"LUDO" – KIDS PLAYING DISTRIBUTED DENIAL OF SERVICE

Jessica Steinberger*,†, José Jair Santanna†, Evangelos Spatharas‡, Hendrik Amler*, Niklas Breuer*, Kristian Graul*, Benjamin Kuhnert*, Ulrike Piontek*, Anna Sperotto†, Harald Baier* and Aiko Pras†

* da/sec - Biometrics and Internet Security Research Group, University of Applied Sciences Darmstadt, Darmstadt, Germany

† Design and Analysis of Communication Systems (DACS) University of Twente, Enschede, The Netherlands

‡ GÉANT Ltd, Cambridge, UK

Research paper

Abstract

Distributed denial of service attacks pose a serious threat to the availability of the network infrastructures and services. GÉANT, the pan-European network with terabit capacities witnesses close to hundreds of DDoS attacks on a daily basis. The reason is that DDoS attacks are getting larger, more sophisticated and frequent. At the same time, it has never been easier to execute DDoS attacks, e.g., Booter services offer paying customers without any technical knowledge the possibility to perform DDoS attacks as a service. Given the increasing size, frequency and complexity of DDoS attacks, there is a need to perform a collaborative mitigation. Therefore, we developed (i) a DDoSDB to share real attack data and allow collaborators to query, compare, and download attacks, (ii) the Security attack experimentation framework to test mitigation and response capabilities and (iii) a collaborative mitigation and response process among trusted partners to disseminate security event information. In addition to these developments, we present and would like to discuss our latest research results with experienced networking operators and bridging the gap between academic research and operational business.

Keywords

Distributed Denial of Service as a Service, Security attack experimentation framework, Mitigation and response, Collaboration Defense, Testing mitigation and response, Firewall On Demand, NSHaRP, Netflow, BGP Flowspec

1. Introduction

Over recent years, Distributed Denial of Service (DDoS) attacks remain the number one operational threat responsible for network infrastructure and service outages (Arbor Networks, 2016). The reason is that DDoS attacks are getting larger, more sophisticated (e.g. multi-vector attacks) and frequent. At the same time, it has never been easier to execute DDoS attacks, e.g., Booter services offer paying customers without any technical knowledge the possibility to perform DDoS attacks as a service (Santanna, et al., 2015a).

Nowadays, mitigation of DDoS is performed using traditional security solutions such as firewall and Intrusion Prevention System (IPS) devices (GÉANT, 2015). These solutions are often not able to handle the large amount of traffic reaching the target network. One reason is that the monitoring equipment located at the victim side might exhaust its own resources as a side-effect of the attack (Sadre, et al., 2012). Further, if a DDoS attack already reached the perimeter of the target network, it is often too late to start mitigation procedures, because the network link is already saturated. To test and optimize mitigation and response capabilities within the network, we present our Security aTtack experimentatiOn fRaMework (STORM). STORM is used to test different types of DDoS attacks and to analyse traffic patterns. Further, STORM is used to analyse the impact on the configured network topology.

Given the increasing size, frequency and complexity of DDoS attacks, there is a need to perform a collaborative mitigation. In addition, exchanging threat information among trusted partners is used to reduce the time needed to detect and respond to large-scale network-based attacks. However, exchanging threat information is currently done on an ad-hoc basis via email or telephone, and there is still no interoperable standard to exchange threat information

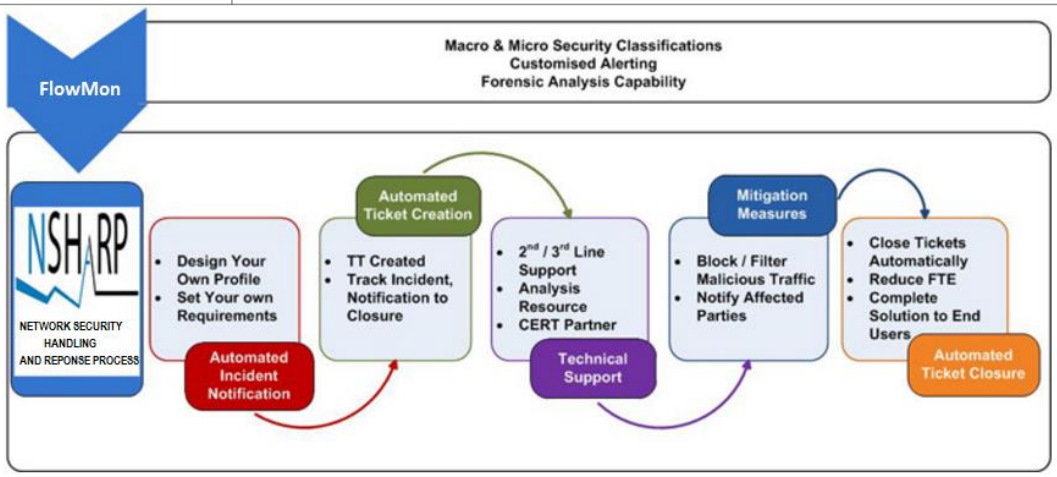


Figure 1: NSHaRP process

among trusted partners (Steinberger, et al., 2015b). To support the dissemination of threat information, we present a communication process that helps organizations to speed up their mitigation and response capabilities without the need to modify the current network infrastructure, and hence make it viable to use for network operators.

The remainder of this paper is organized as follows: Section 2 describes the current attack situation of DDoS at the upstream National Research and Education Network (NREN) provider GÉANT and how GÉANT currently handles, mitigates and resolves DDoS attacks. Section 3 presents the phenomenon of Distributed Denial of Service attacks as a Service and introduces the DDoSDB - an infrastructure to share real attack data and allow collaborators to query, compare, and download attacks. STORM is used to test mitigation and response capabilities of the own network and is presented in Section 4. In Section 5, we introduce a communication process to disseminate security event information among trusted partners, different event producers and consumers, and describe security concerns. Finally, we conclude the paper and give an outlook on future work in Section 6.

2. DDoS at GÉANT

GÉANT, the pan-European network serving 50 million users in almost 10,000 institutions, is a network with terabit capacities which witnesses close to hundreds of DDoS attacks on a daily basis. The question is, how to deal with DDoS attacks, how to detect and stop them and how to learn from them. One thing is certain, and that is that there is no single entity or ISP that would have said "no" to a second hand during a decent sized DDoS. GÉANT takes a leap forward and admits that too, it cannot battle with giant zombie botnets. In that sense, GÉANT "cheats" and seeks for allies in the NREN (and not only) community to battle against DDoS, but this has its own challenges as well. To keep up with its demanding users, network and its systems has built the next generation mitigation into the core.
Today's measures and controls must meet and surpass tomorrow's attacks. As such, we demonstrate how GÉANT has become a leader in how to protect and facilitate research and education in today's world. This Section touches on the technologies and approaches used to protect research in the future when it comes to DDoS attacks.

