F-INSIGHT Feb 16, 2013

ABSTRACT

1. Objectives

Almost all the incidents including pilfering administrator privileges through network attacks do not show any sign of compromising until the victim realizes the damage against the crucial information assets and that company secrets have leaked out.
To probe and respond to these attacks, victims and law enforcement officers investigate the attackers whereabouts by assorting the information found on the hard disk drive, networks and security devices and analyze it.

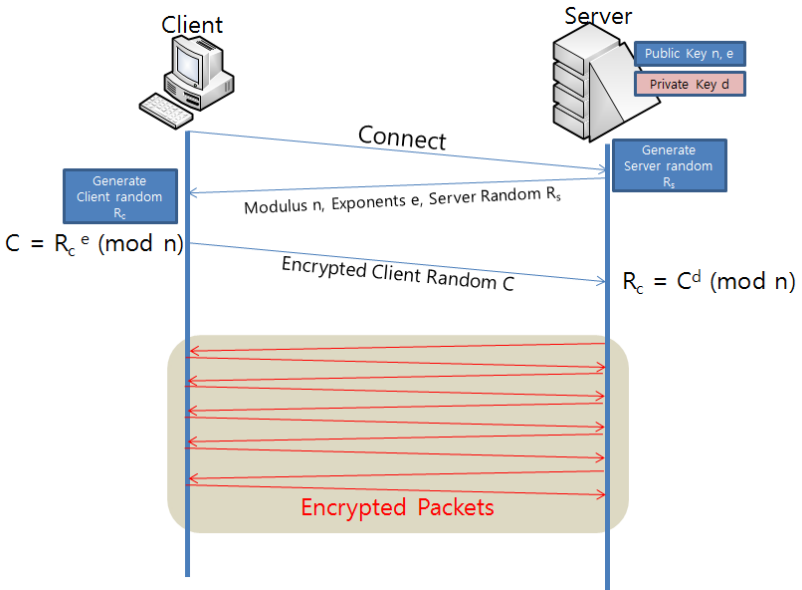
Investigators summarize these processes in a report with many technical bases about the information which can be used to assume the attacker's hacking activities. This report is used to support the investigators' arguments against the arbiter with minimal expertise, in the whole process of the persuasion. This kind of report which is full of technical terms is considered to be hard to understand to the decision maker. If the murder's activities in the homicide case and the whole process of illegally duplicating company's secrets in the espionage case, is recorded in the Closed Circuit Television(CCTV) and can be presented to the decision maker, then he or she can easily understand the whole process of the case and determine what to do next. It is quite obvious that the decision maker can much more easily accessible to the case by watching the recorded screens than reviewing the reports with full of technical jargon.

Up to the present, the whole contents of the Internet communication have not been always captured like the way of CCTV. Even though when we had anticipated or found out the incident and captured RDP and SSH communication packets, there is no easy way to analyze the content of the captured packets. With this reason, to propose an efficient method of visually illustrating the activities of the attacker who have compromised an network and performed illegal activities in the network, we started this research after determined that we can assist the prompt and efficient decision of the arbiter by replaying the activities of the attacker who had connected to the compromised system remotely with the administrator's privilege after breaking into the victim network.

2. Packet replay protocol

After compromising network, the attacker tries to connect to the victim system after acquiring administrator's privilege in one way or another. In the case of Linux or Unix system, the attacker may initiate encrypted connection to the secure shell(SSH) server using TCP/IP port 22 or remote desktop(RDP) server using TCP/IP port 3389, in the case of MS windows operating system. The RDP server also uses encrypted communication.

When we see the RDP communication, as you can see in the right picture, the client receives the public key and 32 bytes long random number from the server, after initiating the communication, and after that, the client generates 32 bytes long random number and transmit it to the server after encrypting it with the public key from the



server. Then the server decrypt the encrypted random number from the client using the private key. After that, both the client and the server communicate each other using the encrypted packets after generating RC4 encryption/decryption key using the random numbers from the server and from the client.

In the process of initializing phase of the encrypted communication, we can secure the public key and the plaintext random number from the server, and the encrypted random number from the client. After finding out the private key from the server, we can generate RC4 encrypted/decrypted keys by ourselves. That means that we can decrypt the whole communication contents. It seems to be pretty obvious, however, it is one thing to be able to decrypt the communication contents and it is another to fully understand the contents of the decrypted one.
It is absolutely possible that we can capture events from input devices, like from the mouse and from the keyboard, so as to convert the corresponding screen changes into the data that the investigator can understand and analyze it, by referring to the RDP specifications and by the Rdesktop program which is one of the open source projects. It is practically, however, hard to implement numerous communicational environments and many complicated procedures for it, unlike SSH protocol.
That is why we determine that it is quite efficient to reuse mstsc.exe program which is produced by Microsoft Inc. to be used as the RDP client software.

