See How ISPs Care: An RPKI Validation Extension for Web Browsers

Matthias Wählisch
Freie Universität Berlin
m.waehlisch@fu-berlin.de

Thomas C. Schmidt
HAW Hamburg
t.schmidt@haw-hamburg.de

ABSTRACT

The Resource Public Key Infrastructure (RPKI) allows BGP routers to verify the origin AS of an IP prefix. In this demo, we present a software extension which performs prefix origin validation in the web browser of end users. The browser extension shows the RPKI validation outcome of the web server infrastructure for the requested web domain. It follows the common plug-in concepts and does not require special modifications of the browser software. It operates on live data and helps end users as well as operators to gain better insight into the Internet security landscape.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Routing Protocols*

Keywords

BGP, RPKI, secure inter-domain routing, deployment, web

1. INTRODUCTION

The successful hijack of an IP prefix within the Internet backbone—either intended or by misconfiguration—is a severe problem albeit it happens surprisingly often. For end users, the consequences of such incidents have been nicely illustrated when Pakistan Telecom claimed to own the IP prefix of YouTube, and China Telecom maliciously announced more than ≈ 37k IP prefixes. Web sites went offline, which was noticeable by the end users but they could not discover why. Furthermore, it was speculated whether parts of the traffic were intercepted and forwarded to its correct destination. Then an end user who opened a website would not necessarily be able to experience any difference.

Most recently the deployment of basic counter measurements against prefix hijacking started. However, those activities on the BGP layer are hardly visible for the end user. The end user needs to rely on the ISP. Note that SSL/TLS helps on the application layer to reveal packet interception but an attacker may forge certificates.

SIGCOMM '15 August 17-21, 2015, London, United Kingdom

ACM ISBN 978-1-4503-3542-3/15/08.

DOI: http://dx.doi.org/10.1145/2785956.2790034

In this abstract we argue for an open and user-friendly view on the current state of the Internet backbone security. We present a proof-of-concept which checks the IP prefix of the requested domain name within a web browser.

In the following, we briefly present the background of currently deployed security measures in the Internet backbone (§ 2) and describe design principles and implementation details of our application (§ 3).

2. RPKI, RTR, AND ORIGIN VALIDATION

To prevent simple prefix hijacking, i.e., an autonomous system (AS) maliciously claims successfully the ownership of an IP prefix, BGP peers need to verify the origin AS of an IP prefix carried in BGP updates. The **Resource Public Key Infrastructure (RPKI)** is a distributed repository that includes attestation objects to prove the ownership of IP resources (AS numbers, IP prefixes). The so called Route Origin Authorizations (ROAs) objects implement the binding between origin AS number and IP prefix.

RPKI-enabled routers [2] do not store ROAs itself but only the validated content of these authorities. The cryptographic validation of ROAs will be performed by trusted cache servers, which will be deployed at the network operator. The **RPKI/ RTR protocol** [1] defines a standard mechanism to maintain the exchange of the prefix to origin AS mapping between the cache server and routers. In combination with a BGP **prefix origin validation scheme** [4] a router is able to verify received BGP updates without suffering from cryptographic complexity. A BGP update can be valid, invalid, or not found (i.e., no information about this prefix in the RPKI).

The deployment of RPKI repositories and the creation of ROAs started in 2010. Major ISPs such as ATT and DTAG, as well as big companies such as Mozilla added their prefixes to the RPKI, and the quality of ROA data improved [6], [3]. However, the ratio of prefixes that cover web servers is low, in particular popular sites are less secured as preliminary results show [7]. We now describe how the RPKI validation can be performed natively within a browser.

3. RPKI VALIDATION IN WEB BROWSERS

Design The design of our solution is driven by real-time analysis and flexibility. To verify the BGP prefix of the web server a URL resolves to, basically the following steps are necessary: (a) DNS resolution of the web domain, (b) mapping of the IP address to prefix and origin AS visible in BGP, (c) comparison of the prefix/origin AS with ROA data. It is worth noting that the DNS resolution as well as

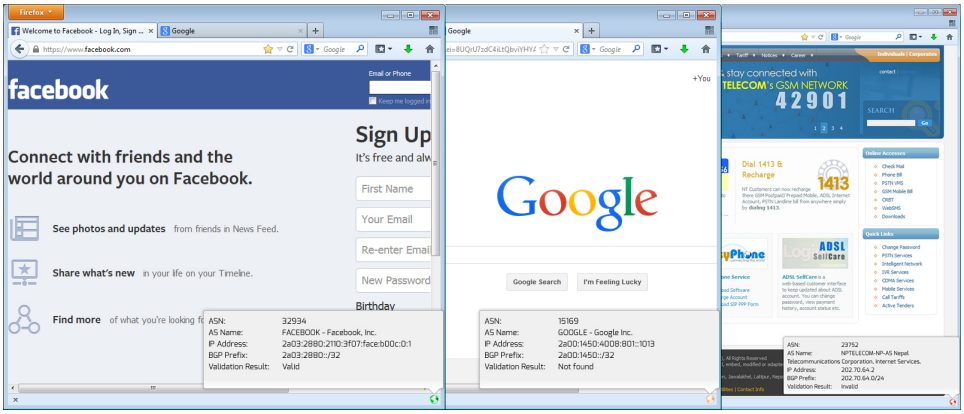

Figure 1: RPKI validation in Mozilla Firefox for the requested websites with different validation outcome

