Proceedings of the Asia-Pacific Advanced Network 2013 v. 36, p. 32-40.
http://dx.doi.org/10.7125/APAN.36.5
ISSN 2227-3026

The flow back tracing and DDoS defense mechanism of the TWAREN defender cloud

Ming-Chang Liang [1,*], Meng-Jang Lin [2], Li-Chi Ku [3], Tsung-Han Lu [4], Jyun-Jie Chen[5] and Ta-Yuan Chou[6]

National Center for High-performance Computing / No. 28, Nan-Ke 3rd Rd., Hsin-Shi Dist., Tainan City, Taiwan, R.O.C. 74147

E-Mails: {liangmc, n00lmj00, lku, kileleu, jjchen, 1203053}@narlabs.org.tw

* Tel.: +886-6-5050940#724; Fax: +886-6-5055909

Abstract: The TWAREN Defender Cloud is a distributed filter platform on the network backbone to help defending our connecting institutions against malicious network attacks. By combining the security reports from participating schools, this system can block the incoming threats from the entry points, thus it helps protecting all connecting institutions in the most economic and effective way. This paper aimed at explaining the analyzer design, its mechanism to back trace DDoS attack flows to their entry points and the defense mechanism it provides to block the threats.

Keywords: cloud computing; distributed processing; information security; backbone defending; flow back tracing; TWAREN.

1. Introduction

The network security issue becomes increasing more prominent over time. While most universities and research institutions tend to buy their own security equipment to defend their network, the equipment capable of handling high network bandwidth could be very expensive and unaffordable to many schools. Furthermore, these independent defense mechanisms are hard to integrate and collaborate without intensive human intervention. Obviously it will be very valuable if these individual efforts can be integrated to become an overall protection mechanism to protect every institution on the network.

To realize the idea, we designed the TWAREN Defender Cloud, shown as a diagram in **figure 1**. The TWAREN Defender Cloud is a distributed platform which collects security reports and

flow records from the backbone and connecting institutions and then reacts to security threats by analyzing and blocking them from the backbone. Since TWAREN[1] has deployed network equipment in all GigaPOPs, we are able to collect the flow data all over the backbone network. In addition, if schools and connecting institutions are willing to provide their flow data to us, the integrated information can cover almost the entire country. Once a malicious activity is detected and reported from any security detection device within this giant network, the IP and port information of the malicious activity can be immediately analyzed and used to inspect the entire backbone flows to discover other victims or other malicious origins of the same attack. Then the administrators can be notified and malicious flows blocked accordingly. Such defense is only possible from the backbone perspective because any individual security device in the schools does not have the overall information about the malicious activity. Furthermore, DDoS attacks usually fake their source IP, thus only a backbone defense mechanism can back trace the DDoS flows to their true entry points to the backbone and block them from there. By protecting the backbone in this integrated way, other connecting institutions without their own security equipment can also get benefit. Thus the backbone defense mechanism is very economic and effective.

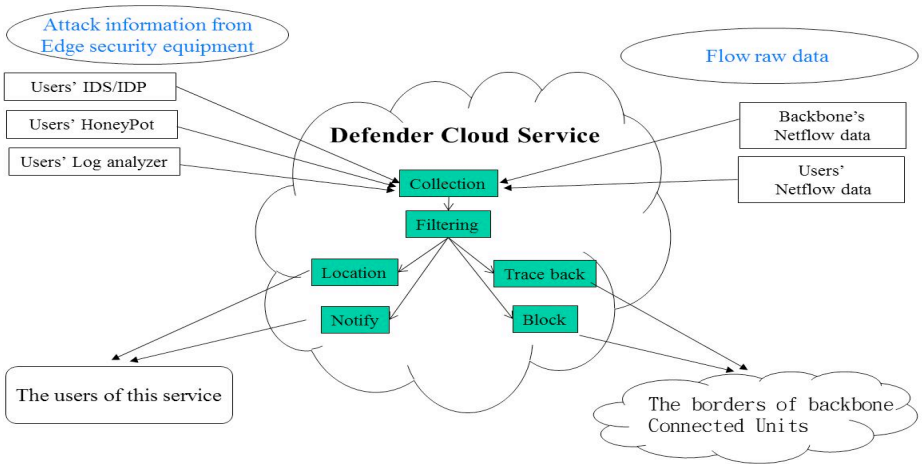

Figure 1. The backbone view of the Defender Cloud.

