The BGP Hackathon 2016 Report

Alberto Dainotti
CAIDA, UC San Diego
alberto@caida.org

Ethan Katz-Bassett
University of Southern California
ethan.kb@usc.edu

Xenofontas Dimitropolous
University of Crete / FORTH
fontas@ics.forth.gr

This article is an editorial note submitted to CCR. It has NOT been peer reviewed.
The authors take full responsibility for this article's technical content. Comments can be posted through CCR Online.

ABSTRACT

Internet routes – controlled by the Border Gateway Protocol (BGP) – carry our communication and our commerce, yet many aspects of routing are opaque to even network operators, and BGP is known to contribute to performance, reliability, and security problems. The research and operations communities have developed a set of tools and data sources for understanding and experimenting with BGP, and on February 2016 we organized the first BGP Hackathon, themed around live measurement and monitoring of Internet routing. The Hackathon included students, researchers, operators, providers, policymakers, and funding agencies, working together on projects to measure, visualize, and improve routing or the tools we use to study routing. This report describes the tools used at the Hackathon and presents an overview of the projects. The Hackathon was a success, and we look forward to future iterations.

Categories and Subject Descriptors

C.2.3 [**Network operations**]: Network monitoring; C.2.5 [**Local and Wide-Area Networks**]: Internet

Keywords

routing, BGP, Internet measurement, hackathon

1. INTRODUCTION AND MOTIVATION

On 6-7 February 2016, around 90 researchers and students from academia, industry and other institutions around the world, participated in the first BGP (Border Gateway Protocol) hackathon. The event, themed on *live BGP measurements and monitoring*, was hosted by the Center for Applied Internet Data Analysis (CAIDA) at UC San Diego's San Diego Supercomputer Center (SDSC) and was organized in cooperation with Colorado State University (CSU), University of Southern California (USC), Universidade Federal De Minais Gerais (UFMG), Foundation for Research and Technology Hellas (FORTH), RouteViews, and RIPE NCC.

The hackathon was held the weekend immediately preceding the 66th North American Network Operators' Group (NANOG) conference [24] and the 8th AIMS academic Internet measurement workshop [12], both also held in San Diego, and it aimed to:

- bring together these different communities, e.g., to discuss problems operators face that academics may want to research;
- advertise (new) tools to the communities, train people to use them, and encourage further use;
- bring people together to work on interesting/important problems, spurring collaborations;
- provide additional incentive for students to attend NANOG and for members of the industry to join the hackathon.

This was the first hackathon centered around BGP, and it involved a mix of programming, live experimentation, and collaborative research, which made our job as organizers particularly compelling. The hackathon also represented a timely and unique opportunity. The Internet would hardly exist without BGP, which has been the subject of experiments and research for decades. Still, BGP suffers from issues in performance [35], security [38, 27], and availability [25], challenging researchers and engineers, who must deal with the scarcity or incompleteness of empirical data (the AS-level graph[23], AS relationships[22], deployed routing policies [3] etc.). We are in need of improvements to measurement standards, capabilities and coverage of data collection platforms, experimental measurement and emulation testbeds and research tools [18].

In this context, there have been interesting recent developments: *(i)* providers of public data such as RouteViews and RIPE RIS started working on live data feeds; *(ii)* Colorado State's BGPMon has transitioned to community-supported software such as goBGPd, OpenBMP and Cassandra in order to offer scalable, robust, web-based real-time access to BGP data as well as access to over 10 years of archived data; *(iii)* the PEERING testbed provided researchers for the first time with an open community testbed that lets them exchange routes and traffic with hundreds of real networks around the world; *(iv)* the first version of BGPStream, CAIDA's open source framework for BGP data analysis, was released in late 2015 (Section 3).

The BGP hackathon served two primary roles. First, it facilitated convergence of these software and infrastructure development efforts and offered a snapshot of the evolving state of the art to a critical mass of young researchers and seasoned scientists and engineers. Second, the hackathon provided a venue to bring together people who are interested in Internet routing, but come from different communities and serve different roles – network engineers, developers, data providers, academic and industry researchers, operators – to meet, discuss, and find opportunities for collaboration and progress. This report describes the involvement of the community in the event (organizers, sponsors, participants, etc.), the platforms used in the hackathon, with emphasis on novel aspects, the challenges the hacking teams worked on, and the resulting collaborations; concluding with lessons learned and ideas on organizing a similar event in the future.

