Argus: An Accurate and Agile System to Detecting IP Prefix Hijacking

Yang Xiang*†, Zhiliang Wang*‡, Xia Yin*† and Jianping Wu*†‡
*Tsinghua National Laboratory for Information Science and Technology (TNList), Beijing, P. R. China, 100084
†Department of Computer Science & Technology, Tsinghua University, Beijing, P. R. China, 100084
‡Network Research Center, Tsinghua University, Beijing, P. R. China, 100084
Email: {xiangy08,wzl,yxia,jianping}@csnet1.cs.tsinghua.edu.cn

***Abstract*—The *de facto* inter-domain routing protocol, Border Gateway Protocol (BGP), plays a critical role in the Internet routing reliability. Invalid routes generated by mis-configurations or malicious attacks will devastate the Internet routing system. In the near future, deploying a secure BGP in the Internet to completely prevent hijacking is impossible. As a result, lots of hijacking detection systems have emerged. However, they have more or less weaknesses such as long detection delay, high false alarm rate or deploy hardness. This paper proposes Argus, an agile system to fast and accurate detect prefix hijacking. Argus already keeps on running in the Internet for two months and identified several possible hijackings. Initial results show that it usually discovers a hijacking in less than ten seconds, and can significantly decrease the false alarm rate.**

I. Introduction

The Internet is composed of tens of thousands of Autonomous Systems (ASes) which operate individual parts of the infrastructure. As Border Gateway Protocol (BGP) [1] controls the packet forwarding path between ASes, it plays a critical role in Internet efficiency and reliability. However, because of the lack of security considerations, several security problems have not been well resolved yet [2]. In BGP, route information received from neighbors can not be validated. Invalid routes may cause packets being forwarded along wrong AS paths. As shown in Figure 1, AS1 announces a route for its prefix f with an AS path $\langle 1 \rangle$. After this announcement, AS2 and AS3 will get a route to f. However, AS4 can also announce a route for f with a forged AS path $\langle 4 \rangle$, which is shorter than the valid path received by AS3. As the forged route is considered as a valid path, then AS4 hijacks the traffic from AS3 to f.

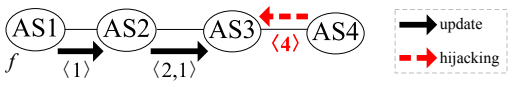


Fig. 1. An Example of Prefix hijacking in BGP

Prefix hijacking often generated by accidental mis-configurations. It may incur serious routing problems and economic losses [3]. For instance, on Feb. 24th 2008, Pakistan Telecom hijacked YouTube for two hours [4]. On Apr. 4th 2010, one AS of China Telecom hijacked more than 50,000 prefixes (15% of the Internet) which belong to 170 different countries [5].

In order to improve the security of BGP, several methods have been proposed, which fall into two categories: cryptographic based prevention and anomaly detection. Cryptographic approaches, such as [6], usually use public key infrastructure (PKI) to ensure the authentication of routing announcements. However, these solutions consume significant extra router resources of computation and storage and will not be rolled out for some years [7]. On the other hand, anomaly detection approaches [8]–[12] aim to discovering anomalous information in BGP announcements. However, they have more or less weaknesses such as long detection delay, high false alarm rate or hard to deploy.

In this paper, we present an accurate and agile system to detecting IP prefix hijacking. Our proposal, *Argus*, exploits a key observation about IP prefix hijacking: polluted routers usually can not get a reply from the victim prefix. In other words, after correlating the control-plane anomaly and data-plane reachability in a large number of public route-servers and look-glasses (we call them *eyes of Arugs*), Argus can accurately distinguish prefix hijacking from legitimate BGP anomalous.

There are four key contributions in this paper: (i) **low false positive rate** – Argus deeply correlates control- and data-plane information to further improves the detection accuracy; (ii) **low false negative rate** – Argus covers almost all kinds of routing anomalies appear in control-plane, including origin AS anomaly, neighbor AS anomaly, and routing policy anomaly; (iii) **fast** – Argus can detect prefix hijacking in seconds after the first abnormal route appears, while existing method often need minutes or even tens of minutes; (iv) **agile** – Argus is easy to implement and to deploy, it only use existing public services directly and do not need to install any new software in these external points; (v) **real system** – Argus already keeps on running in the Internet for two months, and figured out tens of possible hijackings. With the popular and rapid massage deliver service such as *twitter*, Argus can inform the community immediately when the hijacking happens.

