-negatives (URLs not in the original list, unavailable at the time of the experiment, or missed while fingerprinting) and few false-positives (updated and well-configured routers with proper ACL).

For the former, one could repeat the experiments in the future to compare the results. Unfortunately, it is impossible to remove the false positives without performing a real test on the routers – action that we could not perform for ethical reasons.

6 Results

After an initial fingerprinting phase, we collected 919 unique URLs of looking-glass applications, out of which 220 were running one of open-source software listed in Table 1. The remaining ones were either unreachable (184 cases), or running a custom code we were not able to identify with our signatures (515).

6.1 Impacted ASes

An initial lookup on web search engines already proved fruitful, with at least 4 configuration files crawled by indexing bots.

Vulnerabilities	*Affected ASes*
Exposed configuration files	28
Remote command injection	12
Misconfigured CGI	4
Exposed SSH private keys	2

Table 3: Number of vulnerable ASes.

Looking for misconfigurations, we observed a large number of exposed credentials by just visiting the default configuration paths for each software, as gathered from the source code. At least 28 configuration files containing IPs and credentials can be directly downloaded by malicious attackers, and in two cases we also observed *private* SSH keys exposed on the web server.

Focusing on the source-code, we observed a general lack of basic security practices. As already said, none of above software make use of anti-automation mechanisms. The most worrisome result of our review was one case of missing input sanitation mechanism which allows injection of arbitrary commands to the router console (CVE-2014-3927 [19]). We also observed three cases of insecure default paths and permissions, mostly the cause of the exposed credentials mentioned above (CVE-2014-3928 [20], CVE-2014-3929 [21] and CVE-2014-3930 [22]). In one case, the result page was vulnerable to a XSS attack (CVE-2014-3926 [23]).

On the host side, an interesting finding was a remote memory corruption related to the parsing of ICMP-response fields in a bundled ping-like utility, meant to be run as SUID on the looking-glass web server (CVE-2014-3931 [24]).

To summarize, we detected a total of 46 vulnerable ASes, which could be targeted by attackers in order to gain access to the Internet infrastructure. A quick summary of the issues is shown in Table 3.

Figure 2 plots the number of affects ASes by country. The most vulnerable nation is Russia with six ASes, followed by Poland with four. Then there are 16 countries with a vulnerable AS. However, none of these ASes are known tier-1 provider.

Through the use of looking-glasses and network probes, we globally identified 53 routers across 20 ASes publicly exposing telnet or SSH services. This figure does not include known public route servers, typically accessed over telnet, which have been filtered out from this set.

Finally, by combining all this data, we were able to correlate two leaked configurations to ASes whose routers administration services were listening on routable IPs. In total, we observed six routers connected to looking-glass instances which are directly reachable over

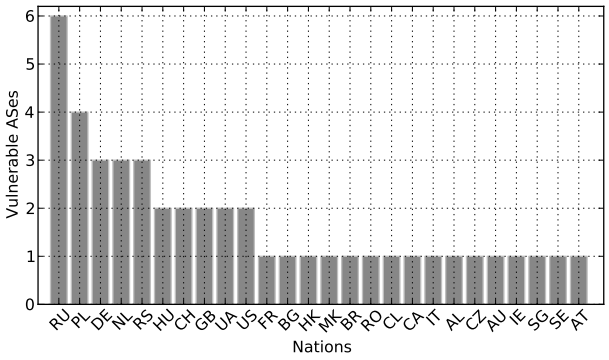

Figure 2: Geographical distribution of vulnerable ASes.

telnet and for which credentials are publicly exposed.

As the final result of our experiments, we identified at least 14 ASes which are highly interesting for attackers targeting the Internet infrastructure: 12 of them run software that is vulnerable to arbitrary commands injection, while the remaining 2 have a largely exposed infrastructure.

6.2 BGP injection feasibility

A malicious attacker capable of manipulating BGP sessions may affect the network traffic across the world. Fortunately, several mechanisms exist to limit wordlwide issues, most notably prefix filtering in which upstreams and peers filter a neighbouring AS to only announcing known routes, effectively preventing malicious injection.

However, several past hijacks [25] and academic studies [26] have shown how often BGP announces are not properly filtered, resulting in misconfigurations and hijacks being observed in the wild.

While BGP anomaly analysis is a complex subject outside of the scope of this paper, we tried to assess if it is possible for an attacker controlling any of the above vulnerable ASes to announce unfiltered routes. At this point, we are interested in knowing if some peers of those ASes were not applying proper filters, by looking into anomalies detected by the RIPE RIS.

By manually analysing historical data, we spotted at least three such cases where anomalous announces where recorded by BGP collectors over the low-visibility threshold (i.e., relied by multiple peers).