2. COMMUNITY INVOLVEMENT

Planning the hackathon was an interesting community experiment itself. The organizing committee, comprising of 7 institutions [6], met in a CAIDA-hosted workshop on 12-13 November 2015 [4]. We defined the format of the event and in particular discussed *(i)* which tools and data sources to make available in the hackathon and how they could complement each other (Section 3) and *(ii)* a

set of challenging projects that participants could pick, transform or use for inspiration to propose entirely new projects. Briefly afterwards we announced the event.

The response from the community was enthusiastic. We received more than 80 applications as competing participant, from which we selected 50 ($\approx$ 30 graduate students) and we assigned 33 travel grants (almost entirely to students). Selection criteria for participation took into account the ideas for potential projects and feedback applicants wrote in the application form, providing equal opportunities to students from different countries and institutions, and gender balance (we accepted all 6 female applicants – students and postdocs). In addition, several experts applied to offer their knowledge to hackathon participants. The 90 attendees, including jury members, represented academia (50), industry (21, including Facebook, Google, NTT, Cloudflare, DE-CIX, Symantec, Cisco, Comcast), US and European government agencies (6, from DHS, NSF, FNISA), and non-profit organizations (12, including ISOC, LACNIC, M-Lab) [7].

Before the event, the participants interacted through a mailing list and used the hackathon wiki to comment on and propose project ideas as well as indicate their specific expertise. During the hackathon, teams of 1 to 7 people formed spontaneously and worked on developing code to extend, use and/or integrate existing open source platforms and tools by demonstrating their utility in understanding or solving practical problems, such as detecting BGP prefix hijacking attacks, studying performance of anycast, etc. (Section 4). Competing participants worked side by side with top experts in the field – some of them the developers/maintainers of the platforms used in the experiments – who sometimes hacked code and experiments themselves. During both days, the jury members sat at the tables with the "hackers" and engaged with them in conversations about the challenges they encountered and the experiments they conducted. The event was extremely social, with people from different teams often exchanging help and discussing ideas.

This event was possible thanks to the generosity of several sponsors (Section 6), who not only helped financially and by providing facilities and infrastructure but also by participating as experts and jury members. In particular, we were granted an educational resource allocation on SDSC's Comet supercomputer, which enabled hackathon participants access to massive computational resources (Comet has a total of 1944 nodes, each with 24 CPU cores and 128GB of RAM). To simplify use of the machine and improve performance, we pre-installed BGPStream and hosted a local replica of the entire RouteViews and RIPE RIS data archives that users were automatically directed to by the BGPStream broker service. In addition, we made use of Comet's virtualization capability in order to provide a specialized emulation PEERING environment over which participants had complete control.

3. RESEARCH AND DATA PLATFORMS

This section briefly introduces the platforms we made available to participants.

PEERING [34] is a system that enables safe, secure, and tightly controlled access for researchers and educators to the Internet routing system. Traditionally, the barriers to conduct Internet routing experiments hindered progress. To experiment with novel routing ideas or to understand aspects of the current routing ecosystem, researchers need the ability to actively participate in this ecosystem by emulating an autonomous system (AS). PEERING operates an autonomous system and has points of presence (PoPs) at multiple Internet Exchange Points and universities. The testbed can multiplex multiple simultaneous research experiments, each of which independently makes announcements and routing decisions as well as exchange traffic with the real Internet.

Spurred by the needs of the Hackathon, the PEERING team expedited a number of major improvements to the testbed. First, to replace manual, ad-hoc configuration and tracking of PEERING sites and experiments, the team built a centralized site that manages experiments and resources and automatically generates and deploys configuration files to PEERING sites and experiments, making it easier to provision and update experiments. Second, the team redesigned the toolkit that experimenters use to interact with the testbed, making it easier to use while adding more sophisticated functionality, such as control over which peers receive an experiment's announcements at an IXP.

BGPStream is an open-source software framework for live and historical BGP data analysis, supporting scientific research, operational monitoring, and post-event analysis. BGPStream allows users to either quickly inspect raw BGP data from the command-line, develop Python apps, or build complex systems using a C/C++ API. BGPStream provides seamless and live access to both the RouteViews and RIPE RIS data archives, and for the BGP Hackathon we added experimental live access to a stream of BGP data generated by BMP-enabled RouteViews collectors. Organizing the hackathon also pushed us to introduce additional features and fixes in time for the event, which were released as version 1.1 of the software framework.