This paper is organized as follows. Section II reviews existing hijacking detection methods. Section III proposes Argus. Section IV illustrates possible prefix hijackings in past two months detected by Argus. Finally section V is the conclusion.

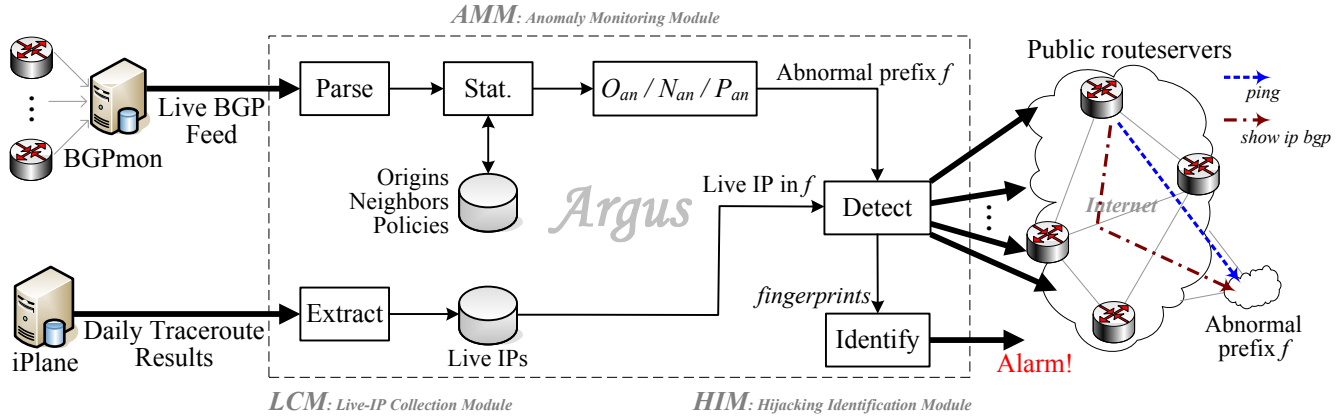


Fig. 2. Architecture of Argus

II. Background

BGP is a path vector routing protocol which maintaining connectivity between ASes. An AS manages a number of networks expressed as IP prefixes in its domain. In this paper, Links between ASes are defined as AS edges, and ASes sharing a common edge are called neighbors. A path $p = \langle a_n, \cdots, a_i, \cdots, a_0 \rangle$ in BGP is a sequence of ASes. The last AS in the path, a_0, is normally the origin of the path. Each AS announces BGP update to its neighbors after padding itself in the front of the received path. Any change of network reachability will generate update through the Internet. In addition, BGP is a policy-based routing protocol. An AS only exports routes to a neighbor if it is willing to forward traffic to the prefix from that neighbor.

A. Prefix Hijacking

Based on how the manipulator deals with the hijacked traffic, academics classify prefix hijackings into the following three categories:

1) Blackholing: the manipulator simply drops the attracted packets.
2) Imposture: the manipulator responds to senders of the hijacked traffic, mimicking the behavior of the hijacked prefix.
3) Interception: the manipulator forwards the hijacked traffic to the hijacked prefix after eavesdropping or recording the information in the packets.

Since Internet is connected by mutual-trusting ASes, prefix hijackings always caused by unintentional mis-configurations. Thus, almost all the hijackings will simply drop the attracted packets. Our proposed system, Argus, only consider the blackholing hijacking in this paper.

B. Related Work

Most existing hijacking prevention proposals [6], [8], [9], [13]–[15] require changes to router software, router configurations, network operations, and some of them require public key infrastructures. These solutions are therefore not easily deployable. On the other hand, hijacking detection proposals, which are more realistic, fall into three categories, based on the type of information they used.

TABLE I
Pros and Cons of Control-plan and Data-plane Methods

Methods	Pros	Cons
Control-plane	1. Realtime (live BGP feed) 2. Luxuriant information (victim and attacker)	1. Very low accuracy 2. Lots of alarms
Data-plane	1. High accuracy 2. Few alarms	1. Heavy-weight, not scalable 2. Lack of attacker's info. 3. Minutes of detection delay 4. Can't detect sub-prefix hijacking

The first category of control-plane approaches [16]–[18] performs passive monitoring of BGP routing information to detect anomalous behavior. They are easily deployable, but can be fairly inaccurate and raise lots of alarms due to legitimate reasons for anomalous updates [19] or inaccurate routing registry data they used. Besides, they usually only consider origin AS anomalies.