In order to handle the huge amount of backbone flow information, a highly scalable distributed filter platform has been designed and implemented as shown in **figure 2**. The filters were designed to compare flows against blacklists in a very high speed and only those targeted flows are sent to the backend analyzer to significantly reduce the loading and processing time of the analyzer. The result has been published in the Network Research Workshop in APAN32[2].

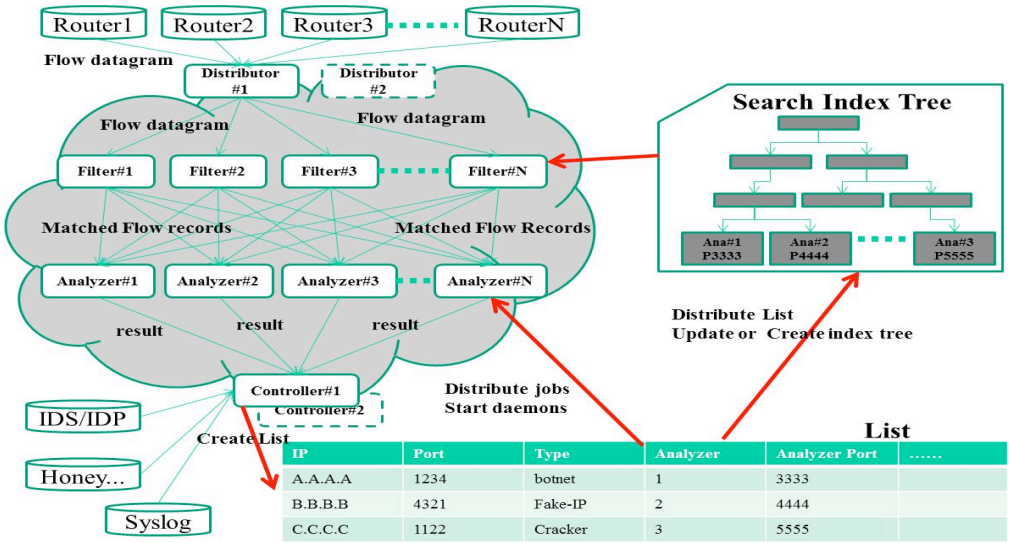

Figure 2. The architecture of the TWAREN backbone defender cloud.

Due to the distributed design of the TWAREN Defender Cloud and the use of UDP to exchange flow records among different modules, the components of the TWAREN Defender Cloud are highly modular and scalable, as shown in **figure 3**. In addition, incorporating external analyzers into the system will be very easy, thus the platform itself can serve as a collaborative environment for external researchers to study the real world network threats. Of course, the flow data will be transformed in prior to protect the user privacy.

