Evidence of Advanced Persistent Threat:

A Case Study of Malware for Political Espionage

Frankie Li, Anthony Lai, Ddl Ddl
Valkyrie-X Security Research Group
{ran2,drakfloyd,dll}@vxrl.org

Abstract

A political figure in Hong Kong continuously receives spear-phishing emails that encourage clicking on shortcuts or opening attachments with file extensions, such as .pdf, .doc(x), .xls(x), .chm, and so on. He suspects that such emails were actively sent from seemingly known parties during the pre- and post-election periods. The emails and samples were sent to us for investigation, and two nearly identical samples were chosen for the case study. These malwares appear to be the first Advanced Persistent Threat (APT) incident to undergo detailed study in Hong Kong. APT is defined by MANDIANT as a cyber attack launched by a group of sophisticated, determined, and coordinated attackers who systematically compromise the network of a specific target or entity for a prolonged period. The malware performs the following functions similar to those of "Operation Shady RAT", it attempts to hide itself from known anti-virus programs, downloads and executes additional binaries, enumerates all file information in the hard disk, gathers email and instant messaging passwords from victims, collects screen captures, establishes outbound encrypted HTTP connections, sends all gathered intelligence to a Command and Control, and deletes all temporary files of the collected information from the victims' machine after uploading. The forensic findings lead us to believe that APT is a real threat in Hong Kong.

1. Introduction

A political figure in Hong Kong continuously receives spear-phishing emails that lure him into clicking on shortcuts or opening attachments with file extensions, such as .pdf, .doc(x), .xls(x), .chm, and so on. He suspected that these emails were actively sent from seemingly known parties that were subsequently identified as fake senders during the pre- or post-election periods. He believed that most members of the Hong Kong "Pan-Democratic" party receive these well-crafted emails whenever influential political events occur, such as the June 4 Tiananmen Square Massacre protest or the annual Hong Kong "Big Protest" on July 1. He also suspected that these malwares were sent from a well-funded party for the purpose of political espionage. He and his colleagues provided several suspected emails and specimens for analysis.

Dynamic and static analysis revealed the important functions of the malware and identified the intelligence collected from the target system. We further investigated the techniques used by the malware for hiding and infiltrating the detection mechanism.

In this paper, studying the modus operandi of attackers confirmed that Advance Persistent Threat (APT) is "not simply a buzz word, created by marketing folks to sell more security products and services" [2]. It is a real category of attacks in Hong Kong, especially those meant for political espionage.

2. Related Work

2.1. An Overview of Advance Persistent Threat

According to Andress (2011) [2], the term "APT" may have originated from several sources. However, two instances in particular stand out as the most likely sources. The first is the term used by Mike Cloppert from the USAF's 8th Air Force in 2006 [3], and the second is a detailed research report published by Mandiant (2010) [4] after "Operation Aurora" [5]. In the present paper, the APT definition proposed by MANDIANT was refined. Here, we consider an APT as a cyber attack launched by a group of sophisticated, determined, and coordinated attackers who systematically compromise the network of a specific

978-1-4673-0034-6/11/$26.00 ©2011 IEEE

target machine or entity for a prolonged period. Traditionally, "Persistent" refers to the fact that APT-type malware maintains the exploitation over a period of time. Actually, this is the same tactic used by most malware, for example ZeuS or Zbot. However, the characteristic of persistent efforts of launching APT-attacks to comprise a system or an organization's network should not be overlooked. If we place more attention on this attack strategy APT-attack may be detected at its early stage, for example at the time of spear-phishing email is identified.

The unique motivation, techniques, and tenacity shown by the attackers suggest that these are not from normal "hackers" looking for potential financial gain; they are in fact, a well-funded and organized group of professional attackers, "motivated by massive hunger for secrets and intellectual property"[4].

2.2. Previous Research Findings on APT

In March 2009, the SecDev group in Toronto released findings regarding GhostNet [9]. In this report, China-based hackers have been identified the ones who orchestrated the attack and systematically infiltrated the Dalai Lama's office and the Falun Gong community.

The Trojan horse malwares were distributed through carefully crafted emails (so-called spear-phishing emails), and the attack was traced back to six Command and Control (C&C) servers, five of which were located in different provinces in China and one in Hong Kong. Furthermore, the report stated that "only 11 of the 34 anti-virus programs provided by Virus Total recognized the malware embedded in the document, and the malwares were either identified as Microsoft Document exploits or as general droppers."