The second category [11], [12] only relies on real-time data-plane information. However, they need to continuously probe a large number of networks in the entire Internet and hence suffer from poor scalability. Besides, even they notify a prefix hijacking, it is hard to figure out the attacker since lack of control-plane information. In addition, they usually use traceroute to probe target networks thus the detection delay can reach up to several minutes. Finally, they can not detect sub-prefix hijacking since there is no way to know beforehand which piece of address may be hijacked.

Table I summarizes the pros and cons of previous two kinds of detection methods. These methods have complementary advantages and disadvantages. Recently, utilizing data-plane information together with control-plane information in hijacking detection is gaining attention [10]. It achieves the same level of accuracy compare with data-plane approaches, and reduces the probing pressure since probing only triggered by the control-plane anomalies. However, its detection delay may reach up to tens of minutes [10]. Besides, the whole system is

not so easy to implement and to deploy, since they are highly relay on the ability of external detection nodes. In order to execute complicated commands such as *nmap*, [10] can only use versatile detection points such as PlanetLab hosts. As a result, it can only obtain data-plane information from those end hosts but not control-plane routing.

Our approach, Argus, is also based on the integration between control- and data- plane information. Unlike [10], it has the following advantages: (1) Argus deeply correlated control- and data- plane information in every detection node to further improve detection accuracy, while [10] only independently consider data-plane information after routing anomaly occurs; (2) Argus covers a border range of anomalies in control-plane monitoring process, including origin anomaly and path anomaly, while [10] only consider origin anomaly; (3) most importantly, Argus can detect a possible hijacking in seconds after the first abnormal route appears; (4) Argus is easy to implement and to deploy, it only need to execute very simple commands such as *ping* and *show ip bgp* in external nodes, these commands are always available in public route-servers.

III. ARGUS

A. Overview

Figure 2 shows the architecture of our hijacking detection system: *Argus*. The whole system consists of three modules: Anomaly Monitoring Module (*AMM*), Hijacking Identification Module (*HIM*), and Live-IP Collection Module (*LCM*).

The Anomaly Monitoring Module receives live BGP updates from BGPmon [20], a well known and public available real-time BGP feed. BGPmon collects BGP updates from routers all over the world. Recently it connects to 130 routers in 74 ASes, including 27 routers in 11 Tier 1 ASes (covers almost all Tier 1 ASes [21]). And more importantly, the number of BGP peers continues to grow. When the AMM receives an update from BGPmon, it will check whether the update contains anomaly according to the routing information database. Section III-B will discuss the details of anomaly discovery.

Once an anomalous update is discovered in the AMM, the Hijacking Identification Module will be activated immediately. This module will simultaneously login to hundreds of public available route-servers and looking-glasses [22] which we call *eyes of Argus*. Suppose the anomaly route discovered in AMM (defined as r_0) is towards prefix f. Then in every eye, Argus queries the best BGP route for f, to check whether the eye has been polluted by the anomaly route r_0. At the same time, Argus probes a live IP (get from local live IP database) in prefix f from every eye, to detect whether f is reachable from the eye. After gathered both control- and data- plane information in every eye, Argus then correlates this information, calculates fingerprints along time, and decides whether or not to raise a hijacking alarm. This process will repeat until a pre-defined detection window W expires. Details of the HIM will be discussed in section III-C.

Beside the above two main modules, Argus also has a assistant module, Live-IP Collection Module, to gather recent live IPs in every routable prefix. We collected a set of IP addresses from the following two sources:

1) x.x.x.1 for each prefix in the global routing table.
2) IPs appear along the traceroute paths in iPlane [23] daily result. iPlane is a project which performs traceroutes from various vantage points to construct a router level atlas of the Internet.

Since IPs from these two sources may not be globally routable, we also use *ping* command to test these candidate IPs and filter out unresponsive ones.

As we can see, Argus does not install additional software into external hosts or routers. It only receives live BGP feed and traceroute daily result, and login to route-servers and looking-glasses to execute simple commands.

