Dynamic Monitoring of Dark IP Address Space [⋆]

Iasonas Polakis, Georgios Kontaxis,
Sotiris Ioannidis and Evangelos P. Markatos
email: {polakis, kondax, sotiris, markatos}@ics.forth.gr

Institute of Computer Science - Foundation for Research and Technology, Hellas

Abstract. A number of security-related research topics are based on the monitoring of dark IP address space. Unfortunately there is large administrative overhead associated with the dynamic assignment of a specific subnet for monitoring purposes, such as the deployment of a honeypot farm or a distributed intrusion detection system. In this paper, we propose a system that enables the dynamic allocation of an unadvertised IP address subnet for use by a monitoring sensor. The system dynamically selects network subnets that have been allocated to the organization but are not being advertised, advertises them, and subsequently forwards all received traffic destined to the selected subnet to a monitoring sensor.

1 Introduction

An important area of Internet research focuses on monitoring of computer networks. Particularly in security-related fields, unused IP address subnets are considered a valuable resource which can enable the collection and analysis of attack traffic. By deploying intrusion detection systems, monitoring systems or honeypot farms (all of which will be referred to as *sensors*), researchers can collect vast amounts of traffic data. As a result, it is very common for researchers across organizations to collaborate and "donate" IP address subnets that are not in use.

The deployment of sensors entails a high administrative overhead. This procedure consists of several phases. First, if the subnet has been allocated but is not being advertised via some routing protocol, it must be, so that it becomes reachable by neighboring networks and may receive attack traffic. Next, routing tables inside the internal network must be altered to forward all traffic destined to the subnet to a specific machine, in our case the *sensor*. Finally, in cases where the subnet must be revoked or substituted with another one, all changes must be done manually which is prone to human error.

To facilitate the dynamic handling of subnets for the deployment of network monitoring sensors, we propose a system that will automate the procedure. The system will be aware of the dynamic routing protocol that is in effect both inter-domain and intra-domain, select the unused subnet that will be monitored, advertise it to the appropriate neighboring routers, and update routing tables so as to forward incoming traffic to the sensor. It will also detect when previously unused subnets are advertised, therefore claimed for use by the organization, and automatically release the specific subnet in respect to the privacy of its users.

[⋆] This work was supported in part by the project SysSec funded in part by the European Commission, under Grant Agreement Number 257007.

Our system, consists of 3 different components. The core component of the system is responsible for managing the other components and keeping an overview of the subnets used by the organization. It also keeps a history of all subnets advertised by an organization's router and decides which unadvertised subnets can be allocated for monitoring. The second component receives commands from the core component that instructs it to start advertising subnets that are not being advertised by the organization and which will be forwarded to the monitoring sensor. It also detects when monitored subnets are advertised by the organization, and informs the core. The final component receives commands from the core and dynamically alters the organization's internal routing tables and adds or removes entries that forward traffic to the monitoring sensor.

2 Related Work

Quagga [1] is a free, open-source network routing software suite providing implementations of protocols such as OSPF, RIP and BGP for Unix platforms. A system with Quagga installed acts as a dedicated software router. By supporting both OSPF and BGP it may be used for inter-domain as well as intra-domain routing. Quagga exchanges routing information with other, neighboring routers using routing protocols. It uses this information to update the routing table of the Unix kernel.

MAPI [5] offers an API for generic passive network monitoring based on the *network flow* abstraction, which enables users to communicate their needs to the underlying traffic monitoring platform. Moreover, MAPI offers the capability of distributed network monitoring using multiple remote monitoring sensors, and supports several different hardware platforms. DECON [3] is a decentralized and scalable coordination system that aims to solve the problem of flow assignment among a set of monitoring sensors. A peer-to-peer overlay network receives reports from all sensors that see a specific flow, and subsequently assigns the flow to one of the sensors based on a first-fit or best-fit strategy. Luca Deri [1] proposes a new dynamic packet filtering technique which overcomes the limitation of BPF by allowing users to specify several filters simultaneously and add or remove filters dynamically without any reconfigurations or downtime. Another dynamic packet filter, Swift [6], three orders of magnitude faster than BPF aims at in-place filter updating.

The Honey@home [4] architecture relies on communities of regular users and organizations installing a lightweight, traffic redirector that monitors unused IP addresses and ports. Similarly, Collapsar [2] deploys traffic redirectors in multiple network domains and examines the redirected traffic in a centralized farm of honeypots. In both cases, deployed probes require some form of initialization regarding the address space they monitor and are unable to adapt to changes in the network schema.

3 Architecture

The idea behind our system is to enable administrators to deploy a *plug-and-play* network monitoring solution. The network advertises routing prefixes for the subnets that

[1] http://www.quagga.net/

are live, i.e., subnets that contain devices that need to communicate with the Internet. Such advertisement already takes place using the existing infrastructure. With the addition of our monitoring solution, the administrator is required to do nothing more than connect their components to the network. The only case were some minor action is required is if the network advertises the dark subnets as well. While this is not the usual case, if so, the administrator will have to stop it, and advertise only the part of the network actually being used.