The **BGPMon** system [2] monitors, stores and analyzes BGP information captured from approximately 400 routers around the world in real-time. The information collected by BGPMon is useful to researchers and network operators. BGPmon is open source and built with community-supported software such as GoBGP, OpenBMP and Cassandra. BGPMon can be deployed as a public service, as done at CSU, or as a private service for sensitive networks. BGPMon interface supports fetching BGP messages in various formats. Requests can be made based on a time range, a specific peer, AS, a prefix etc.. The interface runs over HTTP in a RESTful manner so users can easily integrate BGPMon into their applications. Several modules can be built on top of BGPMon. For example, in time for the hackathon, CSU developed a prefix hijack notification module that allows users to setup alerts for their prefixes.

Periscope [11] is an overlay measurement platform which provides a standardized interface to Looking Glass (LG) servers deployed by individual AS and IXP operators. Periscope offers a RESTful API that can be used to retrieve the available LG vantage points, issue measurements for three types of LG commands (show ip bgp, traceroute, ping), query the status of measurements and retrieve the output in three different formats (json, iplane or the raw format returned by the LG). Periscope imposes two limits (a user-specific and a global) on the frequency and number of requests issued to each LG in order to ensure that the querying rates will conform to the intended LG usage and prevent misuse that can overwhelm the LGs with excessive number of requests. Periscope was made available to participants even before the scientific paper introducing it was presented at the Passive and Active Measurement conference in April 2016 [21].

RouteViews [28] operates BGP routing collectors around the world. Operators use the command lines of the collectors to verify BGP routing. BGP UPDATES and RIBS are stored and saved in the RouteViews data archives. Researchers use the archive data to analyze the routing behaviors of the Internet. Some of the RouteViews peers are "live peers" and provide real-time data streams. The CAIDA BGPStream toolkit can be used to work with both the archive data, and the live data peers. RouteViews data has characteristics of which researchers should be aware. Firstly, most peers on RouteViews are "full table" peers. This was a design

choice, so that operators could see the full table view from the perspective of each peer for debugging purposes, as opposed to just owned/advertised prefixes. Secondly, some RouteViews collectors are "multihop", while some are located at exchange points. The multihop collectors have peers that are located all over the world. The exchange collectors have peers that are located at the specific exchange.

OpenBMP servers were established in a few locations to support the integration of live BGP data with BGPStream in time for the hackathon. Live sessions were forwarded from the primary RouteViews hardware router into OpenBMP, and then into BGPStream. Live peering sessions were provided by a number of large, Tier-1 Internet Service Providers: TATA, TISCALI, NTT, ATT, Level3, and Hurricane Electric.

RIPE RIS [30], the RIPE NCC's BGP collector service is similar to RouteViews. BGP RIB dumps and update messages are available as downloadable MRT-format dump files. For the hackathon, two experimental real-time streaming interfaces were made available, one through websockets, which allows users to specify custom filtering for peer, ASN, prefix, and a JSON over a Kafka [1] interface.

Ark [5] is CAIDA's globally-distributed active measurement platform consisting of 1U rack-mounted servers and Raspberry Pi's. Facilities useful to BGP hackathon attendees include Vela, a web-based interface to performing on-demand ping/traceroute measurements from Ark monitors, tod-client, a command-line interface providing similar functionality as Vela, and archival data of the ongoing large-scale traceroute measurements of every routed /24.

RIPE Atlas [31] is an active Internet measurement network with over 9000 vantage points. Data provided by RIPE Atlas include description about existing measurements and vantage points, as well as measurement results in downloadable and real-time streaming formats. Members of the RIPE Atlas team supported the participants with information, use cases, and most importantly, credits to execute measurements on the fly, for example to correlate BGP events with data plane changes or compare control/data plane.

RIPEstat [32] is a web-based interface that provides "everything you ever wanted to know" about the IP address space, Autonomous System Numbers (ASNs), and related information for hostnames and countries in one place. It presents registration and routing data, DNS data, geographical information, abuse contacts and more from the RIPE NCC's internal data sets as well as from external sources, such as other Regional Internet Registries and IANA. RIPEstat's main web-based interface presents this information in the form of widgets that can be embedded on any webpage. It also provides an API to access the raw data for use in advanced applications.