In two cases, overlong prefixes (more specific than /24) where observed by multiple participating peers, while in the remaining case an event lasting 8h occurred where one AS briefly announced an unrelated prefixes already in use (a short multi-origin AS event, possibly due to a temporary misconfiguration).

Such unfiltered prefixes hint at the possibility for a remote attacker to distribute bogus BGP routes, by compromising one of above ASes.

7 Related Work

Backbone routers are the main players of the Internet core infrastructure, and they are considered key points for cyberspace security. For these reasons, these devices have been studied from several perspectives.

Remote exploitation of routers has been studied in the past, with Lindner showing the feasibility and the technical challenges for Cisco IOS [27, 28, 29] as well as for Huawei VRP [30]. Since then, other researchers focused their attention on the remote exploitation of Cisco routers [31, 32] and the firmware diversity problem [33].

Despite the importance of these devices, serious local flaws are still being found [34], allowing an attacker with console access to escalate his privileges. The impact of these local flaws could be exacerbated by network engineers not following security best-practices, as shown in this paper.

Researchers also focused their attention on the defacto interdomain routing protocol, BGP. In this case, the threat model consists in taking control of a BGP device and announcing false routes or hijacking prefixes [35, 36, 26]. This is possible because BGP has been designed with the concept of trust, at a time in which security was not a real concern [1]. Several solutions have been proposed in the literature, ranging from the use of cryptography (e.g. PKI for a root of trust) [37, 38] to anomaly detection [39, 40]. Unfortunately, they are not widely used by network providers.

More specific to looking-glass applications, researchers have found some vulnerabilities in the past, but their focus was on the execution of code on the server running the looking-glass software [41]. Some concerns related to information leakage were also raised on public mailing lists [42, 43]. However, to the best of our knowledge, no comprehensive studies had been conducted so far on this class of applications.

8 Conclusions

We believe that our study shows how basic best-practices are not uniformly applied by operators across the world, and how an attacker can target several ASes to disrupt the Internet without much effort.

Just by looking at public information gathered on the web and applying simple heuristics, we have been able to detect a large number of attack surfaces in this critical infrastructure. In particular, we directly identified at least 45 exposed ASes, we found six routers across two ASes which could be remotely accessed by malicious attackers via exposed credentials, and at least 12 additional ASes vulnerable to arbitrary commands injection through the web interface. Moreover, we have spotted unfiltered prefixes originating from at least three of these ASes in the

past.

Acknowledgements

We would like to thank Pierre-Antoine Vervier and Quentin Jacquemart for patiently discussing and advising us on the complex topic of BGP analysis. Moreover, we would like to thank Thijs Kinkhorst and ANSSI personnel for directing us in the initial disclosure steps.

References

[1] S. Murphy. BGP Security Vulnerabilities Analysis. RFC 4272 (Informational), January 2006.

[2] Y. Rekhter, T. Li, and S. Hares. A Border Gateway Protocol 4 (BGP-4). RFC 4271 (Draft Standard), January 2006. Updated by RFCs 6286, 6608, 6793.

[3] Quagga Routing Suite. `http://www.nongnu.org/quagga/`.

[4] Xorp - eXtensible Open Router Platform. `http://www.xorp.org/`.

[5] University of Oregon Route Views Project. `http://www.routeviews.org/`.

[6] Cougar LG. `https://github.com/Cougar/lg`.

[7] Cistron LG. `http://www.tux.org/pub/people/miquel-van-smoorenburg/net/`.

[8] MRLG. `http://mrlg.op-sec.us/`.

[9] MRLG4PHP. `http://freecode.com/projects/mrlg4php`.

[10] Telephone LG. `https://github.com/telephone/LookingGlass`.

[11] Hristo Bojinov, Elie Bursztein, and Dan Boneh. XCS: cross channel scripting and its impact on web applications. In *ACM Conference on Computer and Communications Security*, pages 420–431, 2009.

[12] Johnny Long. Google Hacking for Penetration Testers. *Black Hat USA*, 2005.

[13] Cisco on Cisco Best Practices – IP Addressing Policies. `https://www.cisco.com/web/about/ciscoitatwork/downloads/ciscoitatwork/pdf/Cisco_IT_IP_Addressing_Best_Practices.pdf`, 2010.

[14] Ahn, Luis Von and Blum, Manuel and Hopper, Nicholas J. and Langford, John. CAPTCHA: Using Hard AI Problems for Security. In *Proceedings of the 22Nd International Conference on Theory and Applications of Cryptographic Techniques*, EUROCRYPT'03, 2003.