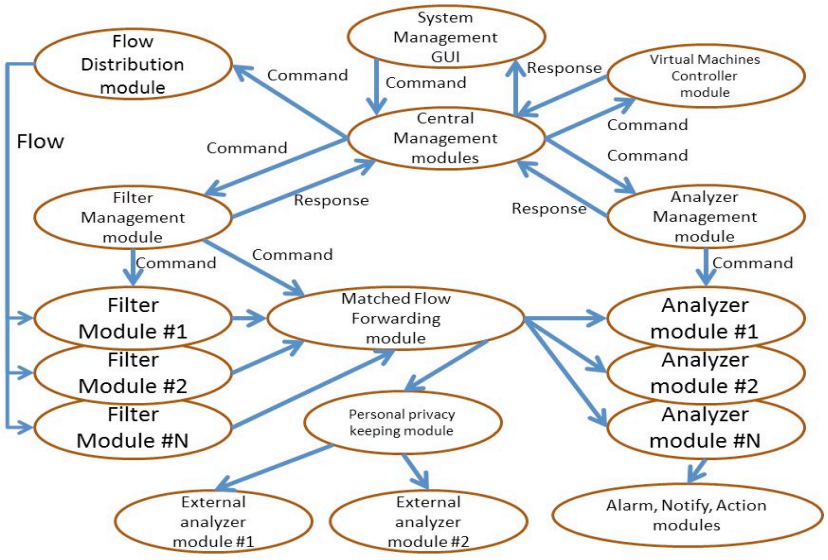

Figure 3. The main components of the defender cloud.

Besides, we designed a module to automatically import the security reports from security equipment and then integrate them into comparison rules. To further reduce unnecessary alarms, the module has been designed to make the comparison rules compact by sorting the security reports according to their frequency, total number and impact. The result has been published in APAN34[3].

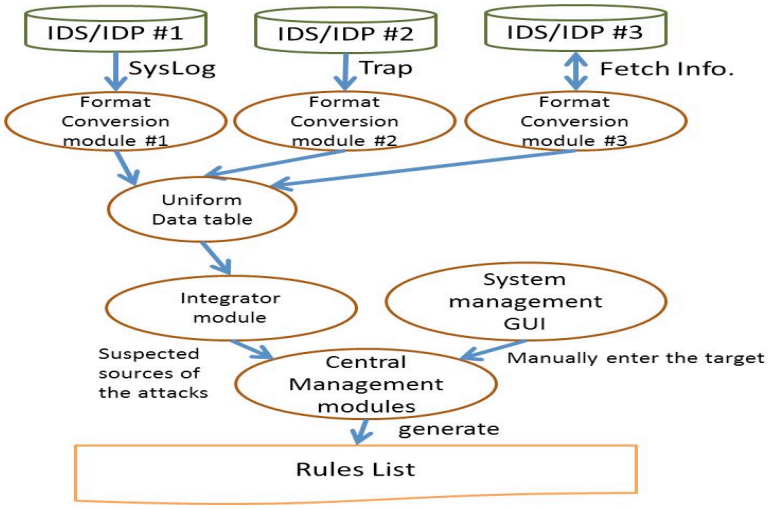

Figure 4. The generation of the comparison rules.

As shown in **figure 4**, there are three main sources of the comparison rules. The first one is from the integration of security reports from security equipment. The second comes from the freely available C&C data and the third one from the manually input rules, which is mainly used to protect specific servers. After sending the rules to the filters, the comparison trees will be automatically built and maintained, as shown in **figure 5**.

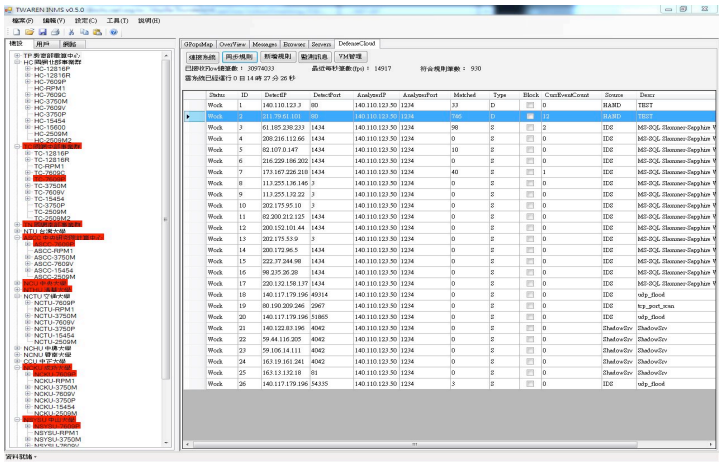

Figure 5. Comparison rules shown in the Defender management system.

