




A Multi-location Defence Scheme Against SSDP Reflection Attacks in the Internet of Things

Xin Liu¹(✉) , Liang Zheng¹, Shuai Cao¹, Sumi Helal², Jiehan Zhou^{3,4}, Hunfu Jia⁵, and Weishan Zhang¹

¹ College of Computer Science and Technology, China University of Petroleum, Qingdao 266580, China

lx@upc.edu.cn

² School of Computing, Lancaster University, Bailrigg, UK

³ Information Technology and Electrical Engineering, University of Oulu, Oulu, Finland

⁴ Electrical and Computer Engineering, University of Toronto, Toronto, Canada

⁵ Nankai University, Tianjin, China

Abstract. The proliferation of the Internet of Things (IoT) has led to a rapid increase in SSDP (Simple Service Discovery Protocol) reflection attacks. However, there is very scarce work on defending these attacks, with only some engineering advices on shutting down attacked services. This paper proposes a comprehensive approach to defend SSDP reflection attacks, which is called multi-location defence scheme (MLDS). MLDS operates at multiple places, working throughout the attacking link, starting from attack sources to victims, without prior detecting attacks. Attackers usually utilized bots in a botnet to launch attacks, but bots can act as defenders to carry out defence strategies in our MLDS, which is an unconventional approach to make the defence effective. Finally, we analyzed thoroughly packet traffic situations when deploying MLDS to different defence locations.

Keywords: Denial-of-service · DRDoS · SSDP reflection attack · TTL

1 Introduction

Countless devices have connected to the Internet, leaving the Internet of Things (IoT) exposed to many security threats without proper security mechanisms. Opened services on IoT devices may be exploited to launch different malicious attacks like the Denial-of-service (DoS) attacks [11], in the format of Distributed Denial of Service (DDoS), or the distributed reflective denial-of-service (DRDoS) [1], for financial, political or purely destructive motivations. During the DoS attack, attackers disrupt services of victims, which can be targeting servers or networks.

The DDoS attack is proliferating in the Internet of Things age. These attacks usually result in heavy network traffic or heavy load on victims. On October 21st in 2016, a series of DDoS attacks caused widespread disruption of legitimate Internet activities in the US [24]. These attacks were made possible by a large number of unsecured Internet-connected devices, such as home routers and surveillance cameras. According to the statistics from Kaspersky Lab [17], 50% of DDoS attacks led to noticeable disruptions of services and 24% of DDoS attacks resulted in services being completely unavailable.

In a DRDoS attack, an attacker makes fake requests by replacing source IP address with the IP address of the selected victim. He sends those fake requests to service providers which then send service response packets to the spoofed IP address. The sizes of those response packets from service providers in DRDoS attacks are always many times larger than that of request packets. Therefore service providers are also called reflectors or amplifiers, which may be various networked devices, such as PCs, printers, routers, WiFi access points, mobile devices, cameras, and so on. Vulnerable service providers are carefully selected as amplifiers by attackers, where response packets are much larger compared with request packets.

Attackers try to find vulnerabilities in various Internet protocols or services to amplify responses from service providers in order to significantly increase communication traffic. Rossow analyzed 14 protocols susceptible to bandwidth amplifications and gave a bandwidth amplification factor (BAF) to every protocol [23]. He found that UPnP enabled hosts can respond with a reply packet per service on Simple Service Discovery Protocol (SSDP) discovery requests.

According to the bi-annual DDoS Threat Report from NSFOCUS [22], the proliferation of IoTs is responsible for increased SSDP reflection attacks. From the Akamai's report on DDoS attacks (Q3 2016 to Q2 2017), the number of DDoS reflector source IPs with different kinds of Internet protocols is shown in Table 1. SSDP is the protocol most frequently used for reflection attacks in three of the four quarters [3].