2.1 A Defense in Depth Approach to Security

Security is at the forefront of all activities on the Internet. Anybody uses the Internet today for most if not all activities and so do researchers aiming to foster collaboration opportunities and to extend research partnerships beyond GÉANT borders. In thinking of the latter, GÉANT has a remit to its community and is responsible to ensure that all research is conducted in a safe, secure and "available" manner, free from disruptions such as denial of service attacks.
To ensure that the network is secure, and always available a defense in depth approach is followed. This enables the use of multiple technologies to protect critical infrastructure and the traffic that transits the network every day. This approach to an extent future proofs the security of the network as it allows for the implementation of diverse and independent technologies that can meet tomorrow's security needs and threats.

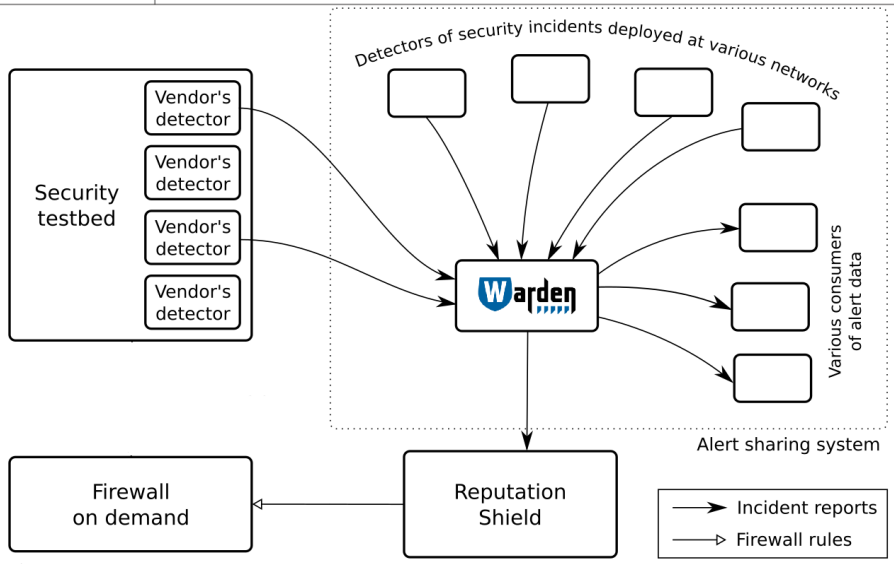


Figure 2: A fast notification system architecture

2.1.1 A Layered Approach

GÉANT deploys novel tools for anomaly detection and reporting. In this area GÉANT provides a mechanism to achieve this secure and safe network. The first part of all security measures is the monitoring of resources, ensuring that carried traffic to free from malicious intent.

The established and effective service in Network Security Handling and Response Process (NSHaRP), is GÉANT's security incident notification service alerting against various kind of attacks as they are identified by NetFlow tools that monitor network traffic. The NSHaRP process is shown in Figure 1. NSHaRP detects denial of service attacks (as well as many others), which prevent researchers from accessing valuable services, raises an automatic ticket with the GÉANT Operations Centre detailing the attack, and also notifies the affected NREN with any relevant information and ticket number. Affected parties can then reply to the e-mail thread asking for attack mitigation or close monitoring of the event.

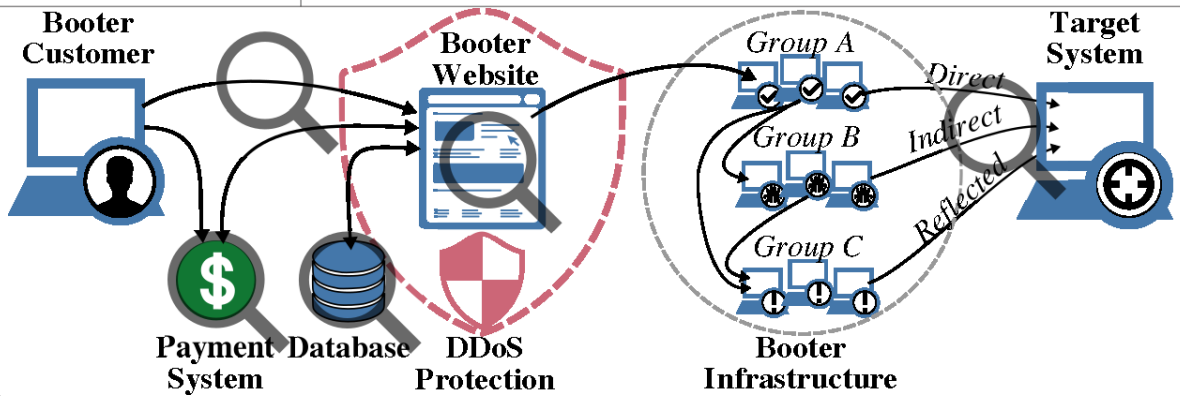
2.1.2 Tomorrow's Technologies

Detection is only one part of the defense mechanisms in place to detect malicious events; intelligence and mitigation strategies play a key role in stopping events from occurring before they start. The GÉANT CERT security team have formed ties with security experts in the GÉANT NRENs community and in the commercial world to ensure that threats to the integrity of the research networks are managed by sharing zero day threats and then ensuring that the bad actors are blocked at the borders of GÉANT further reducing the risks to GÉANT NRENs.

GÉANT uses FlowMon, built to detect and report on malicious traffic on the network, by incorporating the alerting mechanism into the GÉANT Trouble Ticketing System, providing a near real-time detection and reporting mechanism adding value to NREN CERTs. This novel approach extends NRENs detection and mitigation capability to the GÉANT borders and is innovative and unique by catering for different types of user requirements dynamically.

Defending against the several hundred of attacks a day, the network requires an even more state of the art solution to block and/or filter malicious traffic. Firewall on Demand is tomorrow's solution for firewall filters on large and complex networks. GÉANT is incorporating this into the NSHaRP service to enable NREN teams to filter traffic well beyond their borders adding immense value to defending against the Internet's malicious traffic. Using BGP Flowspec, we can dynamically and within seconds deploy complex rules that block malicious traffic whilst allowing legitimate traffic.

Detecting an attack and mitigating it is only good once. If one could learn from an attack it would be greatly more beneficial and also re-usable for more than one times. This approach would focus on the source of the attack and attach an "attack" score against it. Scores associated with a list of IP addresses could be stored on a database available to subscribers and systems to query them in order to make firewalling and other decisions. Figure 2Figure



1

Figure 3: Booters' scenario elements and the five lines of investigation.

shows a high level design of how the "learn and proactively mitigate" approach looks like using the Warden system which processes alerts coming from various systems and calculates and assigns scores for abusers. Reputation Shield in turn is the one that can be queried from other systems to obtain scoring calculations. The plan here is to have Reputation Shield working closely with Firewall on Demand to create firewall rules for highly scored abusing sources, in a proactive manner that would greatly reduce the mitigation time.