4. HACKED PROJECTS

Each team of the hackathon assigned themselves a unique codename (e.g., *Hijacks-2*), which can be used to easily access on the hackathon wiki and GitHub repository the description of the project [9] and the corresponding sourcecode and presentation material [8]. We first introduce the four co-winning projects and then discuss the work of the other teams, follow-ups and other collaborations spurred. In selecting the winners, the jury took into consideration – among several criteria and after a careful discussion – the utility of the code developed (BGPStream-1 and VIZ-2 projects) and the novelty of the experiments (Anycast-1 and Hijacks-2).

4.1 The four winning projects

The **VIZ-2** team developed a version of BGPlay [13], accompanied by other tools, that is easily deployable in a private setup. BGPlay is a visual interface developed in Javascript that allows a user

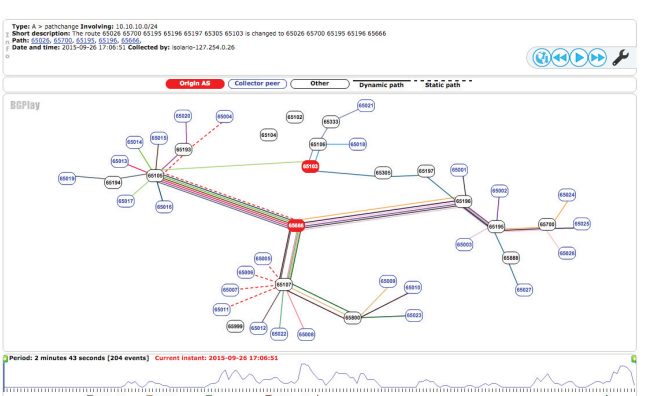

Figure 1: BGPlay is a Javascript-based visual interface that allows a user to track changes in AS paths (observed through BGP data feeds) towards an AS. The VIZ-2 team developed a version of BGPlay, accompanied by other tools, that is easily deployable in a private setup, which includes a data collector and supports real-time streaming and visualization.

to track changes in AS paths (observed through BGP data feeds) towards an AS. Figure 1 shows an example of visualization. The system developed during the hackathon includes a data collector and supports real-time streaming and visualization of the data collected. A user simply needs to configure the data collector provided and it can then observe path changes in real time. This project continued after the hackathon and the authors plan to make available an image of a virtual machine ready to be set up in a private network, which can be used (also) with private data feeds. Based on feedback received at the hackathon and at the latest RIPE meeting [33], Massimo Candela from RIPE decided to continue this project, also with help from Alistair King at CAIDA.

The **Hijacks-2** team developed a system for live detection of BGP hijacking activity. They used the experimental JSON over Kafka RIPE RIS and OpenBMP/BGPStream interfaces – with latencies of few seconds up to 2 minutes – to detect MOAS (Multiple Origin ASes) events and what they defined as "subMOAS", i.e., announcements of prefixes that overlap and are originated from different ASes. To remove benign cases, they developed 6 filters based on RPKI, Route Objects in Internet Routing Registries, private AS numbers, siblings, business relationships, and AS customer cones from CAIDA's AS-Rank [26]. Through these filters, they ruled out 60% of the observed events, and visualized the remaining 40% by showing changes detected in the AS paths. A few of the team members were already involved in a collaborative NSF-funded project of Stony Brook University and CAIDA to detect hijacking attacks [17] and they will continue working on the topic.

Shane Alcock from University of Waikato formed a single-member team (**BGPStream-1**). He significantly improved filtering in BGPStream by developing a BPF-style [20] language for specifying filters that can be used both from command line and with the BGPStream Python/C APIs. In particular, he implemented filters capable of understanding regular expressions [36] applied to AS paths. This project was well received by other team participants who stated they would have used such features during the hackathon if already available. CAIDA started collaborating with Shane to incorporate his code in the forthcoming release of BGPStream.

