ARTICLE IN PRESS

Future Generation Computer Systems ▮ (▮▮▮▮) ▮▮▮–▮▮▮

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

# CoreFlow: Enriching Bro security events using network traffic monitoring data

Ralph Koning [a,b,*], Nick Buraglio [b], Cees de Laat [a,b], Paola Grosso [a]

[a] Universiteit van Amsterdam, Science Park 904, Amsterdam, The Netherlands
[b] Energy Sciences Network, Lawrence Berkeley Lab, Berkeley, CA, USA

## HIGHLIGHTS

- Enriching IDS data with NetFlow information gives a better view of an attack.
- CoreFlow ingests data from the Bro IDS and augments this with flow data from the devices in the network.
- The augmented information can be the starting point for sophisticated countermeasures close to the origin.
- The CoreFlow prototype is tested in the ESnet network, using inputs from 3 Bro systems and more than 50 routers.

## ARTICLE INFO

*Article history:*
Received 20 September 2016
Received in revised form
8 April 2017
Accepted 8 April 2017
Available online xxxx

*Keywords:*
Security
Network
IDS
Netflow
Flow
Detection
IPFIX
DDoS
Carrier networks
Transit networks

## ABSTRACT

Attacks against network infrastructures can be detected by Intrusion Detection Systems (IDS). Still reaction to these events are often limited by the lack of larger contextual information in which they occurred. In this paper we present CoreFlow, a framework for the correlation and enrichment of IDS data with network flow information. CoreFlow ingests data from the Bro IDS and augments this with flow data from the devices in the network. By doing this the network providers are able to reconstruct more precisely the route followed by the malicious flows. This enables them to devise tailored countermeasures, e.g. blocking close to the source of the attack. We tested the initial CoreFlow prototype in the ESnet network, using inputs from 3 Bro systems and more than 50 routers.



## 1. Introduction

As society becomes more reliant on cyber-infrastructures and computer networks, securing this infrastructure becomes increasingly more important. Large scale cyber attacks might be directed toward critical infrastructure components such as the DNS root servers [1]; against commercial network providers such as end-user ISPs [2]; or against educational and research networks serving academia [3]. All these attacks show how fragile computer networks can be.

Given these continuous attacks carefully monitoring Internet systems and components for suspicious activities becomes imperative. There are many developments in monitoring and intrusion detection systems (IDS) that enable them to trigger alerts when such activities are present [4,5]. When such an episode occurs it is the responsibility of the security and incident response teams that monitor this information to further investigate these events; this often requires them to look up and combine information from multiple sources to make a more informed judgment. In this paper we describe CoreFlow, a prototype framework to enrich IDS data with network flow data; this enhancement provides more context to security events and this in turn creates more targeted alerts and more advanced responses. This is in particular important for car-

* Corresponding author at: Universiteit van Amsterdam, Science Park 904, Amsterdam, The Netherlands.
*E-mail addresses:* r.koning@uva.nl (R. Koning), buraglio@es.net (N. Buraglio), delaat@uva.nl (C. de Laat), p.grosso@uva.nl (P. Grosso).

http://dx.doi.org/10.1016/j.future.2017.04.017
0167-739X/

rier networks that due to their characteristics require to correlate information coming from distant elements in the network.

In Section 2 we will briefly review the different challenges carrier networks face to secure their networks, and we introduce ESnet, the network where we tested CoreFlow; in Section 3 we discuss the information sources used in this research. Section 4 and Section 5 describe CoreFlow architecture and implementation. In Section 6 we reflect on the functionality of the framework and discuss what can be improved. Section 7 covers related work and Section 8 contains the conclusion and future work.

## 2. Carrier network security

Carrier networks present different challenges than enterprise or campus networks due to their different characteristics. In Table 1 we list five aspects in which carrier networks differ from enterprise and campus networks when we consider them from a security perspective: external connectivity, application security, restrictions and policies, impact of countermeasures and network capacity. For example, in carrier networks it is unfeasible to run all traffic through a single or a set of security appliance devices due to very high data rates, as well as the large or numerous data flows and multiple ingress and egress points. Additionally, carrier networks are often tasked with adhering to network neutrality laws or policies which prevent filtering or otherwise altering traffic in any way other than to protect the infrastructure of the network.