3. DDoS as a Service

In general, DDoS attacks involves a sophisticated orchestration of third party compromised machines that, under the control of an attacker, generate harmful traffic against a target system. While specialized hacker skills have traditionally been required to perform DDoS attacks, it has recently become possible even for inexperienced Internet users to carry out such threat. Today, services offered by websites referred to as Booters can be easily bought to launch DDoS attacks against anyone in the Internet. In this Section, we present our investigations on Booters, introduce initiatives and describe the elements that compose the Booters' operational scenario.

3.1 Booters' Operational Scenario

Booters are also known as Stressers, DDoS-for-hire, DDoS as a service, and DDoSers; and can be easily found via a Google query. After choosing a Booter website, the elements of their operational scenario interact among them as shown in Figure 3. First, a *Booter Customer* accesses a *Booter Website* and choose one attack plan among dozens offered. Each attack plan is defined as the price that must to be paid allowing customers to perform a predefined number of attacks, with a maximum attack duration each, within a maximum period of time (expiration time). After the customer chose a plan, the Booter website forwards the request to a *Payment System* (e.g., PayPal, BitCoin and credit card), which notifies the Booter when the amount of money is paid by the customer. Such notification unblocks the customer to perform as many attacks as he/she wants respecting the characteristics of the attack plan.

When a customer decides to attack a *Target System*, the Booter website contacts its *Infrastructure* that de facto performs the attack. The infrastructure can consist of three different groups of systems. *Group A* is the set of machines contacted directly by the Booter website, usually composed of a private infrastructure (e.g., set of virtual private servers). *Group B* is composed of compromised machines that are controlled by the Group A to perform attacks. Finally, *Group C* is a set of misused machines, usually composed of online servers that act amplifying and reflecting spoofed requests (generated by either group A or B) towards the *Target System*.

The two last elements of the Booters' scenario are the *database* and the *DDoS protection* service. The *database* stores information about the customer, for example their email account, the IP address, the payment record, the type of attack chosen, and the attack target. The *DDoS protection* service is common in the scenario because Booter owners subscribe to third party companies for protection services. In the next subsection, we describe the parts of the Booters' scenario that are already investigated.

3.2 Lines of Investigation

Booters were reported for the first time by (Prolexic, 2012). Over the years, hundreds of reports and blog posts were written by network security companies and specialist about the topic. Among these investigators, we highlight Brian Krebs, a security specialist whose website has been attacked for consecutive days and who wrote many blog

posts about Booters[1]. In these posts, Krebs describes for example how Booters operate, what are the people behind Booters and the characteristics of attacks against high profile companies (like Microsoft and Sony) involving Booters.

While generally hundreds of people are interested in Booters, currently there are only three main research groups investigating the phenomenon of DDoS-as-a-service as listed in Table 1. The lines of investigation are represented in greater detail in Figure 3 and described below:

Table 1: Lines of Investigation on Booters

Reference	Blacklist	Payment	Database	Dashboard	Attacks
(Karami, et al., 2016)		✓	✓	✓	✓
(Bukac, et al., 2015)			✓	✓	✓
(Santanna, et al., 2015a)					✓
(Santanna, et al., 2015b)			✓		
(de Vries, 2015)	✓				
(Chromik, et al., 2015)	✓	✓			
(Santanna & Sperotto, 2014)	✓				
(Karami & McCoy, 2013a)			✓		✓
(Karami & McCoy, 2013b)			✓		✓

- **Blacklist**: is the type of investigation that collects data. In particular, which are the Booters, how many they are, and what are the information available to the broad audience (i.e., users that find Booters using popular search engines, for example Google). This type of investigations results in an extensive list of Booters to be used as black or grey list;
- **Payment**: is an investigation that analyses Booter owners within payment systems. This type of investigation can result in operations to shutdown Booter owner accounts and reduce the work together with a Payment System (e.g., PayPal);
- **DataBase**: this investigation targets to analyse leaked Booter databases. This investigation is used to highlight the behaviour and identity of Booters customers, which can then be prosecuted;
- **Dashboard**: is the type of investigation on which everything is displayed at the Booter website. In particular, the number of current attacks, the price plans, and the advertised characteristics of attacks. The use of this investigation is to get an overview of what is offered and potentially compare with what is delivered by Booters (in the attack line investigation);
- **Attacks**: is a line of investigation that measures attacks launched by Booters and reveal their characteristics. These characteristics of attacks can be used to mitigate the attacks and correlate generic attacks with Booter attacks;

Besides each element already addressed by the initiatives (Table 1), we still see potential for research in the connection between the Booter website and its infrastructure that performs attacks. We also see some open questions on the fingerprinting of Booters attacks. Both topics were never addressed and can offer provide valuable information to mitigate Booter attacks.

4. Security attack experimentation framework

In this section, we introduce the Security attack experimentation framework (STORM). First, we present the main motivation and objectives of the development of STORM, and the use-case scenarios. Second, we describe the main components of STORM and how these components interact with each other. We also provide insight into the development of STORM, the tools that have been used and how they are integrated.

4.1 Introduction

Due to the increasing use of computer networks, distributed attacks are one of the major threats to network infrastructures. Attackers are challenging the reliability of the Internet by misusing fundamental network technologies to issue catastrophic events that are exhausting bandwidth and resources. Since networks have become an important part of everyone's life, it is necessary to defend the systems against such threats. Yet, DDoS is still one of the most frequent attacks and studying defense mechanisms becomes more and more important.

[1] http://krebsonsecurity.com/category/ddos-for-hire/

DDoS attacks are an effective method of consuming resources on a target network in comparison to non-distributed approaches. They are coordinated by an attacker who uses compromised hosts that are often organized in huge botnets.

System operators of centralized services are challenged to verify the effectiveness of mitigation procedures within their network. Currently, several tools like hping[2] that can be used to initiate a controlled attack on a target network exist, however these tools do not support any kind of measurement or centralized management capabilities. Moreover, solutions for scanning a network for vulnerabilities are also available, but do not support the verification of mitigation procedures of distributed attacks.

Running reproducible attack experiments in separated networks can be a difficult task, since each network node needs a control plane throughout the time span of the experiment. The use of multiple third-party applications to perform various tasks (e.g., create, measure, analyse and report the attack traffic) often do not use a common set of commands. Further, they make use of shells scripts, python or native applications that are not part of the operation system (OS). With an increasing amount of nodes in the target network, configuring and running experiments manually becomes more complex. As a result, a semi-automated and centralized approach is needed.