the BGP data depend on the location of the client. However, in contrast to the name to address mapping in the DNS, there is no standard mechanism to request the prefix/AS pair for an IP address with respect to the customer's ISP routing table. An accurate mapping might be a new service provided by ISPs in the future.

There are two options to implement origin validation in web browsers: (a) the browser extension implements the full router part (i.e., receives valid ROAs from cache server and performs origin validation of BGP data), (b) the extension resolves only the IP address of the web domain and a remote back-end performs the origin validation. We decide for the latter as this allows for easy applicability in most browser platforms, which usually provide add-on concepts based on JavaScript. Back-end and front-end communicate via HTTP as this is native in browsers and does not conflict with most firewall settings.

Implementation—Back-end Currently, the back-end uses the Team Cymru community service to resolve the IP prefix of the web server IP address and the corresponding origin AS. We admit that the result does not necessarily comply with the BGP entry of the client's upstream but the vantage points of Team Cymru provide a good coverage. Furthermore, our architecture is flexible enough to consider multiple BGP sources as well as future mappig services.

To fetch ROAs and to validate the BGP information, we deploy the RTRlib [5], an open source implementation of the RPKI/RTR router part. This C library is very efficient with respect to memory and processing resources. Per default, the implementation establishes RTR sessions to two cache servers for fallback reasons. However, end users can configure their own end point for a cache server in the browser extension, and multiple instances of the RTRlib will be started.

Implementation—Front-end The browser extension is implemented as dynamic add-on for Mozilla Firefox and Chrome. Other browsers can be easily supported as the browser extension only needs to support the REST interface to the back-end. The source code is available on Github[1]. The extension visualizes three states: green (the web server prefix is valid in the BGP), orange (the prefix was not found in the RPKI), and red (the prefix is invalid, the website might be suspicious), see Fig. 1. Note that if an attacker blocks plugin traffic, none of the three states apply, indicating that the communication to the trust anchor is broken. Advanced users can request information about the autonomous system/the IP prefix and configure the host address and port of the RPKI cache server, which will be used by the back-end.

[1]https://github.com/rtrlib

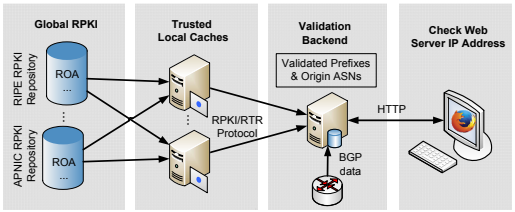

Figure 2: System architecture

Future Challenges Our current solution checks the BGP data for the web server infrastructure of the landing page. Many web pages include embedded content linked via different domains. Analyzing the HTML content fast and combine the different results to a complete picture for the whole web page will be part of our future work.

Acknowledgments We would like to thank Tas Sóti and Sebastian Meiling for helping on the implementation. This work is supported by the German BMBF within the project Peeroskop (http://peeroskop.realmv6.org).

4. REFERENCES

[1] Bush, R., and Austein, R. The Resource Public Key Infrastructure (RPKI) to Router Protocol. RFC 6810, IETF.

[2] Bush, R., Austein, R., Patel, K., Gredler, H., and Waehlisch, M. Resource Public Key Infrastructure (RPKI) Router Implementation Report. RFC 7128, IETF, 2014.

[3] Iamartino, D., Pelsser, C., and Bush, R. Measuring BGP route origin registration validation . In *Proc. of PAM* (Berlin, 2015), LNCS, Springer, pp. 28–40.

[4] Mohapatra, P., Scudder, J., Ward, D., Bush, R., and Austein, R. BGP Prefix Origin Validation. RFC 6811, 2013.

[5] Wählisch, M., Holler, F., Schmidt, T. C., and Schiller, J. H. RTRlib: An Open-Source Library in C for RPKI-based Prefix Origin Validation. In *Proc. of USENIX Security Workshop CSET'13* (Berkeley, 2013), USENIX Assoc.

[6] Wählisch, M., Maennel, O., and Schmidt, T. C. Towards Detecting BGP Route Hijacking using the RPKI. *ACM Computer Communication Review 42*, 4 (2012), 103–104.

[7] Wählisch, M., Schmidt, R., Schmidt, T. C., Maennel, O., and Uhlig, S. When BGP Security Meets Content Deployment: Measuring and Analysing RPKI-Protection of Websites. Technical Report arXiv:1408.0391, Sep. 2014.