In 2010, MANDIANT published a report on APT [4], wherein it has been defined as the "Exploitation of Life Cycle;" this work also discussed the common mechanisms employed by malwares. The APT is usually distributed by spear-phishing emails, with .zip or .rar file attachments containing any of the following:

- a CHM file containing malware;
- a Microsoft Office document exploit; or
- some other client software exploit, such as an Adobe Reader exploit.

After intrusion, the malware may or may not open a backdoor and obtain user credentials by installing various utilities to ex-filtrate intelligence from the target. Several common characteristics of the APT have also been described in that work, including smaller file size, packing mechanism, and dynamic-link library (DLL) injection feature. These characteristics create barriers to detection.

A month before the submission of the present paper, McAfee released a report entitled "Revealed: Operation Shady RAT" [1]. The report stated that "targeted intrusions were found in more than 70 global companies, government, and non-profit organization during the last 5 years. Significant size and valuable intellectual property and trade secrets have been compromised and the great majority of victims rarely discover such intrusion." The organizer of such large-scale attacks has not yet been identified. However, clues have been given as to the identity of the state actor behind such operation, due to the unlikelihood of earning commercial benefits from such attacks.

3. The Analysis

3.1. Methodologies and Lab Environment

The samples were analyzed inside a host-only network, VMware virtual machines.

We adopted the analysis methodology initially proposed by Zelter (2007) [6] and further refined by Brand and Woodward (2010) [7]. The method is basically iterative and recursive, alternately using dynamic (behavioral) and static (code) analysis techniques to extract the full functionality of the executable.

The malware was collected from a source that the Chain of Custody can maintain, i.e., the Collection Stage. After checking strings contained inside the binary (which can be classified as the first step of static analysis), the preliminary information passed through the dynamic analysis process as input. The findings or output on the malware behavior were then used as further input ("influences") for the detailed code analysis. More information were gathered to understand certain malware functions using a debugger to step through the execution by allowing the malware to connect to the Internet, an act that was subject to certain legal and ethical constraints. The malware may be multi-staged in nature; thus, we repeated the Dynamic Analysis Process. Collected output was then fed back to the code analysis process to generate more "influences" until a firm conclusion was reached and a report was produced.

3.2. An Overview of the Samples

The first email and specimen was received on June 7, 2011. The email contained the subject "Democracy Depot meeting" and was supposedly sent by first_name.p0on@<org_name>.org.hk (Note: actual names were withheld). The email attachment was also called "Democracy Depot meeting." According to the targeted victims, they arranged actual meetings before receiving this suspicious email. The attachment was not protected by a password and contained two files "Agenda.doc" and "Democracy Depot Meeting Minutes.doc.lnk," which were generated after unzipping the attached file.

The second email and specimen was received on June 14, 2011, which was supposedly about news of a riot that occurred on June 10, 2011 at a county near Guangzhou City in China. The email was supposedly sent by a political group. The email attachment was an unprotected .rar file called "廣州新塘事件.chm" (lit. "Guangzhou Riot Incident.chm"). Un-raring generated a chm file, which displayed a help file when double-clicked.

3.3. The Dynamic Analysis of the Dropper

The "Democracy Depot Meeting Minutes.doc" is a shortcut, and its "Property" field (inside the "Target" field) contains the following command:

```
%comspec% /c start Agenda.doc&for /f %j in
('dir /b /s %temp%\Agenda.doc') do start "" %j
```

Double clicking "Agenda.doc" invoked Microsoft Word or Wordpad and displayed some cryptic characters. However, clicking on the "Democracy Depot Meeting Minutes.doc" short-cut would start up a command prompt to execute the binary Agenda.doc. Then, the chm file was decompiled, and a binary called "ABC.exe" was generated.

Both samples were unpacked using UPX 3.03w with the –d (decompress) switch, and then executed under the infected box of the virtual machine. These malwares were compared using Bindiff [10], and were found to be executable in nearly the same manner. Only the analysis results of "Agenda.doc" were produced to avoid duplication.