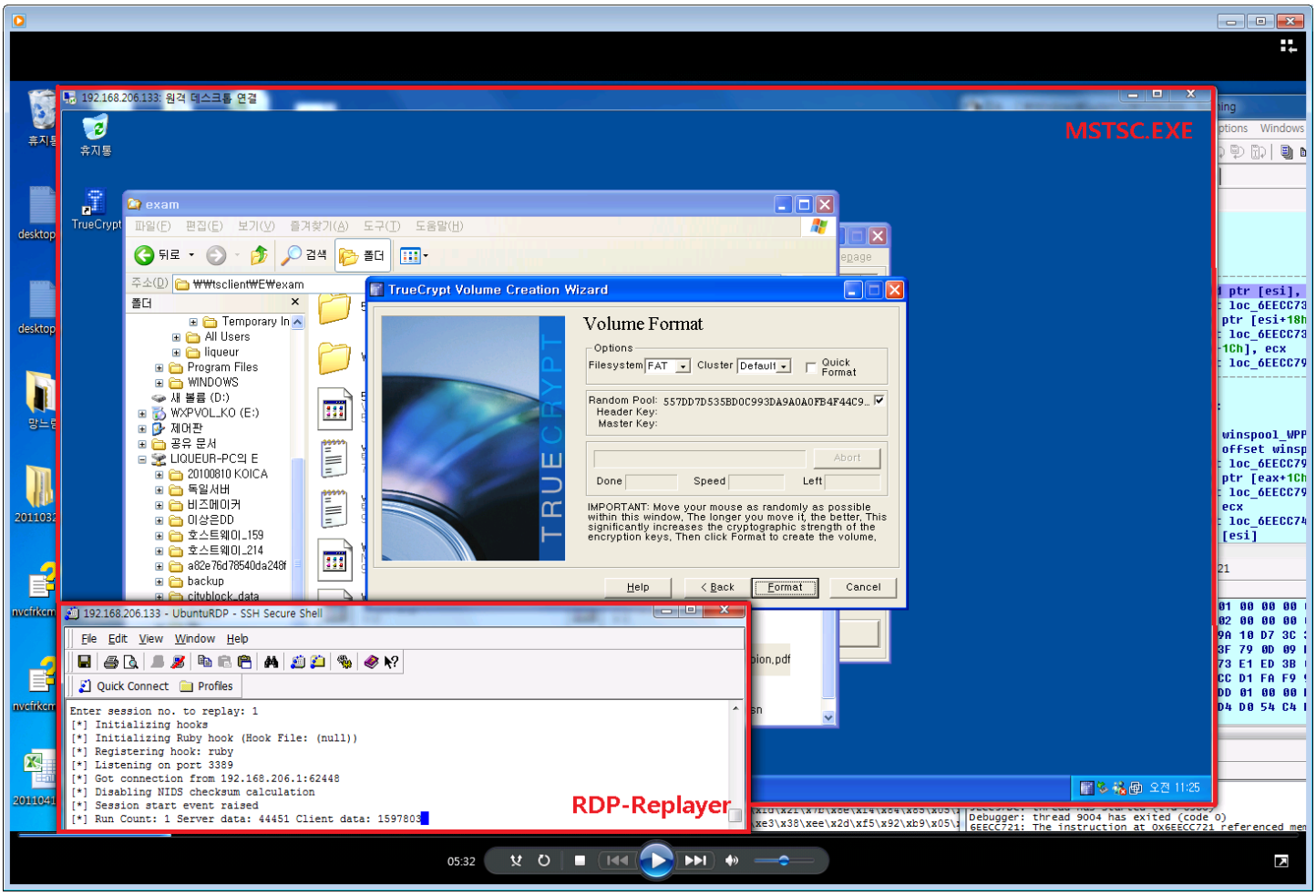
We can replay RDP packets only by retransmitting TCP stream to the RDP client software, mstsc.exe. In this case, we don't need to have full packets, but we only need the initial RDP connection packets to replay saved TCP stream.

In replaying attacker's activity in the form of movie, there are only 2 prerequisites, RDP packets to reply and Private key of the server.
We can find the private key of the server at the lower 64 bytes of the "L$HYDRAENCKEY_28ada6da-d622-11d1-9cb9-00c04fb16e75" value of the LSA secrets, gathered.
The captured packets are encrypted by RC4 key value generated by the client side random key and the server side random key. In replaying the packets, the server side random key is transferred to mstsc.exe program just like the original one, but the random key value of the client side is randomly generated by the mstsc.exe program, so we have to input the original random key value of the client side to the mstsc.exe by force.

We have to find out the original random key value of the client side by decrypting the encrypted random key value of the client side stored in the packet by RDP-Replayer.

The picture below is the screen capture of the successful implementation of the RDP-Replayer.

Usage: RDP-Replayer "pcap filename" "private key filename"



3. RDP-Replayer implementation result and thing to be improved

In case of RDP protocol, when we have communication packets and relevant private key, we can easily identify attacker's activity only by appropriate replaying program without searching for the evidences in the hard disk drives and numerous network packet based records. This methodology is applicable not only to the RDP protocol, but also SSH protocol which is much simpler than RDP protocol. It is also applicable to the similar protocol like VNC. We are absolutely sure that we can stimulate digital forensic researcher to develop replayers of the other kind protocols.

It is quite inconvenient to input the random number of the client side forcefully to the mstsc.exe in every implementation. If the RDP-Replayer can perform the encrypted communication by using encryption and decryption key generated by the random value from the mstsc.exe program, after decrypting the encrypted data by the RDP-Replayer, we can much more easily analyze the packets because we don't need to input the random value forcefully to the client program, mstsc.exe. And in that case, we do not need to be equipped with fully understanding of the embedded technology.

Most of the captured packets transmitted from the client to the server, are the mouse movements and keyboard input values, so we can identify the invisible values like passwords when we print out the packets, synchronizing the screens transferred to the attacker with actual time.

The replayed screen by mstsc.exe is played by the replay time of the packets, so it is possible to replay the screens which are exactly same with the attacker's screen, when we match the packet replay

interval with the time of the attacks. We may save the replayed screen into the movie file and present it to the investigator.

We tried to use Rdesktop and xvidcap program to save the screen into the movie file, but in many cases, we couldn't replay the packets gathered by mstsc.exe with the Rdesktop program. If we find the reason for the failure, we will be able to record the packets into the movie file.

If we place the RDP-Replayer and Rdesktop combination at the IDS to reassemble the TCP Stream, then we can use the program just like the CCTV to monitor the RDP session connected, by replaying it to the screen using appropriate application, like Rdesktop.