Table 1. DDoS reflector source IP count

Protocol	2016 Q3	2016 Q4	2017 Q1	2017 Q2
SSDP	120800	508434	465979	426375
NTP	409646	299855	268338	267376
SENTINEL	34488	36119	50051	59270
CHARGEN	43304	47810	38848	39792
QOTD	27556	40474	30874	30026
RPC	36011	37657	31966	29858
TFTP	16313	22458	19670	18058

SSDP is part of the Universal Plug and Play (UPnP) Protocol standard. This protocol allows Internet devices to seamlessly discover each other's services. It

uses User Datagram Protocol (UDP) as the underlying transport protocol, which is based on HTTPU (HTTP UDP). Attackers have been abusing these protocols to initiate DRDoS attacks, amplifying and reflecting network traffic to their targets. Request packets from attackers are multicasting to service providers. SSDP uses 239.255.255.250 as its target IP address, which is a local multicast IP address. The request packets from SSDP clients to SSDP servers are transferred by multicasting to 239.255.255.250:1900 in local area network.

The United States Computer Emergency Readiness Team (US-CERT) first issued a warning about SSDP in January 2014 [26], and in October 2014 it was used to generate 54 Gbps of traffic in a single attack. PLXsert has observed the first use of the DRDoS attack that abuses SSDP [2]. The threats come from millions of networked devices which can be abused as reflectors by attackers.

For a regular SSDP service (shown in the left part of Fig. 1), a request sender will receive responses from service providers. But if an attacker want to conduct a SSDP reflection attack, he will first collect vulnerable hosts/devices on the Internet as bots to establish a botnet, the attack can control the request sender and spoof the IP address of requests packets using the victim's IP address as follows (shown in the right part of Fig. 1). A botnet known as a zombie army is a number of Internet computers that, although their owners are unaware of it, have been set up to send malicious packets to attack victims, which are servers or networks on the Internet. Second, the attacker will send both commands and vulnerable device lists to those bots in the botnet. According to the commands from the attacker, each bot sends a SSDP request with forged source IP address, which is the IP address of the target, to those vulnerable devices. The actual response receiver is not the requests sender, instead the victim become the receiver. Then massive responses from those vulnerable devices will bombard the server, which leads to a peak stage when massive packets are beyond the processing capabilities of the server. And in a SSDP reflection attack, the attack will exploit lots of request sender on the Internet as bots to establish a botnet, which is a challenge, especially for a large-scale botnet.

Work on defending SSDP reflection attacks. We found only some simple suggestions: UPnP requests should be blocked or UPnP service should be disabled to reduce SSDP reflection attacks is scarce. This limits the availability of regular UPnP services. On the other hand, we can get some ideas from some work on DDoS attacks. For example, Pack et al. [20] proposed to set parameters on servers and routers to disable services when there are attacks. Yan et al. proposed an algorithm that can use different time slice allocation strategies according to the intensity of DDoS attacks to ensure protection to a normal switch under DDoS attacks [29], which work on the victim side. Peng et al. presented an approach where reflectors monitor incoming packets and warn other potential reflectors when any abnormal traffic is observed [21]. This approach works on the service provider side. These works motivate us to think an integrated approach can be used at multiple locations to design a multi-location defence scheme (MLDS), which can work collaboratively at different locations. The majority of the exist-

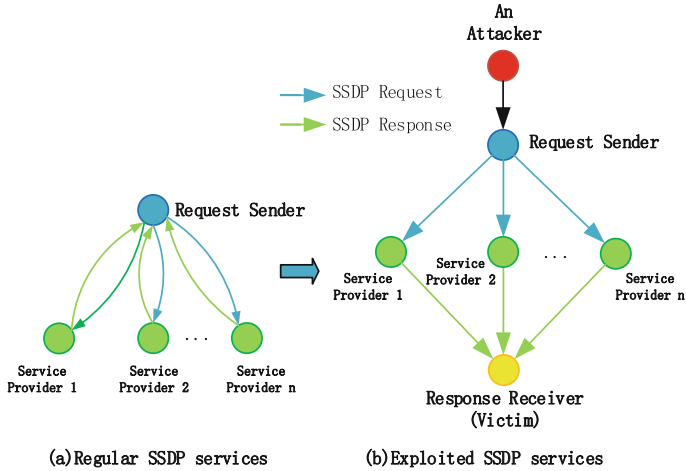


Fig. 1. Regular SSDP services (a) and exploited SSPD services with defending deployments (b)

ing defence mechanisms are designed based on the fact that attacks have to be detected. MLDS doesn't need to do this.

