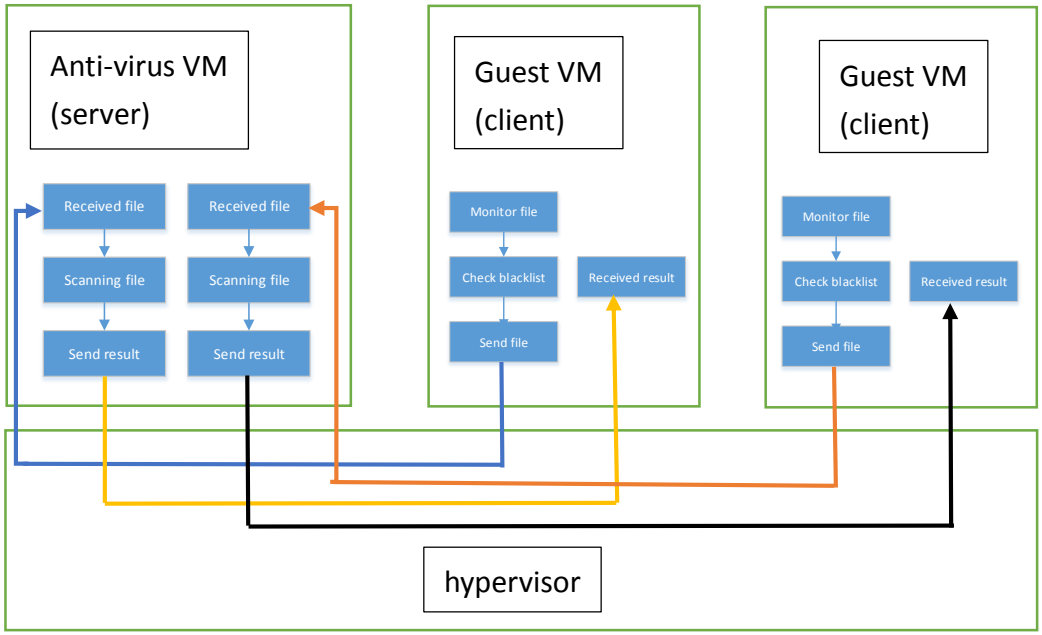
Chun-Kuei Huang
John Bambenek
Security Lab
05/07/2015

Anti-virus agency for virtual machine on a host

1. Abstract:




2. System analysis
   - Saving disk space – only the server maintains the anti-virus database rather than every machine keeps a same copy of the database.
   - Saving updating resource – only the server runs the anti-virus service, so not every machine has to keep tracking whether there is a new virus definition released online.
   - Using latest secure mechanism – the server is always online updating to the latest version. When a client starts after a long time in power off status, it still uses the latest anti-virus database.
   - Increasing the local network usage – the local network is usually idle, transmitting the file which needed be scanned makes a better usage rate. The speed is very fast due to they are using the same physical network card.
   - False positive on client side with md5 – the client use md5 value to check the blacklist but if a md5 collision occurs, the client may retreat a clean file as a threat.

3. Server implementation

The server locates at ip 192.168.111.131 and listens on port 7000 to wait for a client connect to it. When a client establish a connection, the server forks a new process and starts receiving the file and save to server disk with filename “scan_pid”. The pid is to distinguish the files from different connection. After saving the received file to the disk, the server start to do anti-virus with clam-av. From checking the first line of the result given by clam-av, the server will know whether the received file is a threat or not. It will send back a message to client. A “safe” message represents the file is clean, and a “danger” message represents the file is a threat. At the end, the server remove the received file or the disk will blow up.

4. Client implementation

The client uses a tool called minispy(https://code.msdn.microsoft.com/windowshardware/Minispy-File-System-97844844) on windows to help monitoring if a file is accessed by other. The implementation on client side is adding function to mspyLog.c which offer a device list that is accessing file and a blacklist creation for local usage. The blacklist works with md5, a user on client side can define a virus by using addblacklist which is done with calculating its md5 value and insert the result to blacklist.txt. When a file is accessed, the client first check if the md5 value of the file is in blacklist.txt. If yes, log the information into warning.txt. If no, send the file to the server and wait for the response. If the client received “safe” from server, it is all good and the client does not have to do anything. If the client received “danger” from server, it log the information into warning.txt.