B. Anomaly Discovery

Existing detection methods always only consider origin anomalies. However, AS path also can be mis-configured (i.e., the *as-path prepend* command). Unfortunately, it is too hard to figure out whether or not an AS path is abnormal since feasible AS paths are too flexible. In order to achieve a wider coverage, the AMM in Argus checks three kinds of anomalies, as shown in Figure 3. Suppose the received AS path $p = \langle a_n, \cdots, a_0 \rangle$, then:

1) Origin Anomaly (O_{an}): the origin AS a_0 of p is anomalous. For example, in Figure 3a, AS3 is not the origin AS of prefix f, thus path $\langle 3 \rangle$ for f is anomalous.
2) Neighbor Anomaly (N_{an}): an adjacent AS pair (a_i, a_{i-1}) in path p is anomalous. For example, in Figure 3b, AS3 does not directly connect to AS1, so the AS pair $(3, 1)$ is not in the routing information database, and then path $\langle 3, 1 \rangle$ is anomalous.
3) Policy Anomaly (P_{an}): an adjacent AS triple (a_{i+1}, a_i, a_{i-1}) in path p is anomalous. For example, in Figure 3c, although AS3 directly connects to AS1, it should not announce routes learned from its provider AS1 to another provider AS2. After that, if AS2 announces a path $\langle 2, 3, 1 \rangle$, the AS triple $(2, 3, 1)$ becomes a P_{an} and Argus will detect it. Note that we don't care about specific business relationships, and we only consider abnormal AS triples.

Here, *anomaly* means that the origin (or AS pair/triple) does not appear in our routing information database before. Besides, the database will be updated periodically. Origins, AS pairs, and AS triples, if not alive for more than one month, will be removed.

We believe that, using the above three kinds of anomalies, Argus can cover almost all kinds of hijackings. BGP is a policy-based routing protocol. An AS only exports a route to a neighbor if it is willing to forward traffic to the corresponding prefix from that neighbor. Although complex policies (i.e., route filters [24]) exist, AS usually does not differentiate between prefixes or nonadjacent ASes. For example, in path $p = \langle a_{i+1}, a_i, a_{i-1}, \cdots, a_0 \rangle$, when a_i decides whether routes learned from a_{i-1} can be exported to a_{i+1}, it only considers its

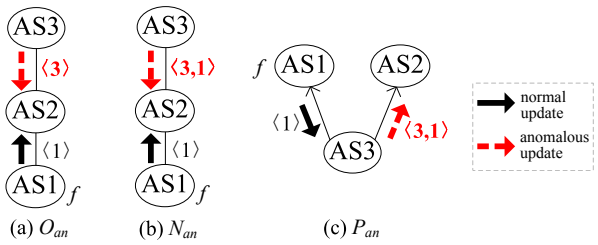


Fig. 3. Three Kinds of Anomalies Considered by Argus

relationships with the two neighbors (a_{i-1} and a_{i+1}), but does not consider other ASes along the path ($a_{i-2}, ..., a_0$). This kind of Neighbor-Based Import and Export policy, which we call it *NBIE*, abstracts the basic functionality of BGP. According to our measurement results in $whois$ database, only a small portion of routing polices (route filters) violate the NBIE rule. An adjacent AS triple (a_{i+1}, a_i, a_{i-1}) actually describes the routing policy of its owner a_i, and implies that a_i can export all routes imported from a_{i-1} to a_{i+1}. Thus, the three kinds of abnormalities in Figure 3 can cover almost all scenarios of route anomalies caused by prefix hijacking.

Once the AMM discovered a anomaly, it will notify the HIM to verify whether it is a hijacking. If it is not a hijacking, the AMM should update the routing information database and insert the new origin, or AS pairs/triples.

C. Hijacking Identification

While an anomalous route for prefix f is discovered by the AMM, Argus will activate the identification process. It will launch two kinds of threads: threads for control-plane querying (*C-Thread*), and threads for data-plane probing (*D-Thread*). These threads are responsible for continually gathering the BGP routes and probing response of f respectively.

In the t-th second after the anomaly occurred ($0 < t < W$, W is the pre-defined detection window), all C-Threads will together construct a vector $C_t = \{c_{t,j} | 1 \leq j \leq m\}$ (m is the number of eyes), according to the best BGP route $r_{t,j}$ for prefix f in j-th eye at time t:

$$c_{t,j} = \begin{cases} 0 & \text{if } r_{t,j} \text{ contains the anomaly} \\ 1 & \text{if } r_{t,j} \text{ does not contain the anomaly} \end{cases}$$

Specifically, C-Threads execute *show ip bgp* command on each eye to extract the current best route for the anomaly prefix f. For O_{an}, C-Threads check that if every $r_{t,j}$ $(1 \leq j \leq m)$ has the anomaly origin AS. For N_{an} and P_{an}, C-Threads check whether every $r_{t,j}$ $(1 \leq j \leq m)$ contains the abnormal AS pair or AS triple.