On the other hand, the existing defence schemes against DRDoS attacks do not consider deploying defence mechanisms to the source of attacks, because attackers are in control of the source of attacks. When we take a closer look at the SSDP reflection attacks, we can see that it is possible to deploy a defence mechanism to the source of the attack, and this will be much effective. To take the target out of service, an attacker usually utilizes a controller to instruct bots to launch an attack. The controller is under the control of the attacker for a SSDP reflection attack, but those bots exploited by the controller are not fully controlled by the attacker. This means regularly used service on those bots can be utilized to defend the victim. In MLDS, SSDP reflection attacks can be mitigated not only by reducing the amplification at reflectors and limiting the number of received response packets, but also by limiting requests at the source of attacks. We try to take full advantage of all possible resources throughout the attack link, especially the ones at the source of the attack.

The contributions of the paper include:

- We propose a comprehensive defence scheme called MLDS for SSDP reflection attacks. It has three main features:
 - MLDS working throughout the attacking link, starting from attack sources to victims
 - MLDS not depending on detecting attacks
 - MLDS is adopting an unconventional way of defence to make bots acting as defenders to carry out defence strategies, which makes MLDS very effective.

- According to the above characteristics, We thoroughly analyze traffic situations in the whole attack link when MLDS is deployed to different locations.

The remainder of the paper is as follows: Sect. 2 reviews related work on defence schemes. Section 3 presents the MLDS details, illustrates the DRDoS attack model in LANs, and then calculates the traffic for each attack location. Section 4 gives the conclusion and future work to be done.

2 Related Work

Work on defending SSDP reflection attacks is scarce. Pack et al. [20] just focused on setting parameters on servers and routers such as disabling underlying services. However, there are quite some works on defending against DDoS attacks [8, 19, 24], which can give us some hints. In this study we classify defending schemes for DDoS attacks into four categories according to the deployment location of the defending schemes.

A. Defending on routers

Ioannidis and Bellovin used Pushback added to upstream routers to drop attack packets with attack signatures that consist of selected prefixes of destination addresses [13]. If this method is used to resist SSDP attacks, dropping SSDP packets may affect normal services of UPnP.

Wang and Reiter proposed a distributed puzzle mechanism in which routers distribute puzzles to clients to require puzzle solutions to consume clients' resource, that is, clients as bots need more resources for attacks. Routers cooperate with each other to check network traffic and then defend networks against flooding attacks [27].

Pack et. al. used ACL (Access Control List) rules to distinguish attack packets from legitimate traffic based on source addresses in packets. These ACL rules were deployed on routers [20].

Dietzel et. al. proposed a blackholing technique that allows a peer via BGP (Border Gateway Protocol) to announce a prefix to another peer which then discards packets destined for this prefix among Internet Exchange Points to mitigate the effectiveness of DDoS attacks [7].

Mirkovic et al. proposed D-WARD which can gather two-way traffic statistics and detect attacks, and then adjust rate limit rules for suspected source addresses to modify associated traffic flows [18].

Chen and Park [6] proposed an Attack Diagnostic (AD) system in which DoS attacks are detected near the victim, and packet filtering is executed at the router close to the attacker. The victim can trace back attack traffic to attack sources and then issues messages that command AD-enabled routers to filter attack packets close to the source.

Huistra proposed that amplification attacks can be detected specifically by monitoring Domain Name System(DNS) packet sizes as well as the number of packets across multiple routers in which the victim and the source of the attack can be discerned [12].

Wei et al. proposed a method to locate suspicious flows on an upstream router then discard these flows on the routers [28].

B. Adding dedicated equipment

Kambourakis et al. deployed a monitor to record both DNS requests and responses using the IPtraf tool, which is a console-based network statistics utility for Linux. It collects a variety of statistics such as TCP connection packet and byte counts [15].

Kim et al. proposed PacketScore in which they prioritized packets based on a per-packet score to estimate the legitimacy of a packet given the attribute values it carries. They used a DDoS Control Server (DCS) to collect reports from routers across the Internet [16].

Saied et al. proposed a method to detect and mitigate known and unknown DDoS attacks in real time environment. They used artificial neural network (ANN) to detect DDoS attacks based on specific characteristic features (patterns) that separate DDoS attack traffic from genuine traffic [25].