After the completion of the filter and rule importing frontend, we shifted our focus to the development of the analyzer. As the backbone defender, DDoS attacks are our first priority to deal with because they are hard to defend within individual schools. DDoS attacks usually come from a lot of places and their source IPs are usually fake. From the perspective of security equipment in schools, due to the lake of the overall scope, the only thing they can do is to passively block those known attack flows. Meanwhile, from the backbone perspective, those DDoS flows can be back traced to their origins. Whether they come from peer networks or connecting institutions, we can notify their administrators to help stop the attack or check the infected computers. Thus the TWAREN Defender Cloud can help a lot from its backbone standing point once the DDoS analyzer is developed.

2. Issues and Methods

Each flow record only contains the information of the incoming and outgoing port numbers when the flow passes through any individual router. Therefore, in order to determine the full path of the flow and the true origin of a DOS attack, a mechanism to discover the topology of the whole backbone network will be necessary. In this section we explains the issues we faced and the solution we designed during the development of the DDoS flow analyzer and the defense mechanism.

2.1. The flow back tracing analysis

The DDoS analyzer needs to know the source routers of each flow records before these partial information can be assembled into the full path of the flow. However the header and the payload of the flow record do not contain the IP of the router, thus the origin of the flow record can only be identified by the source IP of the UDP datagram in which the flow record is sent in the first place. Unfortunately, this information is no longer intact when the analyzer finally receives them, because the flow records need to be processed by the dispatcher and the filter beforehand.

There are two possible ways to keep the IP of the source router somewhere in the flow records: the first possibility is to keep (overwrite) it in the last four bytes of the flow header, originally representing the Engine_Type, Engine_ID and Sampling_Interval, which are not useful for network security. The second possible way is to keep (rewrite) it in the source IP field of the flow record datagram, as if the packet is sent directly from its very source router. Because the first way would makes the flow header incompatible to industrial standard, thus limiting our possible collaboration with the academic researchers, the second way is therefore favored.

The Defender Cloud back traces DDoS attacks from the victim, therefore all suspicious flows sending from the filters to the DDoS analyzers have the same (the victim) IP as their destination IP. Starting with this IP, the full path of the attack flow is then constructed by assembling the flow records sending from all routers along the path. Due to their belonging to the same flow, all

these flow records share the same source and destination IP, with the only difference being the respective incoming and outgoing router interfaces, which are denoting by their SNMP index number. By inspecting the Next Hop field of the flow, the full path of the flow can be back traced by looking up the source IP of the flow datagram (which is the IP of the source router) and the interface SNMP index against the interface address database. As a consequence, the interface address database, which actually represents the full layer-3 topology of the backbone, needs to be established in prior.

Because TWAREN is a hybrid network, rather than wiring directly with each other, the routers may be interconnected by optical light-paths, layer-2 Ethernet switches or VPLS-VPN switches. While the physical topology can be obtained from our network management system, it is far too complex for the purpose of flow back tracing. Therefore we modified the network management system to produce a layer-3 only topology database. By extracting the running-config of all routers, the IPs of all router interfaces are obtained and then those belonging to the same subnet are grouped together. Eventually the layer-3 relationship among all routers will be constructed.

However, the pure layer-3 information is not necessarily enough to represent the network topology. For example, rather than 1:1 wiring, multiple routers may be connected by a VLAN or VPLS-VPN, and sometimes some of these routers do not have IP configured on their interfaces. Thus extra layer-2 information is necessary to fill the gap and make the topology information useful. The SNMP MIB for link neighbor discovery, such as the Cisco Discovery Protocol (CDP) and Juniper Neighbor Discovery Protocol, is used to identify the topology relationship when the interfaces do not have IP configured.

Taking equipment capable of running Cisco CDP as an example, the three MIBs listed in the table 1 will be used.

Table 1. MIBs for Cisco CDP capable equipment.