At the same time, all D-Threads will construct a vector $D_t = \{d_{t,j} | 1 \leq j \leq m\}$ according to the data-plane reachability of prefix f in j-th eye at time t:

$$d_{t,j} = \begin{cases} 1 & \text{if the probe to } f \text{ get a reply} \\ 0 & \text{if the probe to } f \text{ does not get a reply} \end{cases}$$

Specifically, we execute *ping* command on each eye to detect whether the target prefix f is reachable from that eye.

TABLE II
POSSIBLE REASONS WHEN D_t AND C_t HAVE DIFFERENT RELATIONSHIPS.

Relationships of D_t and C_t		Possible Reasons
Unrelated	$d_{t,j} = 0$	Firewall, Not Alive
	$d_{t,j} = 1$	Multiple Origin AS
Negatively Correlated	$d_{t,j} = 1 - c_{t,j}$	Origin AS Changing
Positively Correlated	$d_{t,j} = c_{t,j}$	Prefix Hijacking

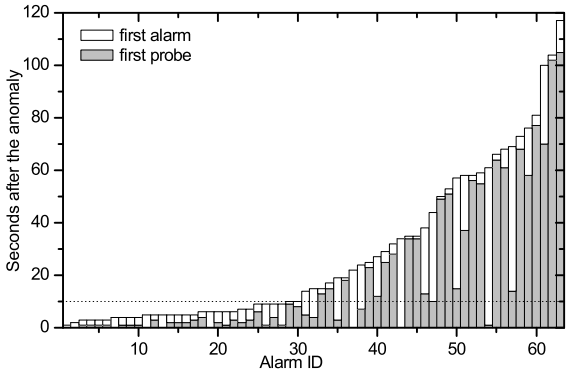


Fig. 4. Detection Delays in Recent Two Months

Since ping is much faster than other probing command (i.e., traceroute), the detection delay can be extremely short.

BGP controls the packet forwarding path between ASes. The data-plane reachability is highly related to the control-plane routing information. If we know C_t and D_t, than we can infer possible reasons of the routing anomaly according to their relations, as shown in Table II.

If D_t and C_t are positively correlated, there is a strong possibility that the anomaly is generated by a prefix hijacking. However, since routing information may inconsistent and instable in the whole Internet during the routing convergence, these two vectors usually do not have a strict correlation. We utilize the *correlation coefficient* of D_t and C_t to determine how closely they are related:

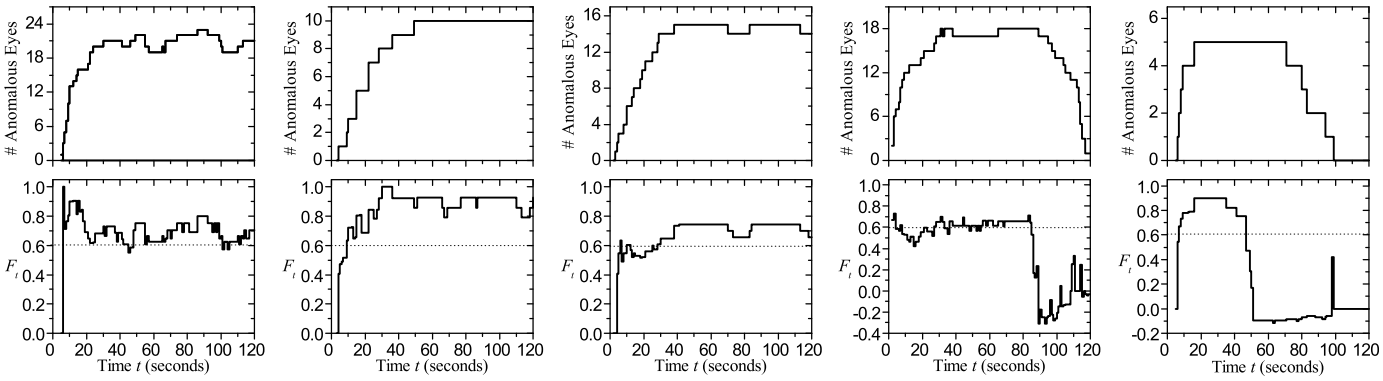
$$F_t = \frac{\sum_{j=1}^{m}(c_{t,j} - E(C_t))(d_{t,j} - E(D_t))}{\sqrt{\sum_{j=1}^{m}(c_{t,j} - E(C_t))^2 \times \sum_{j=1}^{m}(d_{t,j} - E(D_t))^2}}$$