Monowar et al. empirically evaluated several major information metrics such as namely, Hartley entropy, Shannon entropy, Renyi's entropy, to detect both low-rate and high-rate DDoS attacks. These metrics can be used to describe characteristics of network traffic data, and they proposed a model to detect both low-rate and high-rate DDoS attacks [5].

C. Defending at service providers

Peng et al. proposed that each potential reflector could be used to monitor incoming packets and broadcast warning messages to other potential reflectors if any abnormal traffic was detected [21].

Alqahtani et al. proposed a DDoS attack detection approach for service clouds and developed efficient algorithms to resolve the originating service for the attack. The detection approach is composed of four levels such that each level detects symptoms of DDoS attacks from its local data. The detection results of all levels are collaborated to confirm the victim and attacking services. They evaluated their proposed solution using a random dataset [4].

Uzair et al. have combined Ethereum with the traditional IoT to form a decentralized IoT infrastructure that not only prevents malicious devices from accessing servers, but also solves DDoS attacks by using static resource allocation of devices [14].

D. Defending at victims

Yan et al. proposed an effective software-defined networking controller scheduling method to mitigate DDoS attacks. The algorithm can adopt different time slice allocation strategies according to the intensity of DDoS attacks, and use SDN controllers to handle the traffic of different switches, so as to better protect the switches from DDoS attacks in the network [29].

Gilad et al. presented an approach using CDN (content distribution network) that adopts a CDN-on-Demand, software-based defence scheme for small

to medium websites to resist powerful DDoS attacks, at a fraction of the cost of commercial CDN services. When excessive load is detected, CDN-on-demand provides services to clients from proxies that are automatically deployed on various cloud service providers [10].

Attackers usually utilize a botnet to launch DRDoS attacks. From the survey, we haven't found other work that utilized bots in a botnet to defend against reflection denial-of-service attacks, as it is not a conventional approach to make the bots controlled by the controller in a botnet acting as a defender. Usually the user isn't aware that his computer is executing the controller's instruction as his computer still works well normally. Therefore, we have opportunities to utilize the communication function at bots, which is not restricted by the controller, so as to limit the number of requests sent to service providers. That is to say, we can add applications or set parameters on these bots to carry out our defence strategies. Additionally the majority of the existing defence mechanisms are designed based on the fact that attacks have to be detected. Our proposed scheme can be deployed directly to different locations without prior detecting the attacks.

3 Multi-location Defence Scheme

Intuitively, we can design an integrated defence mechanism, working throughout the attacking link, starting from the attack source to the victim. Deploying defence mechanisms to the source of attacks can make the defence very effective, because that is the place where attacks are launched. Additionally, we can make full use of limited resources in the attack sources in an unsafe environment to enhance the security of IoTs.

3.1 Deploying Defence Scheme at Multiple Locations

According to the process of a SSDP reflection attack discussed in introduction, to defend against SSDP reflection attacks, we deploy different defences schemes at multiple locations including the request sender, the service provider and the victim.

For a SSDP reflection attack, after request senders receive the instruction from the controller, they will send discovery packet requests to service providers using SSDP at an unusually high frequency, which is different from that of regular users. So it will be effective to deploy defence at request senders which are attack sources. We limit the number of requests at request senders when they are partially controlled by the controller in a preset time interval. In this way, the traffic from request senders to service providers and victims will be reduced.

After the SSDP requests are received, service providers will send their responses to the spoofed IP address, and the size of the response is many times that of the request. We set the time interval between the same two response packets to the same target at each reflector, which can limit the number of response packets sent from reflectors, and then reduce the reflectivity of service providers.

Another approach at reflectors is setting the Time-To-Live (TTL) value to a reasonable value to limit the distance the packets propagate on the Internet, which can ensure the remote responses can't reach the victim. UPnP enable seamless connection in a home network or a business network or between two home or small business networks thus allowing UPnP devices in a home or small business network discover and interact with UPnP device in another home or small business network using SSDP [9]. According to the scenarios where SSDP is applied TTLs can be set to suitable values to avoid the underlying responses reaching the victim.

When an attacker launches a SSDP reflection attack, the target of the attack is the response receiver. Therefore in our MLDS, the response receiver will drop the same packets from the same service providers with no side-effects to normal services. This can reduce the processing time of SSDP reflection attacks in MLDS.