To overcome the missing capability to perform reproducible attacks, we introduce our Security aTtack experimentatiOn fRaMework (STORM). STORM is used to test and analyse different types of DDoS attacks in a controlled laboratory environment. In addition, STORM contributes to traffic engineering, because network operators testing and analysing deployed mitigation capabilities might saturate link capacities and thus are able to estimate the effects of future attacks and prepare appropriate countermeasures. STORM performs controlled network-based attacks on customized network topologies. We collect data of the performed attacks to gain insight into the functionality of these threats, how to mitigate and resolve them, and verify mitigation procedures. STORM constitutes a viable and reproducible approach to test network defense solutions by using a secure and controlled environment.

4.2 Use-case scenarios of STORM

STORM is a framework that focuses on network testing and analysis. One major use-case is the verification of mitigation strategies to answer the following questions:

- Does a mitigation strategy work against specific threats?
- How does the strategy perform?
- After a hardware or software update, does every device work like expected?

In order to answer these questions, STORM allows to run reproducible experiments. The capability to run reproducible experiments facilitates the comparison of experiment data. This comparison provides an insight into the network performance under different conditions.

4.3 Components of the STORM framework

STORM is a framework that consists of three components: (i) a graphical user interface (GUI), (ii) a STORM client and a (iii) STORM server. The management of STORM is performed using either a web-based or a console-based GUI. Both, the web-based and the console-based GUI offer the possibility to configure a network of "attackers" and launch different types of DDoS attacks (e.g., volumetric, TCP-State-Exhaustion, application-layer). The STORM server controls all STORM components and initiates the actions that are performed via the STORM clients. All STORM components communicate over a network connection and are designed to run independently on different machines. The interaction between the STORM components is illustrated in Figure 4.

4.3.1 Web and console client

The management of STORM is performed using either a web-based or a console-based GUI. Both GUIs are used to perform user interactions with the STORM server. The web client is used as the default GUI and supports both, desktop and mobile platforms. Since standardized protocols like HTML5 and JavaScript are used, STORM ensures platform independency. The console-based client is designed for administrators that need to access the STORM server via remote connection or for automation in scripts.

[2] http://hping.org/

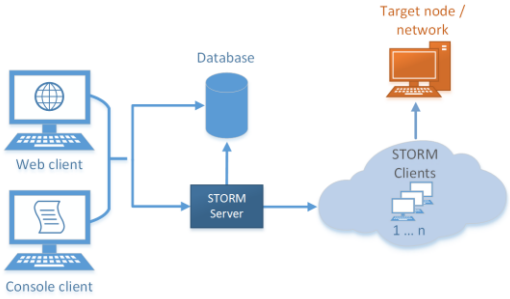


Figure 4: Interactions of all STORM components

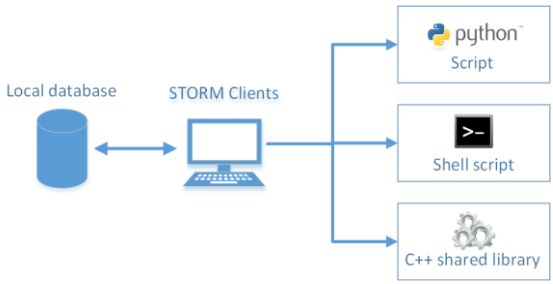


Figure 5: Overview of STORM client

4.3.2 STORM client

The STORM client runs commands send by the STORM server and has to be deployed on every node that wants to join the STORM system. A STORM client can be deployed on a dedicated machine or on the same machine that runs the STORM server. Once started, the STORM client connects to the configured STORM server instance. When the STORM server sends a command towards a STORM client, the STORM client executes the received command and if desired, reports back to the STORM server.

The core STORM client does not contain any attack or measurement functions. Instead a command or task is contained in a external file, that handles command execution and information collectivity. These external files are recognized by the STORM client at runtime and register external modules to the STORM server. Currently, the STORM framework supports Python, Shell script and native C++ for extension as shown in Figure 5. The capability to recognize other file formats than the current external file formats is given by adding customized modules by the user.

A STORM client uses a local database to store collected information. The use of a database can be scaled up to use a full-fledged database server or even operate on system memory only. Storing the measurements to a local client-side database is necessary for several reasons.

The STORM clients can be deployed on dedicated machines to create a higher amount of attack traffic to effectively saturate the available bandwidth. Further, storing the measurement data to the local client-side database allows the STORM server to pull all information after running an experiment. As a result, it is possible to review and analyse the impact of the attack and the network condition in time intervals. Moreover, the measurements of the attack traffic and the mitigation and respond capabilities do not get biased by command traffic between the STORM server and client.

The STORM client was deployed in two different versions: a virtual machine image and as an installable software package. The virtual machine image consists of the STORM client and the virtual device drivers to minimize any possible overhead.

4.3.3 STORM server

The central management interface of the STORM system is the STORM server. All STORM clients connect to a server instance to join a STORM system. The local client-side database is used to store all measured data from the STORM clients. Although there is support for a real-time information gathering, the database is used for post-experiment analysis. That enables the user not only to generate reports from experiment data, but also enables a playback functionality that shows the progression of the experiment. All information can be exported for exchanging data with third party tools.

4.4 STORM configuration examples

As STORM is a framework, there is no default model of a STORM system. Hence, a network operator can build a customized STORM system that suits all his requirements. In this Section, we present two examples of the STORM system deployments: (i) single machine deployment and (ii) and multiple machine deployment.

4.4.1 Single machine deployment

The STORM system can be configured as a single machine solution as shown in Figure 6: Single machine deploymentFigure 6. This STORM server, STORM clients, as well as the web-based and the console-based GUI are deployed on the same physical machine.