### 2.1. ESnet

Our CoreFlow development and validation has taken place at ESnet. ESnet is a national research and education network (NREN) that interconnects multiple national labs in the US to each other, to supercomputing facilities, as well as other research networks in the world. Fig. 1 shows the topology of the ESnet backbone network that spans the US and a part of Europe. The backbone consists mainly of 100 Gbps links and allows sites to connect to ESnet at various speeds.

ESnet primarily transits data within the connected institutions and to other connected research facilities and resources and therefore operates as a carrier or transit networks for scientific traffic.

Given their architectures NRENs like ESnet fall in the category of carrier or transit networks and are therefore a suitable testing ground for CoreFlow.

## 3. Information sources

Different information sources can be used to identify and counteract network attacks.

IDS systems are able to perform in depth inspection of packets to detect security problems, yet they only have a limited end perspective of the network. NetFlow and other flow-based tools provide detailed network traffic information. This information can be collected from all routers over the entire extent of the network and can provide a global view and the origin of the traffic that transits a network. Correlating data from both of these information sources may give a more detailed view on the origin of the malicious traffic and thus provide more context to act upon, this detailed multi-source view makes countermeasure less sensitive to spoofed traffic information.

This is particularly useful when an attack is volume based such as in the case of a Distributed Denial of Service (DDoS) attack. In this case instead of blocking traffic at the end systems, it may be preferable to prevent the malicious data from entering the network at the entry point, or contact an upstream provided to block the specific traffic. This reaction at the network edges is complicated by the fact that this attack traffic is often spoofed to cover its origin, causing it to have another entry point into the network than presented. Since the addressing information cannot be relied upon, one has to determine the origin by checking presence of this traffic pattern on all routers on the path.

In our development of CoreFlow we relied specifically on Bro data, on NetFlow information, on Splunk for data aggregation and on Route Explorer for path calculation.

### 3.1. Bro

Bro [6] is an open source network analysis framework developed at the International Computer Science Institute in Berkeley, CA and the National Center for supercomputing Applications in Urbana–Champaign, IL. Bro focuses on network security monitoring and offers functionality beyond traditional intrusion detection systems. It includes an event engine and a policy module in which one can write custom policies. Due to clustering capabilities, Bro can scale to 100 Gbps links [7]. Bro has an extensive policy system that can be used to react on or to trigger events. Events can thus also be correlated within the Bro framework itself as part of a policy. To implement policies Bro uses its own scripting language. This language is limited but it could in principle be used to implement the CoreFlow functionality as a plugin in the C language. This would require knowledge of two languages, the Bro domain specific language and C; for this reason it seemed more practical to us to implement CoreFlow as a stand-alone system using Python.

Building a stand-alone system makes CoreFlow more flexible since we are able to use multiple input sources or replace out Bro in favor of a different IDS. Python is a widely used and easy to learn language which became very popular among data scientists, therefore by using it we try to lower accessibility for potential collaborators that can help to extend CoreFlow with new features. Additionally, Python has a large set of libraries and tools available that are specifically useful for analysis and working with large data sets, these libraries can be used to aid the correlation and enrichment process.

### 3.2. NetFlow and IPFIX

NetFlow, originally developed by Cisco Systems, but now present on most modern routers is a protocol that allows routers and other network devices to export flow information. According to [8], Cisco traditionally distinguishes a flow based on 7 properties, two of which are not required:

- IP source address
- IP destination address
- source port
- destination port
- L3 protocol type
- Class of service (optional)
- Router or switch ingress port (optional).

These properties are extended in subsequent versions such that NetFlow supports IPv6, vlans, and MPLS labels.

IPFIX (IP Flow Information eXport) described in RFC5115 [9] is a protocol developed by IETF that supersedes NetFlow v9. The major tools and collectors used to work with netflow information are adapted to also accept the IPFIX format. In this paper we use the term NetFlow to refer to both the NetFlow and IPFIX protocols. In CoreFlow the data we import from the routers uses the *nfdump*[1] format.

[1] nfdump website: https://github.com/phaag/nfdump.

Please cite this article in press as: R. Koning, et al., CoreFlow: Enriching Bro security events using network traffic monitoring data, Future Generation Computer Systems (2017), http://dx.doi.org/10.1016/j.future.2017.04.017