In summary, if a node acts as a sender, or an amplifier, or a victim in different attacks at a SSDP node, we should deploy all those strategies before it joins the Internet. Each set of strategies at each kind of SSDP roles are designed as a plug-in, and these plug-ins are packaged as a MDLS package, which can be one type of software security packages downloaded easily on some official websites that issue security improvements.

To launch a massive SSDP reflection attacks, an attacker usually utilizes a lot of bots organized as a botnet. Figure 2 shows an SSDP reflection attack in LANs. If there are not enough vulnerable nodes to utilize for attackers, it is hard to organize effective attacks. There are several ways for SSDP nodes to install plug-ins, such as downloading MDLS from official websites as mentioned before or integrating MDLS into newly produced SSDP devices. Then there will have more SSDP nodes gradually which cannot be exploited by SSDP attackers. As the number of nodes equipped with MLDS increases, the number of vulnerable nodes which can be utilized by attacker will decrease. Therefore, the attackers are unable to implement an effective attack.

An attacker instructs a botnet composed by B_1, B_2, \dots, B_m to send spoofed requests simultaneously to those service providers A_1, A_2, \dots, A_k via a controller (C). Those service providers send amplified responses to the same target, leading to a dramatic increase of traffic flow. The analysis of traffic flow is detailed as follows.

- First, the attacker uses a controller (C) to instruct a botnet and command bots to send requests to service providers in an infinite loop until the victim is down.
- Next, those bots send spoofed requests simultaneously to those service providers A_1, A_2, \dots, A_k . We assume that the size of a request is t_1 and the controller commands the bots request in an infinite loop. So we can calculate the traffic from a bot TR_{fromB} using (1).

$$TR_{fromB} = n * t_1 \tag{1}$$

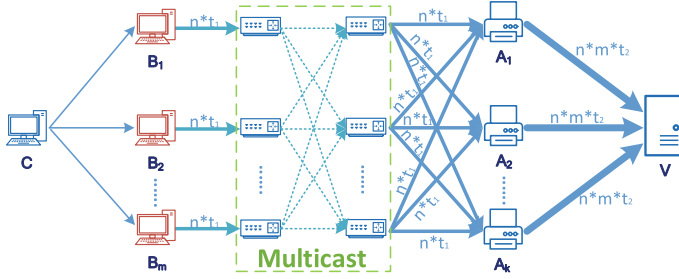


Fig. 2. Traffic flow during a SSDP reflection attack in LANs

Where n is the number of request cycles. Those bots will send spoofed requests by using multicast addresses in LANs, and then these request packets are distributed through routers to the target victim. According to a vulnerable device list the requests will be sent to multiple service providers by n times.

- Those service providers then send amplified responses to the same target, the victim (V). We set the size of the response to t_2 . Every service provider receives n requests from every bot. And there will be m bots sending amplified packets to the victim simultaneously. We can calculate the traffic from a service provider to the victim TR_{fromA} as (2).

$$TR_{fromA_i} = n * m * t_2 \tag{2}$$

The total traffic T_v for a SSDP reflection attack can be calculated as (3).

$$T_v = k * n * m * t_2 \tag{3}$$

where k is the number of the service providers.

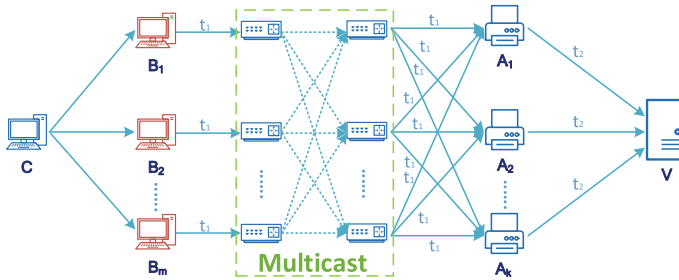


Fig. 3. The restrained attack traffic by the MLDS in LANs

From the above analysis, deploying different defence strategies to multiple locations will make the defence work efficiently in the whole attack link, from

the attack initiating source to the victim. This is the reason why we design the MLDS.

For request senders, there is no retransmission mechanism in SSDP protocol. We can limit that each bot sends a SSDP request only once within a time interval to avoid malicious requests being sent repeatedly, in order to reduce response packets.