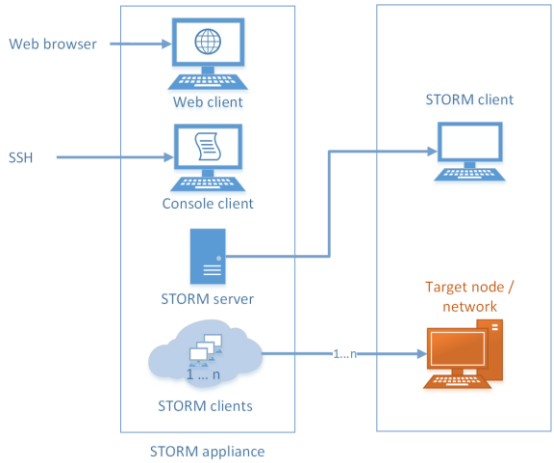


Figure 6: Single machine deployment

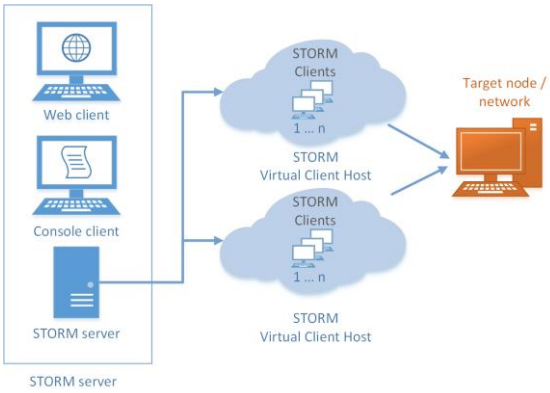


Figure 7: Single machine deployment

For measurements on the target, a STORM client is deployed. All the components run in a virtualized environment. An operator connects to the STORM server via web browser or remote ssh connection. The limitation of this configuration is the maximum bandwidth available at the host machine as well as the resource limitations of the virtualized environment

4.4.2 Multiple machine deployment
Besides the single machine deployment, STORM also supports the use and distribution of multiple STORM clients as shown in Figure 7. The STORM clients can be deployed on dedicated machines and create attack traffic that consumes the amount of available bandwidth. The number of STORM clients that could be used in an experiment is configured by the STORM server.

5. Collaborative DDoS Defense: A communication process

In this Section, we introduce a communication process that facilitates the exchange of threat information among trusted partners and thus supports a collaborative DDoS defense. Further, we describe the main components of our proposed communication process and how these components interact with each other.

5.1 Introduction

Sharing security events is deemed of critical importance to counteract large-scale network-based attacks (e.g., DDoS, DrDoS) at Internet service provider (ISP) networks as these attacks have become larger, more sophisticated and frequent. On the one hand, security event sharing is regarded to speed up organization's mitigation and response capabilities. On the other hand, it is currently done on an ad-hoc basis via email, member calls or in personal meetings only under the premise that participating partners are personally known to each other. As a consequence, mitigation and response actions are delayed and thus security events are not processed in time.
One approach to reduce this delay and the time for manual processing is to disseminate security events among trusted partners. To facilitate the exchange of security events in conjunction with widely adopted monitoring technologies, in particular network flows, we make use of the exchange format FLEX.

To overcome delayed response actions and manual processing of security events, we present a communication process that supports the dissemination of threat information based on FLEX in context of ISPs. We show that this communication process helps organizations to speed up their mitigation and response capabilities without the need to modify the current network infrastructure, and hence make it viable to use for network operators.

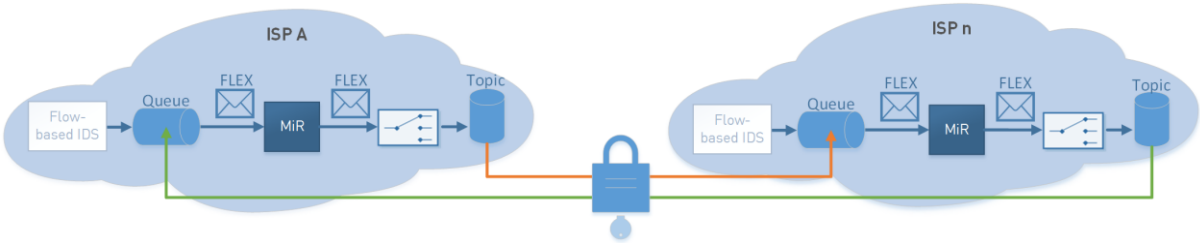


Figure 8: Components of the mitigation and response system

5.2 Components of the communication process

Our communication process consists of gateways that are passed by every single security event message. A security event message is transferred among trusted partners using the Simple (or Stream) Text Oriented Messaging Protocol (STOMP) and the data representation uses the Flow-based Event eXchange format (FLEX) (Steinberger, et al., 2015a). Each collaborating partner contains a mitigation and response system, called MiR. MiR is aligned to the Event Processing Technical Society Reference Architecture (Paschke & Vincent, 2009). The components of the communication process are illustrated in Figure 8.

5.2.1 STOMP

The Simple (or Streaming) Text Orientated Messaging Protocol (STOMP) is a text-based protocol that provides messaging interoperability among many languages, platforms and brokers. Therefore, STOMP is language-agnostic and only uses a SEND semantic with a destination string as it does not provide its own queues or topics. STOMP supports messaging features, such as authentication, messaging models (point to point and publish and subscribe), message acknowledgment, transactions, message headers and properties.

The communication process makes use of STOMP between the gateways of the ISPs and is used to transfer FLEX messages among trusted partners. At each destination STOMP is connected to a Java Messaging Service (JMS) queue or topic.

5.2.2 FLEX

The Flow-based Event eXchange Format (FLEX) (Steinberger, et al., 2015a) is used to share security information among trusted partners based on flow data. FLEX is based on the x-arf specification draft v0.2 *X-XARF:SECURE* and uses a generic template system that is described by an abstract syntax denoted using the language of Abstract Syntax Notation (ASN.1). In addition, FLEX makes use of both, signature and encryption methods to prevent unauthorized access to the security event message at the application layer.

5.2.3 MiR Instances

Each collaborating ISP network contains a mitigation and response system, called MiR. MiR is aligned to the Event Processing Technical Society (EPTS) Reference Architecture (Paschke & Vincent, 2009) and connected to both, a queue and several topics of the JMS. MiR performs pattern matching algorithms to aggregate and consolidate security events into a smaller number of events and thus derives complex events. In addition, MiR enriches the security events through new knowledge gained through previous events or data (e.g., proposes remediations, external publicly available data sources). Besides the queue and the topic, MiR is connected to a database that stores previous security events and their remediation for a defined range of time.

5.3 Data flow of the communication process

Our communication process consists of interconnected instances located in different ISP networks forming an overlay network. Each ISP possesses a list of directly connected collaborating ISP networks to prevent a full mesh within the network and thus ensure scalability. The data flow of the communication process is shown in Figure 9. The white and green rectangle represent the data entering the communication process (white=internal, green=external), whereas the gray rectangle represents a terminator symbol. The blue rectangle represents a sub-process within MiR. The diamond is used to visualize a decision or branching point and the connected lines represent different options.

At a time t the detection engine of an ISP identifies malicious activities and raises a security event of the FLEX message type *Event*. The security event is sent to the JMS queue of MiR via STOMP. The security event remains in the JMS queue until MiR consumes them. Next, the security event is searched within the database of previous events.

