

DDOS防禦系統研究與實作專題

背景

背景

- 本專題主要是在優化與完善實驗室開發之DDoS防禦系統，因為現今企業對於DDoS攻擊的防禦，大多是採取加強硬體設備的方法，而此方法的成本過高，部分企業無法實施，並且一旦惡意人士成功阻斷服務，則服務將會立即中斷。
- 此系統擁有最後一道防線：在不幸被成功攻擊後，系統能透過遷移伺服器來保持服務繼續運作，並且透過系統的客戶端，讓新舊使用者都能繼續與服務伺服器保持連線。

系統簡介

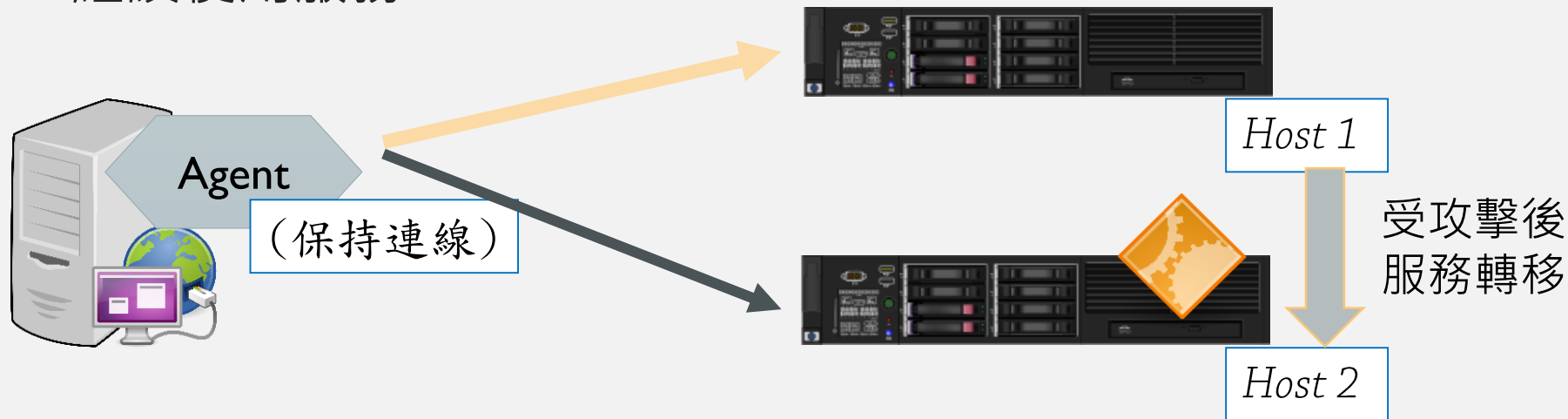
系統簡介

- 系統分成兩個部分：
 1. 保護服務伺服器的系統伺服器端(host)
 2. 使用者執行的系統客戶端(agent)



系統簡介

- 當系統伺服器偵測到DoS/DDoS攻擊後，會啟動防禦機制。伺服器端會在確保當前使用服務的系統客戶端服務不中斷的情況下，將伺服器轉移到其他IP位置並繼續運作，避免再次被攻擊。
- 因此，服務伺服器即使遭受DDoS攻擊，也能轉移到其他主機繼續提供服務，並且使用者也能不受攻擊影響，繼續使用服務。



專題實作——支援WINDOWS客戶端

目的

- 原系統只支援Linux客戶端，為了使其能在Windows上運作，我們透過PyDivert實作iptables、nfqueue，最後完成Windows版本之系統客戶端。

linux系統的client agent

以pydivert改寫原程式由iptables負責處理的部分

windows系統的client agent

方向

- nfqueue → 處理封包
 - iptable → forwarding
- Pydivert

以Pydivert取代原本nfqueue、iptables負責處理封包及forwarding的部分。

架構



程式結果

- 成功使windows上的客戶端也能在伺服器被攻擊而轉移後繼續保持連線

程式結果

- 在服務由server1轉移至server2後，Windows client agent顯示migration完成：

```
*** service_port ==> 22
*** dst_ip ==> 140.115.59.14
*** dst_port ==> 10002
----- MIGRATION!!! -----
.
Sent 1 packets.
Begin emission:
..Finished sending 1 packets.
.....*
Received 16 packets, got 1 answers, remaining 0 packets
.
Sent 1 packets.
----- Update connection successly !! -----
*** checksum ==> 0
*** ack ==> 0
*** local_ip ==> 140.115.142.85
*** service_ip ==> 140.115.59.253
*** service_port ==> 22
*** dst_ip ==> 140.115.59.14
*** dst_port ==> 10012
```

程式結果

- 此時，在libvirt內下ls指令，成功獲得回應，代表server在轉移後仍提供client服務：

```
Microsoft Windows [Version 10.0.19041.173]
(c) 2020 Microsoft Corporation. All rights reserved.

C:\Users\jerry>ssh libvirt@140.115.59.253
libvirt@140.115.59.253's password:
Welcome to Ubuntu 14.04.5 LTS (GNU/Linux 3.19.0-80-generic i686)

 * Documentation:  https://help.ubuntu.com/
Last login: Tue Apr 14 20:39:25 2020 from 140.115.142.85
libvirt@libvirt:~$ ls
exploit  nohup.out  sock_server.py  test.tar  udptest.py
exploit.c  q          task           timing
jyundtest  scptest   test2.py       timing.c
libvirt@libvirt:~$ |
```

程式結果

- 轉移後，原本的host停止接收到封包：

```
flags      =  
frag       = 0L  
ttl        = 60  
proto      = tcp  
chksum     = 0xa358  
src        = 140.115.142.85  
dst        = 192.168.0.2  
\options   \  
###[ TCP ]###  
  sport    = 7971  
  dport    = 8888  
  seq      = 3333256469  
  ack      = 1085616106  
  dataofs  = 6L  
  reserved = 0L  
  flags    = A  
  window   = 8192  
  chksum   = 0xb51e  
  urgptr   = 0  
  options  = [('NOP', None), ('NOP', None), ('EOL', None)]  
DROP
```

程式結果

- 新的host開始接收封包：

```
proto      = tcp
chksum     = 0x37c0
src        = 140.115.142.85
dst        = 192.168.1.2
\options   \
###[ TCP ]###
  sport     = 7971
  dport     = 8888
  seq       = 3333256649
  ack       = 1085616818
  dataofs   = 8L
  reserved  = 0L
  flags     = A
  window    = 259
  chksum    = 0xc40a
  urgptr    = 0
  options   = [('NOP', None), ('NOP', None), ('Sack', (1085616742, 10
85616818))]
```

程式結果

- 根據圖片結果顯示，在經過live migration後，windows client仍與server保持連線。

專題實作——強化偵測

目的

- 原本系統僅針對flood攻擊進行偵測，現透過限制傳輸速度等慢速攻擊防禦機制，實作更廣泛之DDoS攻擊偵測。

目前規劃防禦之攻擊類型

1. Slow headers (slowris) : Header故意不傳送 “\r\n”
2. Slow body (SlowHTTP POST、r-u-dead-yet) : 卡住body不傳送
3. Slow read : 利用滑動窗口的宣告

目前規劃之防禦方法

1. 超時則斷開連線
2. 限制HTTP慢速連線的連線數
3. 限制使用Range-byte header

THE END