MIB OID	Object Name
1.3.6.1.4.1.9.9.23.1.1.1.1.6	cdpInterfaceName
1.3.6.1.4.1.9.9.23.1.2.1.1.6	cdpCacheDeviceId
1.3.6.1.4.1.9.9.23.1.2.1.1.7	cdpCacheDevicePort

The "cdpInterfaceName" stands for the interface name of the query machine, "cdpCacheDeviceId" shows the name of the neighbor device and "cdpCacheDevicePort" means the interface name of the neighbor device which directly connects to the query machine. For example, when we query the "cdpCacheDeviceId" MIB from a router, the returning result may look like:

```
SNMPv2-SMI::enterprises.9.9.23.1.2.1.1.6.49.135 = STRING: "HC-3750P.twaren.net"
SNMPv2-SMI::enterprises.9.9.23.1.2.1.1.6.53.124 = STRING: "TN-7010.twaren.net"
SNMPv2-SMI::enterprises.9.9.23.1.2.1.1.6.54.169 = STRING: "HC-12816P.twaren.net"
SNMPv2-SMI::enterprises.9.9.23.1.2.1.1.6.58.164 = STRING: "HC7609-1A.twaren.net"
```

```
SNMPv2-SMI::enterprises.9.9.23.1.2.1.1.6.59.165 = STRING: "HC7609-1A.twaren.net"
```

The MIB values in the previous example shows the device names of the neighbor devices. And the numbers right after the string "enterprises.9.9.23.1.2.1.1.6." represent the index number of the CDP interface of the query machine. In the example, they are 49, 53, 54, 58 and 59. If we query the MIB values of the "cdpCacheDevicePort", it shows:

```
SNMPv2-SMI::enterprises.9.9.23.1.2.1.1.7.49.135 = STRING: "GigabitEthernet1/0/25"
SNMPv2-SMI::enterprises.9.9.23.1.2.1.1.7.53.124 = STRING: "Ethernet7/8"
SNMPv2-SMI::enterprises.9.9.23.1.2.1.1.7.54.169 = STRING: "GigabitEthernet6/1"
SNMPv2-SMI::enterprises.9.9.23.1.2.1.1.7.58.164 = STRING: "GigabitEthernet2/1"
SNMPv2-SMI::enterprises.9.9.23.1.2.1.1.7.59.165 = STRING: "GigabitEthernet2/3"
```

The numbers behind "enterprises.9.9.23.1.2.1.1.7." are also the index number of the CDP interface of the query machine, which tightly correspond to the result of the "cdpCacheDeviceId" query. If we query the "cdpInterfaceName", the result looks like:

```
SNMPv2-SMI::enterprises.9.9.23.1.1.1.1.6.49 = STRING: "GigabitEthernet2/1"
SNMPv2-SMI::enterprises.9.9.23.1.1.1.1.6.53 = STRING: "GigabitEthernet2/5"
SNMPv2-SMI::enterprises.9.9.23.1.1.1.1.6.54 = STRING: "GigabitEthernet2/6"
SNMPv2-SMI::enterprises.9.9.23.1.1.1.1.6.58 = STRING: "GigabitEthernet2/10"
SNMPv2-SMI::enterprises.9.9.23.1.1.1.1.6.59 = STRING: "GigabitEthernet2/11"
```

Using the interface index as the key to join the three query results, the layer-2 relationship of those connected devices can be discovered. For example, using the interface index 49 to join the query results, we know that the interface GigabitEthernet2/1 of the query machine has a direct layer-2 connection to the GigabitEthernet1/0/25 of the neighbor "HC-3750P.twaren.net".

After the layer-3 information assembly and the layer-2 neighbor discovery, the input interface of each flow record will be able to chain to the output interface of previous hop router and the output interface to the next hop router. Consequently the full path of the flow can be constructed. The two edge devices and interfaces are especially important and will be used in the defense mechanism against the DDoS attack.