Finally, the **Anycast-1** team emulated a service that uses anycast routing. They set up 7 anycast nodes using 7 PEERING sites,

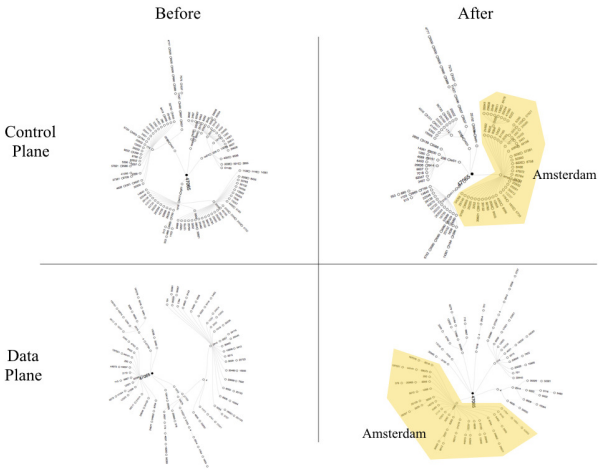

Figure 2: Using the PEERING testbed, the Anycast-1 team emulated a service that uses anycast routing and observed the effects of announcements and withdrawals – e.g., shutting down the most popular anycast instance – on traceroutes (RIPE Atlas) and on the control-plane (using BGPStream). They also showed the impact of these changes through a visualization that they developed using the D3 Javascript framework.

from which they announced a chosen prefix, and repeatedly sent traceroute towards this prefix from RIPE Atlas probes in different geographic locations. They observed the effects of announcements and withdrawals on traceroutes (RIPE Atlas) and on the control-plane (using BGPStream). In particular, they investigated the effect of shutting down the most popular site – the one in Amsterdam – and visualized the impact of these changes through a visualization that they developed using the D3 Javascript framework (Figure 2). University of Twente is currently extending this project, making the code more dynamic and integrating it in a monitoring framework for DNS anycast, in which SIDN (the *.nl* ccTLD registrar) is interested.

4.2 Other projects and collaborations

Hijacking was a popular topic among participants. Another team (**Hijacks-1**) worked on this topic, evaluating automated prefix deaggregation as a defense mechanism to BGP hijacking attacks detected in real-time. The team assumed that an AS uses the streaming BGP data interfaces available from BGPstream, RIPE RIS, and/or BGPmon to detect hijacks of their own prefixes. Upon detection of a hijack, the AS de-aggregates the attacked prefix, e.g. a /23 into two /24 prefixes, to rapidly mitigate the effects of the hijack. The team used PEERING to emulate hijacks of a prefix owned by PEERING in the real Internet and measure how fast BGP hijacks can be detected and mitigated. Researchers at the University of Crete and FORTH further developed the project on detecting BGP hijacking, which resulted in a demo that will be presented the ACM SIGCOMM 2016 conference [14] and a collaboration with CAIDA and Stony Brook to join efforts with their NSF-funded project on detecting BGP hijacking and man-in-the-middle attacks [17].

The **RPKI-42** team investigated whether ASes consider route validity (based on route origin authorization (ROA) certificates in ARIN's RPKI) as part of their routing policies. Researchers from USC, IIJ, UFMG, and Freie Universität Berlin worked on the project at the hackathon, and they are continuing it now in collaboration with Matthias Wählisch (Freie Universität Berlin), Thomas C. Schmidt (HAW Hamburg), and Doug Montgomery (NIST). The goal is to conduct a longitudinal study of deployment of RPKI based origin validation on the Internet. This project has received feedback and support from Josh Bailey and Chris Marrow at Google, who Anees Shaikh from Google (hackathon attendee) put Ethan Katz-Bassett in touch with.

The **DataPlane-1** group, wondered if a researcher can leverage control-plane measurements to use their data-plane measurement budget more effectively. In other words, in order to reduce a set of periodic data-plane measurements, can we use an update of a BGP path toward an AS as a trigger to mark previous traceroutes towards such AS as "stale"? They tested this hypothesis with data from *(i)* periodic (15 min) traceroutes from RIPE Atlas nodes to Atlas anchors and *(ii)* RouteViews and RIPE RIS monitors. The team picked control-plane monitors in the same ASes of Atlas nodes and examined paths towards the ASes hosting RIPE Atlas anchors. They considered source/destination AS pairs compatible with their study when the BGP AS path and the traceroute-derived AS path differed of less than 40% (which was true for 20% of the pairs). In a demo, they showed that for one third of the pairs they could examine, a change in the BGP path observed on the control-plane caused a significant change in traceroute results.

The **ASRank-1** team updated CAIDA's AS-Rank system [26] to automatically crunch RouteViews, RIS, and Ark data on a daily basis (instead of the current manual monthly data processing) using BGPStream and to store the results into a SQL database which they designed. They also populated the database with historical data from before 1998 and developed web visualizations showing the expansion of the customer cone of given ASes over the years.