Configuration file. This file contains all the information regarding the specific organization where the sensor will be deployed. It contains the IP address range that has been allocated to the organization. Furthermore, information regarding the sensor is contained, so our system may be able to forward all traffic destined to the monitored subnet back to the sensor by dynamically altering the necessary routing tables.

Allocator component. This component is responsible for the management of all the components and dynamically changes the system's behavior based on messages received from the other components. First of all, the Allocator parses the configuration file and splits the organization's IP address range into subnets of the specified size. Based on the available subnets, the core component instructs the Advertiser component to monitor all announcements by the organization's border router. Based on that information the component can infer which subnets are dark and select one for monitoring. Then, the Advertiser is instructed to start advertising the selected subnet, and the Updater to update the routing tables and forward all traffic destined to the subnet back to the sensor. In certain cases, the system will have to dynamically change the monitored subnet. In those cases, the Allocator instructs the remaining components to stop all actions and remove all entries concerning the previous subnet. Based on the announcement history, a new subnet is selected for monitoring and all components are instructed accordingly.

Advertiser component. This component monitors all announcements by the organization's border router and keeps a history of all advertised subnets which it sends to the Allocator component. In order to do so, the administrator must designate the host running this component as a "neighboring" router (or peer) to the border router's configuration. This must be done once during the deployment phase of our system. When the appropriate instruction is received from the Allocator, the Advertiser starts advertising the selected subnet. As the host running this component does not reside at the border of the network, the border router must be configured to accept and propagate advertisements coming from our system. This acts both as a fail-safe and an assurance towards network administrators that they can be aware, control and block network updates pushed by our system. If at any moment the subnet is advertised from the border router itself, meaning that the subnet has been selected by the organization to be used, the Advertiser ceases to advertise the subnet and informs the Allocator.

Updater component. This component is responsible for updating and maintaining the routing table entries concerning the monitored subnets, in the Unix kernel of the host routing the advertised subnets of our system. That host is designated as the responsible router for a given prefix by the Advertiser, during its advertisements. The Allocator instructs the Updater to create new entries that will forward all traffic arriving at the routing host, destined to the selected subnet, back to the monitoring sensor. When the

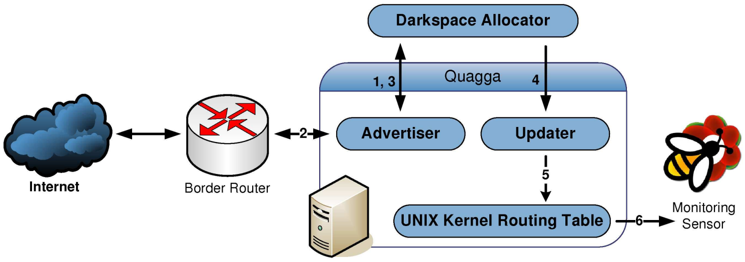

Fig. 1. System Architecture

system dynamically shifts from one subnet to another, the Updater is instructed to clear all entries regarding the previous subnet and add entries for the new subnet.

We can see a depiction of the system's architecture in Figure 1. The Allocator instructs (1) the Advertiser to log all subnet announcements from the border router (2) which are used to update the announcement history (3) and select the subnet that will be monitored. The Updater is then instructed (4) to alter the routing tables and add entries to the routing tables for the selected subnet (5) so the appropriate traffic can be forwarded to a sensor (6), such as Honey@home.

4 Conclusion

We presented the design of a system that enables the dynamic allocation of dark address space for monitoring purposes. Our system aims to facilitate organizations that want to donate IP address space for monitoring purposes, and allows the automatic handling of unadvertised IP address subnets. This is work in progress, and we are currently in the process of implementing a prototype of our system.

References

1. L. Deri. High-speed dynamic packet filtering. *Journal of Network and Systems Management*, 15(3):401–415, 2007.
2. X. Jiang and D. Xu. Collapsar: A VM-Based Architecture for Network Attack Detention Center. In *Proceedings of the* 13^{th} *USENIX Security Sumposium*, 2004.
3. A. Di Pietro, F. Huici, D. Costantini, and S. Niccolini. Decon: Decentralized coordination for large-scale flow monitoring. In *IEEE Conference on Computer Communications (INFOCOM)*, 2010.
4. K. Anagnostakis S. Antonatos and E. P. Markatos. Honey@home: A new approach to large-scale threat monitoring. In *the Proceedings of the 5th ACM Workshop on Recurring Malcode (WORM)*, 2007.
5. P. Trimintzios, M. Polychronakis, A. Papadogiannakis, M. Foukarakis, E. Markatos, and A. Øslebø. DiMAPI: An application programming interface for distributed network monitoring. In *Proceedings of the* 10^{th} *IEEE/IFIP Network Operations and Management Symposium (NOMS)*, 2006.
6. Z. Wu, M. Xie, and H. Wang. Swift: a fast dynamic packet filter. In *NSDI'08: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation.*