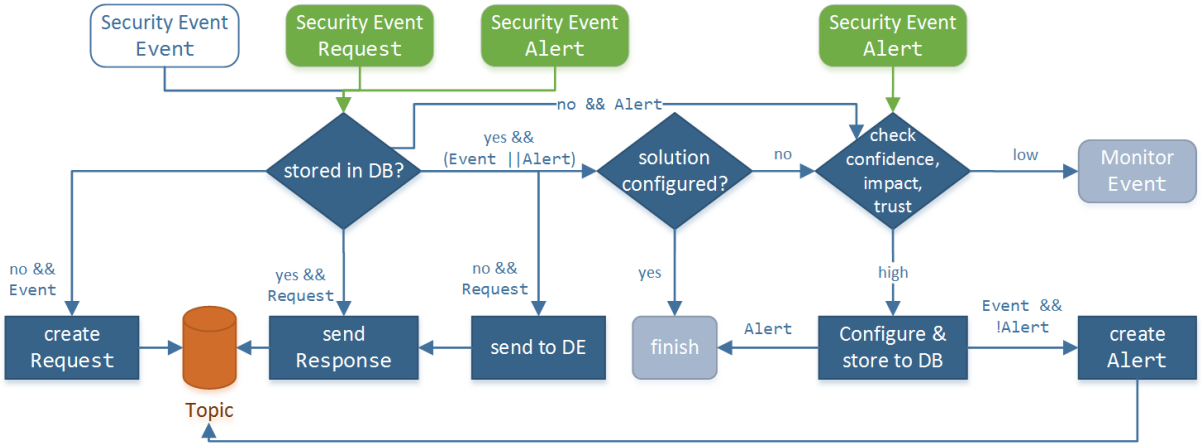


Figure 9: Data flow of the communication process within MiR from the perspective of one ISP

In case the security event could not be found within the database of the previous events, the MiR system initiates a detection process at collaborating neighbour ISP networks to reduce the amount of false positives of the security events. The detection process at the collaborating neighbour ISP networks is initiated by creating a FLEX message of the type *Request* and publishing it to the JMS topic of the adjacent ISP networks. The detection engine of the adjacent ISP network receives the FLEX message of the type *Request*, analyses the network traffic and tries to identify similar behaviour as described within the security event. The result of the analysis is sent back to the requesting network as a FLEX message of the type *Response*.

In case the security event could be found within the database of the previous events, the MiR system examines whether the proposed remediation has been configured. In case the proposed remediation has not been configured yet, the ISP evaluates the feedback received in response to the FLEX message *Request* of the connected collaborating ISPs. In case the majority of the connected collaborating ISPs had seen similar behaviour, the confidence ranking of the security event is increased, the proposed remediation is configured and stored to the database. Subsequently, the information within the security event is preprocessed and restricted to different level of details depending on the level of trust. Finally, the tailored security events are routed based on their content to the appropriate JMS topic as a FLEX message of type *Alert* and thus send out to the connected collaborating ISPs.

5.4 Evaluation methodology

We performed a quantitative evaluation of our communication process using DeterLab. DeterLab (Mirkovic & Benzel, 2012) is an infrastructure designed for experimentation in context of cyber-security. DeterLab is funded by the Department of Homeland Security, the National Science Foundation, and the Department of Defense, hosted by USC/ISI and UC Berkeley, and based on Emulab.

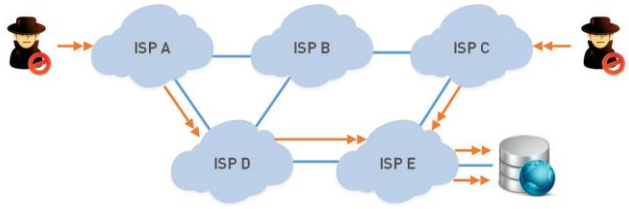


Figure 10: A semantic representation of the DeterLab testbed

The experiment is composed of physical machines for a limited time. DeterLab is used because it is a controlled environment in which it is possible to safely test security threats and defense measures. Our communication process consists of 5 nodes representing Internet Service Providers (ISP A … E) connected through a 500Mb link. In each ISP network, MiR is installed as shown in Figure 8. Figure 10 shows a schematic representation of the DeterLab testbed.

The objective of the experiments is to show that an attack target network with constrained resources benefits from collaborating partners during an ongoing network-based attack, because the target network has no possibility to react itself due to resource saturation. Further, we show that the network-based attack is not propagated further and that our communication process supports the automatic dissemination of threat information and thus speeds up the mitigation and response capabilities of ISP networks. In addition, we show that our communication process is lightweight in numbers of exchanged messages and fast.

To simulate a network-based attack, we performed a distributed TCP SYN flood attack from the ISP networks A and C to consume resources on the web server within ISP network E and render it unresponsive as shown in Figure 10. In our test scenario, ISP E is not able to effectively block the malicious traffic itself and requires collaborating partners in the stream of traffic to mitigate and respond to the TCP SYN flood attack. To create the TCP SYN flood attack, we used empty TCP packets with a TCP packet size of 40 bytes and a TCP FLAGS value of 0x02. To ensure that our network-based attack fully utilizes the requested 500Mb link, we sent 40 000 000 TCP SYN packets in total to ensure an attack duration of 26 seconds. However, the internal function of DeterLab does not always allocate resources as requested and thus we received a link connection with 412Mb and were able to perform a TCP SYN flood attack with a duration of 42 seconds.

In our test scenario, the detection engine of the ISP network A initially identifies the malicious network traffic, creates a FLEX message of the type *Event* containing Cisco Netflow version 5 and starts its communication process. The MiR system of the communication process, located in each ISP network, is able to automatically deploy response actions. In our test scenarios, we make use of automatic notifications via email messages including remediation suggestions that make use of *iptables*. In the initial state of the testbed networks no filtering rules are inserted to show the effects of our communication process. During the experiment within DeterLab suspicious IP addresses are identified and inserted into the packet filter ruleset to block the network traffic.

The MiR diagram of the ISP network A in shows four different types of incoming messages. First, ISP A receives two event messages at the same time containing threat information about the network-based attack from ISP network A and C targeting the web server of ISP E. Second, the adjacent collaborating networks report if they had seen similar behaviour in their network. Next, ISP A adds two blocking rules to the packet filter ruleset to hit its outgoing links and starts a chain of information that is passed along to ISP C. Therefore, ISP A creates an alert and informs the adjacent collaborating networks that deploy the including remediation suggestions. The alerts shown in the MiR diagram of the ISP network A in Figure 11 only represent the incoming alerts and not those that are outgoing. Further, the MiR diagram of the ISP network A in Figure 11 shows that the attack traffic (red line) is sent over 42 seconds. Immediately, after ISP A inserted the blocking rules the effects of the attack traffic are mitigated and the malicious traffic is not propagated further though the network of ISP A (black line). As a result, the traffic at ISP D and E dropped as shown in Figure 11. Next, ISP C deploys the remediation suggestion out of the alert message and as a consequence the traffic at ISP E drops. The MiR diagram of the ISP network E in Figure 11 shows that the collaborating partners in the stream of traffic effectively mitigate and respond to the ongoing network-based attack and thus the network of ISP E is benefiting from the collaboration and the web server recovers.