A few projects would have highly benefited from an additional day of baking. The **Stability-1** project investigated the correlation between BGP control-plane instability and loss of data-plane connectivity. For prefixes that showed a high rate of BGP updates in 5 minutes bins, they inspected the packet loss experienced by RIPE Atlas pings targeting them. The **LinkRank-1** team proposed different metrics to rank links in the AS graph over time (counting, for each directed link, the number of paths, prefixes and addresses covered, as well as link stability). One possible application is to use linkrank metrics over time as a baseline to detect route hijacks and leaks. Members of the **BGPD-3** team developed a set of perl scripts to replay BGP updates from MRT files or BGPReader output into BGP daemons. The biggest problem they encountered was to have a daemon accept multiple paths towards the same prefix instead of overwriting them with the last one read. Thanks to suggestions from jury member Alvaro Retana (Cisco) they overcame this problem by using the BGP Add-Path feature, which is currently supported in the BIRD Internet Routing Daemon Project [16].

The **VIZ-3** team created an interactive version of CAIDA AS Core map [10] as a javascript-based visualization with zooming functionalities that allow a user to inspect each single node of the AS topology – an impressively eye-candy visualization that CAIDA plans to host on their web site and make available soon.

The BGPMon group formed a (non-competing) **GoBGPMon** team with the NTT group to increase synergy between NTT's BGP deamon implementation (GoBGP) in Go language and the new BGPMon platform. The groups worked together on defining the RPC Monitoring API for GoBGP, which allows BGPMon to obtain incoming messages to any RIB of the router. They also worked on changing internal data paths to the router as to mitigate the possibility of messages wrongly being advertised to a GoBGP/BGPMon peer. Following the hackathon BGPMon folks have continued working with the GoBGP team to deploy more GoBGP daemons as col-

lectors internally at CSU and at other ISPs. They are working on reducing the memory consumption of the BGP router and improving the wire data format for the actual BGP message exchange.

Folks from Jive and Google, together with Mattijs Jonker, PhD student at University of Twente, created an **OpenConfig-1** team. Every BGP speaker (hardware or software) requires a significant amount of proprietary integration to speak the dialect of its management language. OpenConfig [29] is an operator working group that is defining vendor-neutral APIs for configuration and operational state for all parts of network infrastructure, including BGP and routing policy. These models are being supported natively on commercial routers. The goal of the OpenConfig-1 team was to extend OpenConfig support to open source software BGP implementations, focusing on Quagga and its emerging configuration API support. During the hackathon they successfully implemented a prototype to extract data from Quagga BGPd and present it in OpenConfig structure and format. This work was extended further at the IETF hackathon in April in Buenos Aires [19] to form the basis of a model-driven management interface for Quagga.

Finally, there had been several opportunities for discussion and collaboration unrelated to the specific hackathon projects. For example, Roya Ensafi from Princeton got a chance to talk to iGreedy developers [15] from TELECOM ParisTech, who helped her to set up their tool for her research. At the hackathon, Roya also discussed about router's geolocation and issues of current geolocation databases with Christos Papadopoulos and his group at CSU, and they started a collaboration that reportedly made good progress since the hackathon. John Kemp from RouteViews spent time discussing BMP as a data collection tool with Tim Evens of OpenBMP.org. In particular, they discussed how to move forward with the development of BMP integration in Quagga or BIRD. This would produce a standardized method for live BGP data collection. CAIDA also started a collaboration with Cisco and OpenBMP.org to add to BGPStream native support for OpenBMP. Finally, the network security group at TU Wien, is integrating practical BGP analysis exercises based on BGPStream and RouteViews/RIPE RIS data sources into their network security labs used in undergraduate and graduate classes [37]. Conversations between a number of participants continued at NANOG and AIMS in the days immediately following the hackathon.

5. FINAL THOUGHTS

The Internet is central to all our lives, and BGP routing is central to the Internet. Despite its importance, BGP is known to suffer from performance, reliability, and security problems. This first BGP Hackathon achieved two concrete goals towards enabling research into and ultimately improvement of Internet routing. First, it introduced attendees to a tool suite now available to researchers and practioners in the area, including real-time feeds of data, query interfaces for processing and streaming the data, measurement platforms, Python and C APIs for BGP data analysis, routing configuration languages, and testbeds to enable experiments. Second, it brought together people from diverse communities and with diverse backgrounds who shared a common interest in routing. The rich tool suite and the range of skills and backgrounds allowed attendees, in a short time, to develop projects that exposed and explored various interesting aspects of Internet routing. We believe that providing these tools and bringing together these communities are important steps towards a better Internet, and we hope the BGP Hackathon serves both as a springboard for ongoing collaborations and as a trial run for a recurring event in the future.