The dynamic analysis revealed the following results:

- The "Agenda.doc" is a binary.
- The file copied itself to %AppData%\IECheck.exe.
- The file made an almost "identical" version at %AppData%\ws2help.PNF
- The file moved and renamed itself to %MyDocuments%\My Pictures_@D.tmp.
- The file created %AppData%\msvcr.dll.
- The file modified %System%\netstat.exe.
- The file created and then deleted %Windows%\inf\1.txt.
- The file created %System%\2525.
- "explorer.exe" was created %System%\ipsetstap.dat.
- "explorer.exe" was created %Programs%\Startup\Internet Explorer Security Check.lnk.
- "explorer.exe" tried to connect test.3322.*.cn, 172.16.0.61 and 202.*.*.* by TCP port 8080.
- "explorer.exe" was injected with the file "msvcr.dll" at the base address = 0x10000000 with entry point of 0x1001F000 with image size=00039000.
- "explorer.exe" sent encoded traffic to remote IP of 202.*.*.* once it was connected with the Internet.

3.4. Static Analysis of the Dropper

Based on the above findings, we suspect that "Agenda.doc" was the dropper (the "Dropper") that created and injected the malicious "msvcr.dll" (the "Malicious DLL") into the process of the "explorer.exe" that, in turn, sent out the malicious encrypted network traffic to the C&C.

3.4.1. Stage One. First, the file checks if it has been initiated from a command line with "Update" or "Special" argument.

Second, the file checks if the victim installed several famous Chinese anti-virus programs such as "卡巴," "Kaspersky," "360安全卫士" and "瑞星," by enumerating the registry key at "SOFTWARE\Microsoft\windows\CurrentVersion\Uninstall."

Immediately after checking, the Dropper tries to create a mutex of "VistaDLLPro RUNNING" to avoid duplication of malware installation on the victim's machine. This is the same technique used by some notorious malwares, such as ZeuS, to mark their presence in the system.

Then, the Dropper copies itself as %AppData%\w2help.PNF at 0x00403DA3.At

0x00402BC9, the Dropper performs its first important function and creates the Malicious DLL. The file calls a few APIs, incuding FindResource and LoadResource, to locate the encoded Malicious DLL in the resource section at offset of "VISTADLL."

At 0x00403680, the file loads 0x1C000 (114,688) bytes from the resource section starting from offset at VISTADLL to 0x0040F1F0. The decoding routine is executed at 0x0043692 to 0x004036AC, and the Malicious DLL is completely decoded at memory location 0x0040F1F0.

After decoding, the Dropper packs the Malicious DLL with its own packer and proceeds to write the packed file "msvcr.dll" at 0x00402CAD.

At 0x0040181A, the Dropper makes a copy of %System%\netstat.exe using a random, generated file name. In our example, the named file was was %System%\2525. The Dropper also makes a copy of %System%\SFC_OS.dll at %Windows%\inf\1.txt. Then the "netstat.exe" is patched after calling a hidden function #5 inside "SFC_OS.dll" to bypass the Windows File Protection mechanism.

The Dropper then injects "msvcr.dll" to the running process, "explorer.exe," by calling CreateRemoteThread API. At this point, we assume that the malware author used such a name because it resembled the standard C-runtime library of "msvcrt.dll," which is also used by explorer.exe.

Before termination, the Dropper moves and renames itself to %MyDocuments%\My Pictures_@D.tmpas system file.

In the meantime, we gathered another sample with a .chm file extension and found an embedded executable entitled ABC.exe. A different binary [10] between ABC.exe and Agenda.doc was made, after which we found that their structures were nearly the same. Thus, it can be concluded that both files came from the same generator, but that a number of changes and updates were made before release.

3.4.2. Stage Two. The "msvcr.dll" was packed by a proprietary packing routine. Thus, we have to manually unpack it for further analysis. The next step was to patch the Dropper so that it must inject the "msvcr.dll" into the debugging "explorer.exe" process. After injection of the Malicious DLL, we let the Dropper run until its termination.

The injected "msvcr.dll" checked the existence of the mutex "VISTADLL IS RUNNING"at address 0x10009349. Then, a unique machine ID was generated by referring to the network adaptor information and hard disk volume serial number at 0x100009455. The wording "STAPPro" and "hk0603" were added at the end of this machine ID. Then these strings were encoded at 0x10009594, and written and saved as a newly created file at C:\WINDOWS\system32\ipsecstap.dat.

Afterwards, the injected thread invoked an API call to GetPrivateProfileString to retrieve strings from a non-existent initialization file at %Windows%\msip.ini. At 0x100099FA, an "Assiant Thread" (sic, found in "msvcr.dll") was created to execute a function at 0x10008E57. This thread created a mutex named "VistaDLLPRo Want Wood To Exit?" and ran a loop to perform several file activities.

