Richard Kennedy

Derek Zhou

Professor Babu

Compsci 182

Duqu: Case Study Summary

Our case study was about Hacking: Evolution and Implication. We decided to concentrate on the Duqu malware. Duqu is a remote access trojan. It shares many lines of source code with Stuxnet which led researchers to conclude that it shares the same authors or at least authors that have access to Stuxnet's source code. However, Duqu is a highly targeted sophisticated virus. It is designed to infect and steal data from manufacturers of industry control systems, which in turn is used to attack such systems. The difference between Stuxnet and Duqu is just that. Stuxnet was made to attack and sabotage such systems and Duqu was made to gather information. Duqu installs keyloggers and network enumerators to steal information. As a targeted sophisticated virus, it is designed not to self replicate or propagate and is configured to remove itself after 36 days.

Duqu is installed onto a user's computer through an email attachment. It appears as a harmless legitimate .doc document that triggers the exploit. This special Word document contains an undisclosed zero day kernel exploit that is based on Microsoft Win32k TrueType font parsing engine.

This exploit loads the shellcode and runs an .exe file which in turn decrypts the driver and launches the silent installer. The driver file (.sys) is installed and the shellcode then executes the driver. Upon execution of the driver, the installer (.dll) library is injected into the services.exe process (which is a legitimate process started at upon bootup). After injection, the services.exe process has the installation code which is composed of 3 components: the main dll, the load point driver, and the config file. Once the installation code is loaded onto the user's computer, they are decrypted and run as a driver. Because

the driver is loaded upon system start and has direct access to the kernal as a driver, it is both silent and dangerous. Finally, the shellcode will zero itself out and only the main dll, driver, and config files remain on the computer which make duqu's existence untraceable.

After data is stolen, it is transferred to the command and control servers as a lightly encrypted jpeg file. Unlike most worms, Duqu uses a peer-to-peer command and control model. Each computer that is infected connects only to the computer that infected it and to the computers that it infects. The command and control servers, thus, are only connected directly to a minority of the set of infected computers. This architecture makes it very difficult to trace to the C&C servers. Another advantage of the peer-to-peer model is that it allows Duqu to infect computers in a “secure” zone. Many networks have a “secure” zone where there is heavy monitoring and less secure zones, that freely interact with the internet. Duqu can directly infect the less secure computers, and through the peer-to-peer model, use that computer to infect computers in the “secure” zone.

Another advantage of this model is that those running the command and control servers can control the spread of the virus. If the virus freely replicated, like most, that would make the virus more likely to be found. Instead each network and computer that infected is chosen by the C&C server operators. When a computer is infected, that computer collects the information of all the computers connected to its local network and forwards them to the C&C servers. The server operators then instruct the infected computer to infect the computers in the network, which subsequently forward the information of computers in their networks. In this manner, the Duqu virus is able to spread into an enormous range of systems, while being able to do so with the utmost precision.

So far only 3 command and control servers have been found. All these computers simply forwarded logfiles to other command and control servers, but they still provided value information. Even though

these computers have all been scrubbed, the attackers made some mistakes deleting the evidence, that allowed analysts to gain information. Their first mistake was to not delete the emails intended for the root user. This means that the analyst could discover what information was being forwarded to root command and control server. The second mistake was not to overwrite the unused space with zeros. The operators of the C&C servers deleted not only the pointers to files, but the files themselves; however, they forgot to look at the unused space where files were written on occasion. Doing binary analysis the analysts were able to better understand how the Duqu virus worked.

One point of contention is how an illegitimate driver is verified and downloaded as a legitimate driver. We can attribute this to a digitally verified signature given to the driver which was stolen from C-media. This is the same company which Stuxnet's driver's signature was from. Similarity with Stuxnet suggests that the private key from the company was compromised. This means that this was a physical intrusion rather than virtual.

Regarding the question brought up in class about public key cryptography which is the main cryptographic system used, pkc is the process of using a public and a private key to ensure security. The process being, the person with the private key writes some data (in this case, a driver) and then encrypts it using their private key, then sends it out. Then for those people who have the public key they can verify that the file originates from the person they know it to be from (the person with the private key), then they can use the public key to decrypt the message of the private key. In the case of Duqu, the private key was compromised and thus content that was not from JMicron or C-Media. The fact that this was due to a physical intrusion leads us to conclude that no matter how safe and secure the internet or data is, physical security will always be vulnerable.

Like Stuxnet, Duqu is very controversial, because the skill and exposure of the code drives us to

believe that either a big company or a nation is responsible for the virus. This leads us into the ethical and legal discussion of cyber warfare. With Stuxnet, we discovered the target to most likely be Iranian missile silos. In this case Duqu seems to be targeted at PLC makers, thus setting the stage for the next Stuxnet-like attack. There are a whole range of ethical issues that surround cyber warfare. The first is that it is anonymous. It is nearly impossible to whole people responsible for cyber attacks. Thus, it is uniquely problematic if the Stuxnet-like code landed in the hands of a terrorist organization. How would the US hold someone responsible for the attack? No one so far has taken responsibility for Stuxnet attack itself, though Iran has publically stated that it was carried out by the US. Similarly cyber attacks on the US by China have never been publically accounted for, though the US insists that China is to blame. Another major issue with Duqu and Stuxnet, is that their type of program could easily be used for a civilian attack. PLCs are used to clean water, operator elevators, electrical grids and a whole host of other civilian purposes. Although, so far the only attack was carried out against an Iranian nuclear site, civilian PLCs are much more vulnerable to an attack that could kill millions. Yet, there seem to be many good benefits to cyber warfare. Since the focus, so far, has been on industrial sabotage and intelligence gathering, it is much more preferential than the alternative, which is traditional warfare which kills millions and does incalculable psychological damage. Whether for better or for worse, there seems to be no doubt that the way of the future is cyber warfare. The US, China, and Iran, have spent millions of dollars on cyber warfare. Duqu is just the beginning of a transformation of the way nations gather intelligence, fight wars, and conduct industrial sabotage.