Materials related to the event are at `https://www.caida.org/workshops/bgp-hackathon/1602`

6. ACKNOWLEDGEMENTS

The BGP Hackathon was supported by ACM SIGCOMM, Cisco Systems, the Comcast Innovation Fund, Google NetOps Group and Open Source Research Group, the Internet Society (ISOC), the US Department of Homeland Security Science and Technology Directorate, and the US National Science Foundation (through NSF grant CNS-1423659).

This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation award number ACI-1053575. Specifically, it used the Comet system, which is supported by NSF award number ACI-1341698 "Gateways to Discovery: Cyberinfrastructure for the Long Tail of Science", at the San Diego Supercomputer Center (SDSC).

This material is based on research sponsored by the Department of Homeland Security (DHS) Science and Technology Directorate, Homeland Security Advanced Research Projects Agency, Cyber Security Division (DHS S&T/HSARPA/CSD), BAA 11-01-RIKA and Air Force Research Laboratory, Information Directorate under agreement number FA8750-12-2-0326. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of Department of Homeland Security, Air Force Research Laboratory or the U.S. Government.

Hackathon organizers: CAIDA, UC San Diego: Alberto Dainotti, Alistair King; USC: Ethan Katz-Bassett, Brandon Schlinker; DCC/UFMG: Italo Cunha; FORTH: Xenofontas Dimitropolous; RouteViews: John Kemp; CSU/BGPMon: Christos Papadopoulos, Spiros Thanasoulas; RIPE-NCC: Robert Kisteleki, Romeo Zwart, Vesna Manojlovic.

7. REFERENCES

[1] Apache Kafka. `http://kafka.apache.org/`.

[2] COLORADO STATE UNIVERSITY. BGPmon. `http://www.bgpmon.io/`, 2015.

[3] ANWAR, R., NIAZ, H., CHOFFNES, D., CUNHA, I., GILL, P., AND KATZ-BASSETT, E. Investigating interdomain routing policies in the wild. In *Proceedings of the 2015 ACM Conference on Internet Measurement Conference* (New York, NY, USA, 2015), IMC '15, ACM, pp. 71–77.

[4] CAIDA. BGP Hackathon Planning Meeting, 2015. `https://www.caida.org/workshops/bgp-hackathon/1511/`.

[5] CAIDA. Archipelago (Ark) Measurement Infrastructure, 2016. `http://www.caida.org/projects/ark/`.

[6] CAIDA. BGP Hackathon 2016, 2016. `https://www.caida.org/workshops/bgp-hackathon/1602/`.

[7] CAIDA. BGP Hackathon 2016 – Attendees, 2016. `https://github.com/CAIDA/bgp-hackathon/wiki/Attendees`.

[8] CAIDA. BGP Hackathon 2016 – GitHub Repository, 2016. `https://github.com/CAIDA/bgp-hackathon`.

[9] CAIDA. BGP Hackathon 2016 – List of Challenges, 2016. `https://github.com/CAIDA/bgp-hackathon/wiki/List-of-Challenges`.

[10] CAIDA. IPv4 and IPv6 AS Core: Visualizing IPv4 and IPv6 Internet Topology at a Macroscopic Scale, 2016. `https://www.caida.org/research/topology/as_core_network/`.

[11] CAIDA. Periscope Looking Glass API. http://www.caida.org/tools/utilities/looking-glass-api/, 2016.

[12] CAIDA. The 8th Workshop on Active Internet Measurements (AIMS-8), 2016. https://www.caida.org/workshops/aims/1602/.

[13] CANDELA, M. AND DI BATTISTA, G. AND SQUARCELLA, C. BGPlay.js. http://bgplayjs.com, 2016.

[14] CHAVIARAS, G., GIGIS, P., SERMPEZIS, P., AND DIMITROPOULOS, X. ARTEMIS: Real-Time Detection and Automatic Mitigation for BGP Prefix Hijacking. In *ACM SIGCOMM Demo* (2016).