5.5 Evaluation results

The primary focus to mitigate and respond to network-based attacks is maintaining the availability of the organization's network infrastructure and services. The Message Flow diagram in Figure 11 shows that the overall duration until all ISP networks have a common knowledge took 6 seconds. Further, 18 messages have been sent until all participating ISP networks had a common knowledge. Even though network B has not been actively involved in the network-based attack, ISP network B has been notified by the ISP network A and thus supports a proactive and collaborative mitigation and response approach. Figure 11 shows that a collaboration among trusted partners facilitates a proactive network-based attack mitigation and response approach and thus contributes to ensure availability of the organization's network infrastructure.

Through the increasing amount of security events per day, there is a need to automatically process security events and thus lessen the time to mitigate and respond to ongoing network-based attacks. In addition, the automated dissemination of security events among collaborating partners facilitates a proactive network-based attack mitigation and response approach. In our experiment, we have shown that the dissemination of threat information including remediation suggestions exchanged with FLEX are automatically processable and deployable.

6. Conclusion

Security is a mouse-cat game being played on a non-stop basis. GÉANT, the pan-European research and education network that interconnects Europe's NRENs has a sit at the centre of the security arena. By using a blend of diverse technologies and state-of-the-art approaches, GÉANT ensures security as a key aspect on every activity on a day to day basis, enabling users to span activities and services almost without restrictions. When it comes to security, GÉANT uses novel approaches for firewall filtering on the backbone, anomaly detection for new threats and unique configurations and mechanisms in monitoring a network with hundreds of access points.

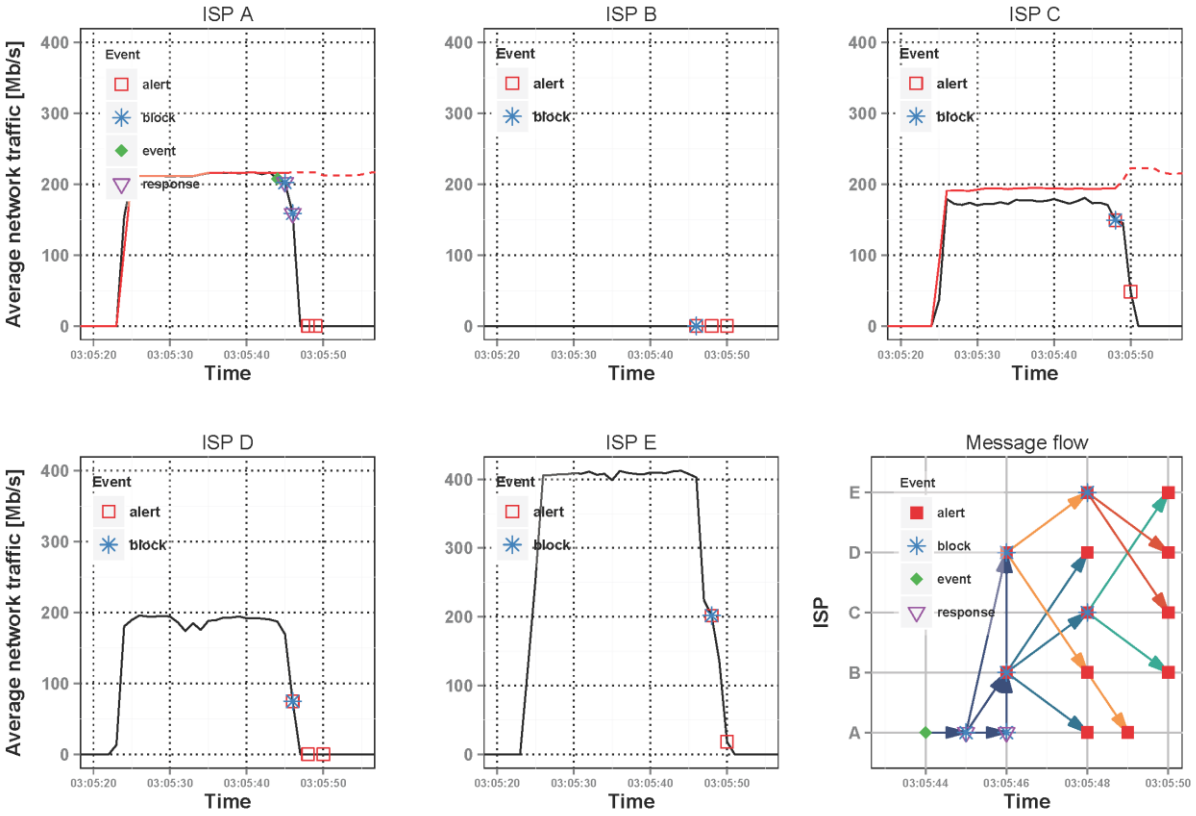


Figure 11: Simulation results of the attack traffic (red line) and cumulative traffic (black line)

In this paper, we presented how GÉANT envisage security and described the use of new technologies through to reality against DDoS. Further, we describe the added value of the new technologies to customers and owners and how those technologies ensure collaboration in free and open means facilitating tomorrow's discoveries.

We introduced the phenomenon of DDoS attacks as a Service and presented the DDoSDB - an infrastructure to share real attack data and allow collaborators to query, compare, and download attacks. Further, we presented STORM, which is an open and extensible framework and is used to test mitigation and response capabilities of the own network. Finally, we introduce a communication process to disseminate security event information among trusted partners, different event producers and consumers, and describe security concerns.

The development of the DDoSDB, STORM and the communication process is intended to be used by network operators that cooperate among trusted partners to minimize or prevent damages caused by DDoS attacks and use an automated threat information exchange. Moreover, the development of DDoSDB, STORM and the communication process ensures a collaborative mitigation and response approach that moves from a reactive to a proactive approach. As a result, collaborating network operators achieve insight into the current threat landscape that otherwise would not be obvious, enhance security expertise, speed up the mitigation and response capabilities and thus lessens the time to understand the threat for each collaborating partner. All our developments easily integrate with the existing infrastructure and are easy to deploy. Therefore, our developments constitute a viable and collaborative approach to query, compare and download attack, test and analyse mitigation and response capabilities of the own network infrastructure, and disseminate security events among trusted ISP networks using the communication process.

Acknowledgments

The work has been funded by CASED and by EU FP7 Flamingo (ICT-318488).

References

Arbor Networks, 2016. *Worldwide Infrastructure Security Report - Volume XI.* [Online] Available at: https://www.arbornetworks.com/images/documents/WISR2016_EN_Web.pdf ,[Accessed April 2016].