In this equation, $E(C_t)$ is the average of $c_{t,j}$ $(1 \leq j \leq m)$, and $E(D_t)$ is the average of $d_{t,j}$ $(1 \leq j \leq m)$.

The correlation coefficient F_t, which we call it the *fingerprint* of the anomaly at a specified time t, implies the root cause of the anomaly. If F_t is close to 1.0, then there is a high possibility that it is a prefix hijacking. We use a threshold μ to make the judgment. If $F_t > \mu$, Argus will raise a hijacking alarm. A higher μ will achieve a lower false positive rate, but also results in a higher false negative rate. There is a tradeoff in practice.

TABLE III
POSSIBLE HIJACKINGS IN RECENT TWO MONTHS

ID	Time of Anomaly mm-dd hh:mm:ss	Prefix (*whois*)	Normal Origin AS (*whois*)	Abnormal Origin AS (*whois*)	First Alarm	Alarm Lasts
1	06-10 03:36:10	210.14.74/23 (Sci. & Tech. Network Comm. CN) *and other 3 prefixes*	AS17775 (Shanghai Guangdian Elec., CN)	AS4812 (China Telecom, CN)	6	111
2	05-20 22:09:53	199.120.249/24 (TeamQuest Corp., US)	AS1239 (Sprint, US)	AS5056 (Iowa Network Services, US)	9	111
3	05-21 15:32:57	204.124.199/24 (Coca Cola, US)	AS2686 (AT&T, US)	AS4589 (Easynet Global Services, EU)	6	96
4	04-28 19:13:14	208.5.103/24 (NASCAR, US)	AS30171 (NASCAR, US)	AS40457 (ISC Motorsports, US)	25	95
5	06-22 12:21:12	193.227.117/24 (1api GmbH, DE)	AS26753 (In2net Network Inc., CA)	AS29066 (Velia.net, UK)	10	83
6	05-27 11:44:53	209.255.224/24 (Eisai Research Inst., US)	AS46783 (Eisai Research Inst., US)	AS1239 (Sprint, US)	6	78
7	06-23 01:11:37	76.72.238/24 (Townes Telecom, US) *and other 2 prefixes*	AS701 (Verizon, US)	AS27005 (Pinpoint Comm., US)	6	73
8	06-26 03:04:12	64.74.95/24 (Basswood Partners, US)	AS29783 (Basswood Partners, US)	AS29784 (Greenlight Capital, US)	15	69
9	05-20 15:43:03	202.3.14/24 (Infomedia, ID)	AS55698 (Infomedia, ID)	AS17670 (Infokom, ID)	5	65
10	06-08 07:38:01	72.19.192/18 (EZ Publishing, US)	AS19864 (O1 Comm., US)	AS18687 (MPower Comm., US)	61	56
11	06-18 19:21:30	64.136.120/24 (MCI Comm, US)	AS813 (MCI Comm, CA)	AS33007 (NCS Technologies, CA)	2	55
12	04-29 22:43:47	66.117.128/19 (LanMinds, US)	AS7235 (LanMinds, US)	AS26803 (Fiber Internet Center, US)	7	40
13	06-22 14:47:03	119.252.226/24 (PNG-ARNET, PG)	AS9229 (SPEEDCAST Limited, HK)	AS9731 (ST Teleport Pte Ltd, SG)	9	40
14	06-24 19:00:11	46.182.111/24 (YISP, NL)	AS20495 (We Dare BV, NL)	AS47869 (Netrouting Data Facilities, NL)	5	35
15	05-02 08:13:48	92.122.64/22 (Akamai, EU) *and other 7 prefixes*	AS39369 (Port80, SE)	AS16150 (Port80, SE)	44	18
16	04-27 22:45:56	178.253.102/23 (190 Syrian, SY) *and other 16 prefixes*	AS29256 (Syrian Telecom, SY)	AS29386 (Syrian Telecom, SY)	38	12
17	05-27 04:10:54	203.24.50/23 (Universitas Tanjungpura, ID)	AS55687 (Universitas Tanjungpura, ID)	AS18007 (Indonesia Higher Education, ID)	34	12
18	05-02 07:06:33	12.153.160/24 (Avnet, US) *and other 123 prefixes*	AS14990 (Extreme Networks, US)	AS9848 (SEJONG Telecom, KR)	4	11