The injected thread then tried to contact a few non-resolved DNS names (test.3322.*.cn, 1.test.3322.*.cn, 2.test.3322.*.cn, 3.test.3322.*.cn, and 4.test.3322.*.cn) and a non-routable IP address (172.16.0.61). After various trials, the injected thread contacted the single valid IP address of 202.*.*.* using the TCP port number 8080. Once the socket was created, the thread sent out several encrypted network traffic.

Without enabling the network connection to the actual IP address of 202.*.*.*, the thread ran in an infinite loop and waited for the response from the C&C.

3.5. The Dynamic and Static Analysis of the Droppee

We allowed the malicious DLL to contact the actual C&C to gather more responses and perform another dynamic analysis session. We also turned on Wireshark [11], and found that the injected thread successfully connected to IP address 202.*.*.*.

After connecting to the C&C, the injected thread jumped into function at 0x1000C752 to gather several basic information by calling APIs, including the following: GetVersionaExA, GetComputerNameA, GetUserNameA, GetLogicalDrives, GetDriveTypeA, GetDiskFreeSpaeExA, and GetACP. The registry key "SYSTEM\CurrentControlSet\Control\ProductOption" "HARDWARE\DESCRIPTION\System\CentralProcessor\0" was also enumerated. All collected information was temporarily kept in memory for further encryption.

This memory address was passed to an encryption function at 0x1000D438. The encryption algorithm block was found at 0x1000D44A to 0x1000D472.

Based on this encryption algorithm, we wrote a script that decrypted part of the communication. We found that the Malicious DLL first sent a standard HTTP request with the encoded unique machine ID, and waited for the standard HTTP response from the connected C&C. Then it sent all collected information by encryption HTTP traffic to the C&C.

Three binaries (i.e., fvcwin32.exe, acvcwin32.exe, and avcwin32.exe) were downloaded from the C&C, after which the injected "msvcr.dll" initiated two binaries named with a*.exe and fvcwin32.exe. Furthermore, a few files (e.g., drive.cab, iestorge.dll, SAM.dll, system.dll, and 20110704145735.bmp) were created under the %SYSTEM%\Debug\Data folder. All files with the names *.dll and *.v2 were removed from the folder after uploading to the C&C through the encrypted HTTP traffic.

3.6. Static analysis of the downloaded binaries

3.6.1. avcwin32.exe. We used the binary "avcwin32.exe" to create two mutex with the names "my lovely wood" and "SPI64 RUNNING" before entering the main routine.

If the mutex creation was successful, information was extracted from the SAM file and a temporary file with prefix of "SAM" was generated at %AppData%. Furthermore, all passwords from "foxmail," "outlook," "outlook express," "IE Form Storage," "MSN," "Passport DotNet," and "protected storage," were collected from the infected machine.

All passwords were written in a temporary file with the prefix "自动表单.txt." (Figure 1.) These files were subsequently compressed in cab format, moved to c:\Windows\Debug\Data, and then renamed as "SAM.dll" and "iestorage.dll."

Figure 1. Contents of自动表单.txt

Finally, the binary "avcwin32.exe" renamed itself as "svcwin32.exe" and was then terminated.

3.6.2. fvcwin32.exe. Similar to "avcwin32.exe," the binary "fvcwin32.exe" tried to create two mutex called "my lovely wood" and "SPI64 RUNNING."

If the creation was successful, the hard disks, CDROM, and floppy disks were all scanned to collect all file names together with the MAC times.

The collected information was kept inside a file called drive (Figure 2.), which was compressed into cab format and placed under %Windows%\Debug\Data. The injected "msvcr.dll" removed this file after uploading to the C&C.

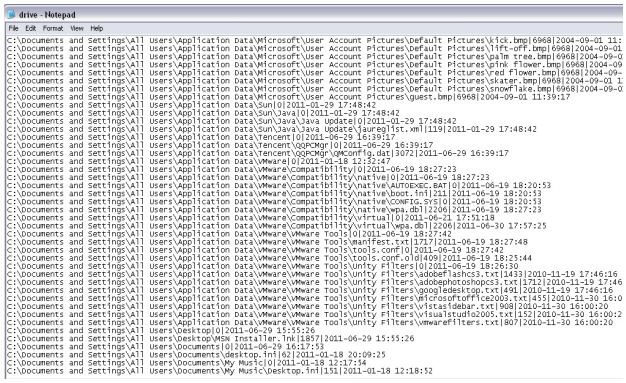

Figure 2. Extracted content of drive

3.6.3. acvcwin32.exe. The binary "acvcwin32.exe" created and compressed screen captures in bitmap format for every [value] milliseconds. All these screen captures were renamed with extension *.v2 and were removed after uploading to the C&C.