For service providers, the number of response packets to the same target can be limited. We count the requests received from different senders with the same source IP address. The response to the target is allowed only once within a time interval.

For victims, if the same packets arrive to the same response receivers, the following packets should be discarded.

The MLDS works as follows (Fig. 3):

- MLDS at the requests sender will limit the number of requests from B_1, B_2, \dots, B_m . We add a counter to count the same requests at B_1, B_2, \dots, B_m if the port is the 1900 and the packet sent is the same as SSDP discover request. We also add a timer to record the time starting from the time when the value of the counter is 1. The initial value of the counter is set to 0. Before a bot send a request it should check the counter. If the value of the counter is 1, the request should be stopped. Otherwise the request is sent regularly. After a request is sent the counter is set to 1. When the time interval is running out, its value is set to 0.

That is, the network traffic can be restricted at the request senders. If each bot can limit their numbers to 1, each service provider will receive m requests from m bots. We can calculate the traffic from request senders to each service provider T'_{toa} as (4).

$$T'_{toa} = m * t_1 \quad (4)$$

- After a request is received, A_1, A_2, \dots, A_k can limit the number of its responses to V using the following method. We add a timer to record the time between two consecutive response packets. We set the interval threshold time between two packets. If the time is shorter than the interval threshold, the reflectors should stop the next response packets to the same target in this time interval. At the same time before an IP packet is sent, the value of TTL should be set according to the distance (how many hops) from the request sender to the service provider and the distance from the service provider to the victim. After each service provider limits their responses to V in an interval to only once, the corresponding load received at the victim from k service providers T'_v can be calculated as (5).

$$T'_v = k * t_2 \quad (5)$$

We can see that the traffic T'_v is reduced dramatically comparing with T_v .

- At the victim, We add a counter to count the same responses from the same service provider. If the value of the counter is 1, the following packets within the time interval should be discarded. If some service providers don't limit the number of responses, the victim will ensure the traffic T'_v is $k * t_2$.

4 Conclusion

The proliferation of IoTs is confronted with increasing SSDP reflection attacks. However, there are very scarce studies on a comprehensive solution for defending SSDP reflection attacks. Only some advices on disabling UPnP services exist. The work on DDoS attacks proposed some defence schemes here and there in the attack link, without considering tackling the attacks from the source end. In this paper, we propose an integrated approach called MLDS, where a multi-location defence scheme is designed and deployed to different places in the whole attack link, which can work collaboratively at different locations. MLDS does not depend on detecting attacks like other existing approaches, and resolves the defence problem from the key part by deploying defence strategies to attack sources, service providers, victims, etc. We try to make full use of all possible resources in the whole attack link, especially the resources at the source of attacks. The article show that tackling security attacks from the very beginning of attacking sources is the most effective approach, and also the integrated defence scheme in the whole attack link is a comprehensive solution which can be a reference for resolving other security attacks.

Acknowledgements. This work is supported by the Key Research Program of Shandong Province (No. 2017GGX10140), the Fundamental Research Funds for the Central Universities (19CX05027B, 19CX05003A-11) and the National Natural Science Foundation of China (61702399, 61772291, 61972215).

References

1. Distributed Reflection Denial of Service Attacks. Accessed April
2. Akamai: SSDP REFLECTION DDOS ATTACK, akamais [state of the internet]/Threat Advisor
3. Akamai: State of the internet security 4(2) (2017)
4. Alqahtani, S., Gamble, R.F.: DDoS attacks in service clouds. In: 2015 48th Hawaii International Conference on System Sciences, vol. 1, pp. 5331–5340, January 2015. <https://doi.org/10.1109/HICSS.2015.627>
5. Bhuyan, M.H., Bhattacharyya, D., Kalita, J.: An empirical evaluation of information metrics for low-rate and high-rate DDoS attack detection. *Pattern Recogn. Lett.* **51**, 1–7 (2015)
6. Chen, R., Park, J.M.: Attack diagnosis: throttling distributed denial-of-service attacks close to the attack sources. In: 14th International Conference on Computer Communications and Networks, pp. 275–280. IEEE (2015)
7. Dietzel, C., Feldmann, A., King, T.: Blackholing at IXPs: on the effectiveness of DDoS mitigation in the wild. In: Karagiannis, T., Dimitropoulos, X. (eds.) PAM 2016. LNCS, vol. 9631, pp. 319–332. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-30505-9_24
8. Douligeris, C., Mitrokotsa, A.: DDoS attacks and defense mechanisms: classification and state-of-the-art. *Comput. Netw.* **44**, 643–666 (2004)
9. UPnP Forum FROUM: UPnP remote access-connecting two home or small business networks, June 2012