(a) Hijacking #1 (b) Hijacking #2 (c) Hijacking #3 (d) Hijacking #11 (e) Hijacking #12

Fig. 5. Fingerprints of Some Hijackings

IV. RECENT ALARMS

Argus has been running in the Internet for two months, started from 26th Apr 2011. At present, Argus sets $\mu = 0.6$ as the alarm threshold, and $W = 120$ seconds as the size of detection window. Since most of route change events in the Internet (especially T_{short} and T_{long} which are similar to prefix hijacking) can converge to a stable state in less than 1 minute [25], the 120-seconds detection window is enough. Every time there is an anomaly in the AMM, the HIM will create $m = 40$ C-Threads and 40 D-Threads to gather necessary information, calculate fingerprints, and raise hijacking alarms if needed. In the near future, Argus will connect to more look-glasses [22].

In the past two months, Argus monitored 11688 anomalies and finally figured out 63 possible hijackings. Figure 4 illustrates all these hijackings, sorted by their detection delay (seconds elapsed to raise the first alarm after the anomaly discovered in the AMM). Half of alarms in Figure 4 have a detection delay less than 10 seconds. There are some worse cases which have a detection delay longer than 30 seconds. They are mainly caused by low efficiency of our early code. However, although the identification process launches a little late due to some technical reasons, it usually can raise the first alarm in seconds after the first probe.

In Table III, we pick out all the 18 possible hijackings which with alarms ($F_t > \mu$) for more than 10 seconds.

Among them, 12 hijackings were detected in no longer than 10 seconds. All these 18 hijackings, except for #15 and #16 (since the abnormal and normal origin AS belong to the same organization), are believed to be prefix hijacking with a high probability. It is worth noting that, the suspect attacker (AS9848) in Hijacking #18 possibly hijacked totally 124 prefixes at the same time.

Figure 5 shows the fingerprints F_t and the number of polluted eyes of five cases in Table III. As we can see, even short time hijackings, such as Hijacking #11 in Figure 5d and Hijacking #12 in Figure 5e, also can be identified by Argus in less than 10 seconds. Since the global state of inter-domain routing is inconsistent during the prefix hijacking and Argus currently only has a small number of eyes, sometimes the fingerprint F_t is not very close to 1. In the future, Argus will connect to all public available looking-glasses and route-servers [22], and then the accuracy will be observably promoted.

BGP is so flexible, that so far there is not an automatic detection method can perfectly figure out hijackings from legitimate anomalies. We also can not conclude that all these routing events listed in Table III must be prefix hijackings. The meaning of Argus (and all the other hijacking detection systems) is that, it can tremendously reduce the false positive rate and liberate network operators from vast alarms (11688 anomalies in two months, 194.8 per day). In the future, we plan to send emails to the network operators of both victims and attacks after an alarm raised, to verify the accuracy of our system.

V. Conclusions and Future Works

In this paper, we present a accurate and agile system, Argus, to detecting IP prefix hijacking. Argus covers almost all kinds of anomalies in control-plane, including origin AS anomaly, neighbor AS anomaly, and routing policy anomaly. Argus can accurately detect a prefix hijacking in seconds after the first abnormal route appears, while existing methods often need minutes or even tens of minutes. Besides, Argus is easy to implement and to deploy, it only use existing public services directly and do not need to install any new software in these public advantage points. Having been running in the Internet for two months, Argus already figured out tens of possible hijackings.

At present, Argus only controls about 40 eyes. There are a large number of public looking-glasses, we plan to use them to enhance the accuracy of our system. In the future, we hope Argus can become a fast and accurate source of prefix hijacking alarms that helping network operators monitoring their ASes.

Acknowledgment

This work is supported by (1) the National Key Technology R&D Program of China under Grant No. 2008BAH37B03, and (2) the National Basic Research Program of China (973 Program) under Grant No. 2009CB320502.