The binary looped into the function at 0x00401BEA to generate bitmap screen snapshots by calling APIs, such as "CreateCompatibleDC" and "CreateCompatibleBitmap." Then, looping continued to 0x004022A0 and 0x00401DB0 for every 1000 milliseconds (0x00401BF4).

4. Summary of the Findings

The spear-phishing emails were sent promptly, and directly corresponded to recent political incidents that easily attracted the attention of the specific targets.

The malicious function of capturing screenshots is normally employed if the victims use virtual keyboard software to enter their passwords, especially on financial websites. Keeping screen captures every 1000 millisecond is too much if only financial gain is the

main purpose. Moreover, collecting all file system information is an exceptional function for normal malware. These suggest the presence of ulterior motives behind such attacks.

From the technical perspective, the executable file bears similar characteristics and features to that described by the MANDIANT [4] and "Operation Shady RAT" [1] reports. These are described below.

- Nearly the same executable file has been sent repeatedly to targets, and these malwares are distributed in two different attack vectors (*.doc and *.chm).
- We used the TotalVirus online scanner and found that only 14/42 anti-virus engines can detect such malwares. Unlike traditional malwares, the online scanner cannot identify its type and family. In this work, the malware was recognized as Generic Downloader or Generic Backdoor because of the malicious behavior of DLL injection.
- The malware acts as a Dropper (the "Agenda.doc"), then generates the Droppee (the "msvcr.dll") and triggers the download of additional binaries ("fvcwin32.exe," "acvcwin32.exe," and "avcwin32.exe") that act as core modules performing the actual malicious functions.
- Filtered information is collected and compressed using an archival utility, and then uploaded through encrypted HTTP traffic; afterwards, the information is removed to hide its temporary presence.
- The malware uses large amounts of Windows API calls to keep its size compact.
- The dropper and malware do not use complicated encryption routines, but rather employ simple XOR encoding methods to avoid detection from intrusion detection or/and prevention systems.

5. Discussion, Limitations and Future Works

5.1. Discussion

Compared with other malware, the APT-type attacks follow a difference attack roadmap.

Take ZeuS bot as an example. It is a widely-spread Trojan for stealing passwords from the victim's online banking activities. The crimeware kit does not provide any propagation mechanism, but victims are frequently lured to open malicious attachments or click a malicious link to download the malware through spammed emails. From the attackers' perspective, victims' background and their affiliations are not a major concern and the pre-infection reconnaissance is not required.

The malware (T_0) generates a droppee (D_0), which in turn will be injected into a running process to perform pre-defined malicious functions, including collecting online banking password (I_0) or acting as a Trojan to control the machine for a long time or launching various attacks like DDoS (A_0). C&C is used to store collected information and control bots. Multiple C&Cs are usually deployed and the malware will connect to different C&Cs if the primary C&C is not working. The malware or its associated configuration file sometime may be updated (D_1) to generate a new version of droppee (P_1) to perform similar pre-defined malicious functions for collection of more information (I_0 or I_1) or performing more functions (A_0 or A_1). However, there is no clear relationship between the information collected and the added functions.

Our study illustrates a different roadmap for APT-type malware. It is a narrow-spread espionage malware for exfiltration of email/massaging passwords and documentation related information from the victim's hard disk. The spear-phishing emails are well crafted and consistently distributed to the victim in respond to any update incidents closely happening with the victim or the victim's organization. High pre-infection reconnaissance is required. The attackers require understanding the identity and background of the victims or the victims' affiliations. Also, the attackers are actively monitoring all of the happening events around the victims or victims' organization.

Similar to ZeuS bot, the APT-type malware (T_0) generates a droppee (D_0), which will be injected into a running process to collect preliminary information (I_0), including email or messaging passwords and all file names on the whole hard disk. In order to hide from the detection mechanism, the APT-type malware (T_0) usually does not carry obvious malicious functions, for example, it seldom change the infected system as a zombie machine for launching DDoS attack or sending spam emails. C&C is designed to allow the attackers to monitor the status of the infected systems, issue commands to drop additional payloads and retrieve more information whenever a particular event happens. Single C&C may be implemented and it is usually located in a country whose cybercrime laws are not effectively enforced. After the preliminary intelligent (I_0) is collected, we found that the APT-type malware

will be equipped with functions that can react to the collected information (I_0) or receive instructions from the attackers to download different payloads (P_1, P_2, P_3 … P_{t-1}) to collect different intelligent (I_1, I_2, I_3 …I_t). (Figure 3.)