10. Gilad, Y., Goberman, M., Herzberg, A., Sudkovitch, M.: CDN-on-demand: an affordable DDoS defense via untrusted clouds. In: Network and Distributed System Security Symposium (2016)
11. Handley, M., Rescorla, E., IAB: Internet denial-of-service considerations. RFC 4732, RFC Editor, January 2006. <http://www.ietf.org/rfc/rfc4732.txt>
12. Huistra, D.: Detecting reflection attacks in DNS flows. In: 19th Twente Student Conference on IT, February 2013
13. Ioannidis, J., Bellovin, S.M.: Implementing pushback: router based defense against DDoS attacks. In: Proceedings of Network and Distributed System Security Symposium (NDSS) (2002)
14. Javaid, U., Siang, A.K., Aman, M.N., Sikdar, B.: Mitigating IoT device based DDoS attacks using blockchain. In: Conference Paper, June 2018
15. Kambourakis, G., Moschos, T., Geneiatakis, D., Gritzalis, S.: Detecting DNS amplification attacks. In: Lopez, J., Hämmerli, B.M. (eds.) CRITIS 2007. LNCS, vol. 5141, pp. 185–196. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-89173-4_16
16. Kim, Y., Lau, W.C., Chuah, M.C., Chao, H.J.: PacketScore: a statistics-based packet filtering scheme against distributed denial-of-service attacks. *IEEE Trans. Dependable Secure Comput.* **3**(2), 141–155 (2006)
17. Lab, K.: DENIAL OF SERVICE: how businesses evaluate the threat of DDoS attacks IT security risks special report series (2014)
18. Mirkovi, J., Prier, G., Reiher, P.: Source-end DDoS defense. In: Second IEEE International Symposium on Network Computing and Applications, pp. 171–178. NCA, IEEE (2003)
19. Mirkovic, J., Reiher, P.: A taxonomy of DDoS attack and ddos defense mechanisms. *Newsl. ACM SIGCOMM Comput. Commun. Rev.* **34**, 39–53 (2004)
20. Pack, G., Yoon, J., Collins, E., Estan, C.: On filtering of DDoS attacks based on source address prefixes. In: Securecomm and Workshops, September 2006
21. Peng, T., Leckie, C., Ramamohanarao, K.: Detecting reflector attacks by sharing beliefs. In: Global Telecommunications Conference, pp. 1358–1362 (2003)
22. Reading, D.: Report: IoT connected devices leading to rise in SSDP based reflection attacks. Accessed 21 Apr 2015
23. Rossow, C.: Amplification hell: revisiting network protocols for DDoS abuse. In: Proceedings of NDSS. Internet Society (2014)
24. Ryba, F.J., Orlinski, M., Wahlisch, M., Rossow, C., Schmidt, T.C.: Amplification and DRDoS attack defense - a survey and new perspectives. arXiv preprint (2015)
25. Saied, A., Overill, R.E., Radzik, T.: Detection of known and unknown DDoS attacks using artificial neural networks. *Neurocomputing* **172**, 385–393 (2016)
26. US-CERT: UDP-based amplification attacks (2014)
27. Wang, X., Reiter, M.K.: Mitigating bandwidth-exhaustion attacks using congestion puzzles. In: 11th ACM Conference on Computer and Communications Security, pp. 257–267 (2004)
28. Wei, W., Chen, F., Xia, Y., Jin, G.: A rank correlation based detection against distributed reflection DoS attacks. *Commun. Lett.* **17**(1), 173–175 (2013)
29. Yan, Q., Gong, Q., Yu, F.: Effective software-defined networking controller scheduling method to mitigate DDoS attacks. *Electron. Lett.* **53**(7), 469–471 (2017)