[15] Cisco IOS Password Encryption Facts. `http://www.cisco.com/c/en/us/support/docs/security-vpn/remote-authentication-dial-user-service-radius/107614-64.html`.

[16] Cisco PSIRT. Cisco IOS and Cisco IOS XE Type 4 Passwords Issue . `http://tools.cisco.com/security/center/content/CiscoSecurityResponse/cisco-sr-20130318-type4`.

[17] Cisco PSIRT. AAA Command Authorization by-pass . `http://tools.cisco.com/security/center/content/CiscoSecurityResponse/cisco-sr-20060125-aaatcl`.

[18] Juniper Networks SIRT. Unauthorized user can obtain root access using cli. `http://kb.juniper.net/InfoCenter/index?page=content&id=JSA10420&actp=search&viewlocale=en_US&searchid=1400663655904`, 2010.

[19] mrlg4php: remote command injection to router's console via "argument" parameter. `http://www.s3.eurecom.fr/cve/CVE-2014-3927.txt`, 2014.

[20] Cougar-LG: Unsafe configuration file path/ACL. `http://www.s3.eurecom.fr/cve/CVE-2014-3928.txt`, 2014.

[21] Cougar-LG: Unsafe SSH keypairs path in default config. `http://www.s3.eurecom.fr/cve/CVE-2014-3929.txt`, 2014.

[22] Cistron-LG: Unsafe configuration file path/ACL. `http://www.s3.eurecom.fr/cve/CVE-2014-3930.txt`, 2014.

[23] Cougar-LG: XSS in title via "addr" parameter. `http://www.s3.eurecom.fr/cve/CVE-2014-3926.txt`, 2014.

[24] MRLG: remote memory corruption in fastping (SUID binary). `http://www.s3.eurecom.fr/cve/CVE-2014-3931.txt`, 2014.

[25] Earl Zmijewski. Indonesia Hijacks the World. `http://www.renesys.com/2014/04/indonesia-hijacks-world/`.

[26] Ballani, Hitesh and Francis, Paul and Zhang, Xinyang. A Study of Prefix Hijacking and Interception in the Internet. In *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '07, New York, NY, USA, 2007.

[27] FX, FtR and kim0. Attacking networked embedded systems. *Black Hat USA*, 2002.

[28] Felix "FX" Lindner. Cisco Vulnerabilities - Yesterday, Today and Tomorrow. *Black Hat USA*, 2003.

[29] Felix "FX" Lindner. Cisco IOS Router Exploitation. *Black Hat USA*, 2009.

[30] Felix "FX" Lindner. Hacking Huawei Routers. *DEFCON XX*, 2012.

[31] Michael Lynn. Cisco IOS Shellcode. *Black Hat USA*, 2005.

[32] Andy Davis. Remote Cisco IOS FTP exploit, 2007.

[33] Ang Cui and Jatin Kataria and Salvatore J. Stolfo. Killing the Myth of Cisco IOS Diversity: Recent Advances in Reliable Shellcode Design. In *WOOT*, pages 19–27, 2011.

[34] Juniper Networks SIRT. Multiple privilege escalation vulnerabilities in Junos CLI (CVE-2014-0615). `http://kb.juniper.net/InfoCenter/index?page=content&id=JSA10608&actp=search&viewlocale=en_US&searchid=1400663655904`, 2014.

[35] Butler, Kevin R. B. and Farley, Toni R. and McDaniel, Patrick and Rexford, Jennifer. A Survey of BGP Security Issues and Solutions. *Proceedings of the IEEE*, 98:100–122, 2010.

[36] Ramachandran, Anirudh and Feamster, Nick. Understanding the Network-level Behavior of Spammers. In *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, SIGCOMM '06, 2006.

[37] *Public-key infrastructure for the Secure Border Gateway Protocol (S-BGP)*, 2001.

[38] RFC 2385 - Protection of BGP Sessions via the TCP MD5 Signature Option. `http://www.ietf.org/rfc/rfc2385.txt`, 1998.

[39] Caesar, M. and Rexford, J. BGP Routing Policies in ISP Networks. *Netwrk. Mag. of Global Internetwkg.*, 2005.

[40] Nordström, Ola and Dovrolis, Constantinos. Beware of BGP Attacks. *SIGCOMM Comput. Commun. Rev.*

[41] rgod. Looking Glass v20040427 arbitrary commands execution / cross site scripting. `http://retrogod.altervista.org/lookingglass.html`, 2005.

[42] BGP vulnerability? http://www.gossamer-threads.com/lists/cisco/nsp/11323?do=post_view_threaded#11323, 2004.

[43] TCP BGP vulnerability looking glass and route server issues. http://www.nanog.org/mailinglist/mailarchives/old_archive/2004-04/msg00684.html, 2004.