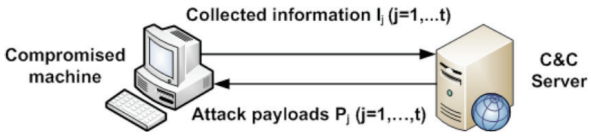

Figure 3. A Feedback Control Loop

5.2. Limitations

Unlike McAfee, we cannot access the C&C server nor allow too much contact with the C&C. Thus, we cannot identify some other malware functions. For example, in the code of the Malicious DLL, we found strings of *.v2 and *.dll inside the code segment while uploading the files from %Windows%\Debug\Data folder. However, strings of *.k2, *.dk, and PIfos.dat have also been found from the same code segment. We believe files with such extensions may be uploaded to the C&C when further instructions are received. Further investigation should explore whether or not the controller of the C&C sends back any instruction to the malware, such as to trigger it to copy a specific file, to rename it as *.k2 extension, or compress it in cabinet format. These files can be made available to the controller of the C&C. The intelligence monitoring party is possibly a non-technical user or group whose administrator can simply login on the C&C using a specific client plus the proxy browser and review all collected information safely from a remote location.

Nevertheless, with only select samples for analysis, we cannot generate reliable statistics to describe the correlation between malicious attachments and the associated political messages or the timeliness of suspicious emails with political incidents. We believe more accurate attack patterns can be identified if we can gather more samples from all political parties.

We also reviewed other suspicious emails and samples, and found that all of them were well crafted, bearing political title or contents, and then sent promptly to recipients. These emails can be traced back to the Hong Kong pre-election periods in 2007. Time and restricted resources prevent us from providing a detailed analysis of all emails sent during this period.

5.3. Future Works

Based on the attack roadmap discussed in the above, we believe the human factor is a critical aspect for detecting the APT-attack. At present, there is no security product on the market that can effectively mitigate spear-phishing attacks. We are going to develop a framework to identify, detect or prevent the APT-type attacks.

However, before a proper mitigation solution can be implemented, APT-attack has to be identified, measured and studied either on the spot or at postmortem. If sufficient statistics are obtained, the organization's threat level has to be raised and the analyzed results must be bought to the decision makers' attention.

6. Conclusion

No evidence was found to prove claims that the malware is prepared and sent from any state actor or any related political group. However, the findings show that the malware bears the same *modus operandi* mentioned in the GhostNet report [9], the MANDIANT report [4], and "Operation of Shady RAT" [1]. Based on high numbers of promptly sent spear-phishing emails and the functions of the malware sample, we believe the attacks were launched by a group of sophisticated, determined, and coordinated attackers actively targeting political parties in Hong Kong. This paper only demonstrates that APT is real and an actual threat in Hong Kong. Political espionage is only an illustrative example, and we believe similar threats exist, such as commercial or industrial espionage. Spying or even unethical detective investigations may be found if these cases are further studied.

7. References

[1] Alperovitch D., (2011). Revealed: Operation Shady RAT.

[2] Andress J., (2011). Advanced Persistent Threat, Attacker Sophistication Continues to Grow? ISSA Journal

[3] Cloppert M., (2009, July 22). Security Intelligence: Introduction (pt 1) (2009), Computer Forensics Blog, SANS.org. Retrieved on Aug 4 from http://computer-forensics.sans.org/blog/2009/07/22/security-intelligence-introduction-pt-1

[4] Mandiant. (2010). M Trends, the Advanced Persistent Threat.

[5] Operation Aurora (n.d.). Wikipedia. Retrieved on Aug 12 from http://en.wikipedia.org/wiki/Operation_Aurora

[6] Zeltser, L. (2007). Reverse Engineering Malware: Tools and Techniques Hanks –On. Bethesda: SANS Institute.

[7] Brand M., Valli C. & Woodward A., (2010). Malware Forensics: Discovery of the Intend of Deception. Edith Cowan University, Australian Digital Forensics Conference

[8] Valli C., Brand M., (2008). The Malware Analysis Body of Knowledge (MABOK), Edith Cowan University, School of Computer and Information Science

[9] The SecDev Group. (2009). Tracking GhostNet: nvestigating a Cyber Espionage Nework.

[10] Bindiff version 2.3 software from Zynamics (www.zynamics.com)

[11] Combs G., Wireshark version 1.2.2 software from Wireshark (www.wireshark.org)