2.2. Defense mechanism

Because DDoS attack usually comes from a large number of zombie computers (Bot), blocking all of them by adding a lot of access control lists (ACL) to routers can severely undermine the router performance. Instead of attempting to completely block the DDoS flows, our defense mechanism is designed to mitigate it by finding out those source routers having largest incoming DDoS impact and only blocking them there.

Firstly, every DDoS flows are back traced from the victim IP to their corresponding origins, with all involved nodes and links along the flow paths being separately counted. For those nodes and links being passed more times by different DDoS flows, they get higher counting, which means that they have higher impact than other nodes and links in this attack, as shown in **figure 6**.

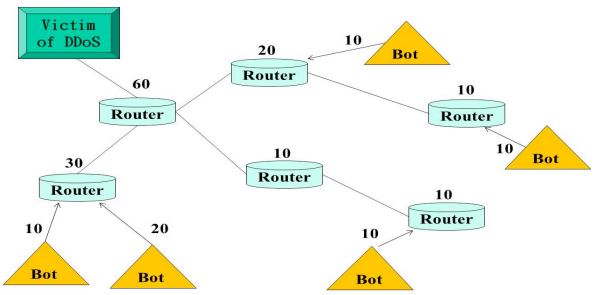

Figure 6. Impact counting of the DDoS defense.

Secondly, all those incoming edge routers having the counter value greater than a threshold will receive a command from the network management system to enlist an ACL to carry out the block operation. Therefore the impact of the DDoS attack can be greatly reduced.

Because the DDoS bots may change over time, the high impact edge routers may have to change as a consequence. Thus the counters will be reset after a certain period of time. Meanwhile the ACLs will be tracked to avoid duplication and eventually be expired after a predefined time in order not to influence the router performance.

In practice, although the defense mechanism can be fully automatic, there are still some issues. If the ACLs are implemented against the source IPs of the DDoS origins, the mechanism itself will be effective and precise, but due to the fact that most DDoS source IPs are fake, it may generate a large number of ACLs as a result, thus causing a significant burden to the backbone routers. On the other hand, if the ACLs target the destination (victim) IP and port, only one or a few ACLs are needed. While the extra burden is minimal or even marginal, it may block legal flows at the same time, causing false positive problem. To overcome this dilemma, we still need more DDoS experience and data to fine tune our defense mechanism design.

Another issue of the automatic defense is that the connecting institution may not want their network flows being artificially filtered without their explicit consent. In order to solve this issue, a web system needs to be implemented to allow the connecting institution to manually activate the DDoS defense mechanism against their selected flows.

We consider inviting all connecting institutions to help improving the system by notifying the TWAREN NOC about their DDoS incidents. By investigating the DDoS flow history, our system can be further enhanced.

3. Conclusions and Future Work

After finish developing the filter module and the rule import module, we shifted our main focus on developing the analyzer module of the TWAREN defender cloud. By collecting the layer-3 interface IP information and the layer-2 neighbor discovery SNMP MIB data, the topology of the network can be constructed. The full path of any flow can then be chained by

joining flow records against the topology database, thus making the back tracing of the DDoS flows to their entrance possible. By accumulating and resetting counters on the network nodes and links along each flow, the edge routers can be sorted according to their impact to a given DDoS attack. ACLs can be installed on those high impact edge routers to help mitigating the DDoS impact to the victims.

References

1. TaiWan Advanced Research and Education Network. (http://www.twaren.net/).
2. Ming-Chang Liang; Hui-Min Tseng; Shin-Ruey Hsieh; Wei-Jie Liau; Jyun-Jie Chen; Te-Lung Liu; Jee-Gong Chang. The Design and Implementation of the Defender Cloud on TWAREN Backbone. *APAN 32th Fall Meeting at New Delhi*, India, 2011.
3. Ming-Chang Liang, Meng-Jang Lin, Pin-Hsuan Chen, Shin-Ruey Hsieh, Jyun-Jie Chen, Tsung-Han Lu. The modular architecture of the TWAREN backbone defender cloud. *APAN 34th Fall Meeting at Colombo*, Sri Lanka, 2012.