[15] CICALESE, D., AUGÉ, J., JOUMBLATT, D., FRIEDMAN, T., AND ROSSI, D. Characterizing IPv4 Anycast Adoption and Deployment. In *ACM SIGCOMM CoNEXT* (Dec 2015).

[16] CZ.NIC. The BIRD Internet Routing Daemon Project, 2016. http://bird.network.cz.

[17] DAINOTTI, A. HIJACKS: Detecting and Characterizing Internet Traffic Interception based on BGP Hijacking. http://www.caida.org/funding/hijacks/, 2014. Funding source: NSF CNS-1423659.

[18] DAINOTTI, A. Measuring and Monitoring BGP, 2015. Keynote at IETF 94 http://www.caida.org/publications/presentations/2015/measuring_monitoring_bgp_ietf/.

[19] FORCE, I. E. T. IETF 95, 2016. https://www.ietf.org/meeting/95/.

[20] FREEBSD. bpf(4) Berkeley Packet Filter, 2010.

[21] GIOTSAS, V., DHAMDHERE, A., AND CLAFFY, K. Periscope: Unifying Looking Glass Querying. In *Passive and Active Network Measurement Workshop (PAM)* (Mar 2016).

[22] GIOTSAS, V., LUCKIE, M., HUFFAKER, B., AND CLAFFY, K. Inferring complex as relationships. In *Proceedings of the 2014 Conference on Internet Measurement Conference* (New York, NY, USA, 2014), IMC '14, ACM, pp. 23–30.

[23] GREGORI, E., IMPROTA, A., LENZINI, L., ROSSI, L., AND SANI, L. On the incompleteness of the as-level graph: A novel methodology for bgp route collector placement. In *Proceedings of the 2012 ACM Conference on Internet Measurement Conference* (New York, NY, USA, 2012), IMC '12, ACM, pp. 253–264.

[24] GROUP, N. A. N. O. 66th NANOG meeting, 2016. https://nanog.org/meetings/nanog66/home.

[25] LABOVITZ, C., AHUJA, A., BOSE, A., AND JAHANIAN, F. Delayed internet routing convergence. *IEEE/ACM Trans. Netw. 9*, 3 (June 2001), 293–306.

[26] LUCKIE, M., HUFFAKER, B., DHAMDHERE, A., GIOTSAS, V., AND K CLAFFY. AS relationships, customer cones, and validation. In *IMC* (Oct. 2013).

[27] LYCHEV, R., GOLDBERG, S., AND SCHAPIRA, M. Bgp security in partial deployment: Is the juice worth the squeeze? In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM* (New York, NY, USA, 2013), SIGCOMM '13, ACM, pp. 171–182.

[28] OF OREGON, U. Route Views Project. http://www.routeviews.org/, 2015.

[29] OPENCONFIG.NET. OpenConfig, 2016. http://www.openconfig.net.

[30] RIPE NCC. Routing Information Service (RIS), 2008. http://www.ripe.net/ris/.

[31] RIPE NCC. RIPE Atlas: A Global Internet Measurement Network. *The Internet Protocol Journal 18*, 3 (September 2015).

[32] RIPE NCC. RIPEstat. https://stat.ripe.net/, 2015.

[33] RIPE NCC. 72th RIPE Meeting. https://ripe72.ripe.net, 2016.

[34] SCHLINKER, B., ZARIFIS, K., CUNHA, I., FEAMSTER, N., AND KATZ-BASSETT, E. Peering: An as for us. In *Proceedings of the 13th ACM Workshop on Hot Topics in Networks* (New York, NY, USA, 2014), HotNets-XIII, ACM, pp. 18:1–18:7.

[35] SPRING, N., MAHAJAN, R., AND ANDERSON, T. The causes of path inflation. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications* (New York, NY, USA, 2003), SIGCOMM '03, ACM, pp. 113–124.

[36] THOMPSON, K. Programming techniques: Regular expression search algorithm. *Commun. ACM 11*, 6 (June 1968), 419–422.

[37] WIEN, T. Network Security Laboratory, 2016. https://www.nt.tuwien.ac.at/netsec-lab.

[38] ZHENG, C., JI, L., PEI, D., WANG, J., AND FRANCIS, P. A light-weight distributed scheme for detecting ip prefix hijacks in real-time. In *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications* (New York, NY, USA, 2007), SIGCOMM '07, ACM, pp. 277–288.