References

[1] Y. Rekhter, T. Li, and S. Hares, "Rfc 4271: Border gateway protocol 4," http://tools.ietf.org/html/rfc4271, 2006.
[2] S. Murphy, "Rfc 4272: Bgp security vulnerabilities analysis," http://tools.ietf.org/html/rfc4272, 2006.
[3] R. Mahajan, D. Wetherall, and T. Anderson, "Understanding bgp misconfiguration," in *SIGCOMM*, 2002, pp. 3–16.
[4] RIPE, "Youtube hijacking: A ripe ncc ris case study," http://www.ripe.net/news/study-youtube-hijacking.html, 2008.
[5] Renesys, "China's 18-minute mystery," http://www.renesys.com/blog/2010/11/chinas-18-minute-mystery.shtml, 2010.
[6] S. Kent, C. Lynn, J. Mikkelson, and K. Seo, "Secure border gateway protocol (s-bgp)," *IEEE Journal on Selected Areas in Communications*, vol. 18, pp. 103–116, 2000.
[7] S. M. Bellovin, R. Bush, and D. Ward, "Security requirements for bgp path validation," http://tools.ietf.org/html/draft-ymbk-bgpsec-reqs-02, 2011.
[8] L. Subramanian, V. Roth, I. Stoica, S. Shenker, and R. H. Katz, "Listen and whisper: Security mechanisms for bgp," in *NSDI*, 2004, pp. 127–140.
[9] J. Karlin, S. Forrest, and J. Rexford, "Pretty good bgp: Improving bgp by cautiously adopting routes," in *ICNP*, 2006, pp. 290–299.
[10] X. Hu and Z. M. Mao, "Accurate real-time identification of ip prefix hijacking," in *IEEE Symposium on Security and Privacy*, 2007, pp. 3–17.
[11] Z. Zhang, Y. Zhang, Y. C. Hu, Z. M. Mao, and R. Bush, "ispy: Detecting ip prefix hijacking on my own," in *SIGCOMM*, 2008, pp. 327–338.
[12] C. Zheng, L. Ji, D. Pei, J. Wang, and P. Francis, "A light-weight distributed scheme for detecting ip prefix hijacks in real-time," in *SIGCOMM*, 2007, pp. 324–334.
[13] W. Aiello, J. Ioannidis, and P. D. McDaniel, "Origin authentication in interdomain routing," in *ACM Conference on Computer and Communications Security*, 2003, pp. 165–178.
[14] R. White, "Architecture and deployment considerations for secure origin bgp (sobgp)," http://tools.ietf.org/html/draft-white-sobgp-architecture-02, 2006.
[15] P. C. van Oorschot, T. Wan, and E. Kranakis, "On interdomain routing security and pretty secure bgp (psbgp)," *ACM Trans. Inf. Syst. Secur.*, vol. 10, no. 3, 2007.
[16] "Ripe myasn system," http://www.ris.ripe.net/myasn.html.
[17] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang, "Phas: A prefix hijack alert system," in *USENIX*, 2006.
[18] Y.-J. Chi, R. Oliveira, and L. Zhang, "Cyclops: The as-level connectivity observatory," *ACM SIGCOMM Computer Communication Review*, pp. 7–16, 2008.
[19] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. F. Wu, and L. Zhang, "An analysis of bgp multiple origin as (moas) conflicts," in *1st ACM SIGCOMM Workshop on Internet Measurement*, 2001.
[20] "The bgpmon project," http://bgpmon.netsec.colostate.edu, 2011.
[21] "Tier 1 network," http://en.wikipedia.org/wiki/Tier_1_network, 2011.
[22] "Public route-servers and looking-glasses," http://www.traceroute.org, 2011.
[23] H. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, and A. Krishnamurthy, "iplane: An information plane for distributed services," in *OSDI*, 2006, pp. 367–380.
[24] C. Alaettinoglu, C. Villamizar, E. Gerich, D. Kessens, D. Meyer, T. Bates, D. Karrenberg, and M. Terpstra, "Rfc 2622, routing policy specification language (rpsl)," http://tools.ietf.org/html/rfc2622, 1999.
[25] R. Oliveira, B. Zhang, D. Pei, R. Izhak-Ratzin, and L. Zhang, "Quantifying path exploration in the internet," in *Proc. of the 6th ACM SIGCOMM Internet Measurement Conference (IMC)*, Rio de Janeriro, Brazil, 2006.