Bukac, V. et al., 2015. Service in Denial – Clouds Going with the Winds. *Network and System Security*, November, pp. 130-143.

Chromik, J. J., Santanna, J. J., Sperotto, A. & Pras, A., 2015. Booter websites characterization: towards a list of threats. *In Proceedings of the 33rd Brazilian Symposium on Computer Networks and Distributed Systems*, May, pp. 445-458.

de Vries, J., 2015. *The Generation of Booter (black)lists*, Enschede: Universiteit Twente.

GÉANT, 2015. *DDoS Mitigation in the NREN Environment Workshop.* [Online] Available at: https://wiki.geant.org/display/SIGISM/DDoS+Mitigation+Workshop+Agenda [Accessed April 2016].

Karami , M. & McCoy, D., 2013a. Rent to Pwn: Analyzing Commodity Booter DDoS Services. *;login: The USENIX magazine*, December.

Karami, M. & McCoy, D., 2013b. Understanding the Emerging Threat of DDoS-as-a-Service. *Presented as part of the 6th USENIX Workshop on Large-Scale Exploits and Emergent Threats*.

Karami, M., Park, Y. & McCoy, D., 2016. Stress Testing the Booters: Understanding and Undermining the Business of DDoS Services. *Proceedings of the 25th International Conference on World Wide Web*, pp. 1033-1043.

Mirkovic, J. & Benzel, T., 2012. Teaching Cybersecurity with DeterLab. *IEEE Security & Privacy*, January.

Paschke, A. & Vincent, P., 2009. A Reference Architecture for Event Processing. *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems*.

Prolexic, 2012. *Prolexic Threat Advisory - Threat: DDoS Booter Shell Scripts.* [Online] Available at: http://www.prolexic.com/kcresources/prolexic-threat-advisories/prolexic-threat-advisory-ddos_Booter-scripts_041912/Prolexic_Threat_Advisory_DDoS_Booter_Scripts_052612.pdf [Accessed August 2015].

Sadre, R., Sperotto, A. & Pras, A., 2012. The effects of {DD}oS attacks on flow monitoring applications. *IEEE Network Operations and Management Symposium (NOMS)*, pp. 269-277.

Santanna, J. J., Durban, R., Sperotto, A. & Pras, A., 2015b. Inside booters: An analysis on operational databases. *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 432-440.

Santanna, J. J. & Sperotto, A., 2014. Characterizing and Mitigating the DDoS-as-a-Service Phenomenon. *Monitoring and Securing Virtualized Networks and Services*, pp. 74-78.

Santanna, J. J. et al., 2015a. Booters - An analysis of DDoS-as-a-service attacks. *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, pp. 243-251.

Steinberger, J. et al., 2016. In Whom Do We Trust - Sharing Security Events. *In Proceedings of the IFIP 10th International Conference on Autonomous Infrastructure, Management and Security*, June.

Steinberger, J., Sperotto, A., Baier, H. & Pras, A., 2015a. *Exchanging Security Events of flow-based Intrusion Detection Systems at Internet Scale.* [Online] [Accessed April 2016].

Steinberger, J., Sperotto, A., Golling, M. & Baier, H., 2015b. How to Exchange Security Events ? Overview and Evaluation of Formats and Protocols. *IFIP/IEEE International Symposium on Integrated Network Management (IM)*, May, pp. 261-269.

Biographies

Jessica Steinberger is a PhD student at the Design and Analysis of Communication Systems Group (DACS) of the University of Twente, The Netherlands and a scientific staff member of the Biometrics and Internet-Security Research Group (da/sec) of the University of Applied Sciences Darmstadt, Germany. She received a M.Sc. degree in Computer Science from the University of Applied Sciences Bingen, Bingen, Germany, in 2011. Her main topics of interest include mitigation and response to network-based attacks in context of high-speed networks.

José Jair Santanna is a PhD student at the Design and Analysis of Communication Systems Group (DACS) of the University of Twente, The Netherlands. He received a M.Sc. degree in Computer Science from the University of Rio Grande do Sul, Porto Alegre, Brazil, in 2012. His main topics of interest include DDoS as a Service.

Evangelos Spatharas is a security engineer at the G\'{E}ANT Ltd, Cambridge, UK. He received his M.Sc. degree in Systems, Information and Network's Security from the University of Bradford, UK, in 2013. His main work and responsibilities on a day to day basis include monitoring of network security issues through the use of NetFlow, logging and IDS tools, investigating and remediating.

Benjamin Kuhnert is a Master student and a research assistant at the Biometrics and Internet-Security Research Group (da/sec) of the University of Applied Sciences Darmstadt, Darmstadt, Germany. His main topics of interest include the development of a framework to mitigate network-based attacks using complex event processing.

Hendrik Amler is a Master student at the University of Applied Sciences Darmstadt, Darmstadt, Germany. His main topic of interest is the development of a framework to test mitigation and response capabilities.

Niklas Breuer is a Master student at the University of Applied Sciences Darmstadt, Darmstadt, Germany. His main topic of interest is the development of a framework to test mitigation and response capabilities.

Kristian Graul is a Master student at the University of Applied Sciences Darmstadt, Darmstadt, Germany. His main topic of interest is the development of a framework to test mitigation and response capabilities.

Ulrike Piontek is a Master student at the University of Applied Sciences Darmstadt, Darmstadt, Germany. Her main topic of interest is the development of a framework to test mitigation and response capabilities.

Anna Sperotto is an assistant professor at the Design and Analysis of Communication Systems Group (DACS) of the University of Twente, The Netherlands. She received a M.Sc. degree in Computer Science from the Ca'Foscari University, Venice, Italy, in 2006 and a PhD degree from the University of Twente, in 2010. Her main topics of interest include intrusion detection and traffic modeling.

Harald Baier is a professor at the Biometrics and Internet-Security Research Group (da/sec) of the University of Applied Sciences Darmstadt, Germany. He received a PhD degree for his thesis titled "Efficient Algorithms for Generating Elliptic Curves over Finite Fields Suitable for Use in Cryptography". His research interest include host based and network based intrusion detection, digital forensics, security protocols and security infrastructures.

Aiko Pras is a professor at the Design and Analysis of Communication Systems Group (DACS) of the University of Twente, The Netherlands. He received a PhD degree for his thesis titled "Network Management Architectures". His research interests include network management technologies, network monitoring, measurements and security. He is chairing the IFIP Technical Committee 6 on "Communications Systems", and is Project Leader of the European Network of Excellence on "Management of the Future Internet" (FLAMINGO). He is steering committee member of several conferences, including IM/NOMS and CNSM, and series/associate editor of ComMag and IJNM.