A Host-Based Approach to BotNet Investigation?

Frank Y.W. Law, K.P. Chow, Pierre K.Y. Lai, and Hayson K.S. Tse

The University of Hong Kong,
Pokfulam Road, Hong Kong
{ywlaw,chow,kylai,hkstse}@cs.hku.hk

Abstract. Robot Networks (BotNets) are one of the most serious threats faced by the online community today. Since their appearance in the late 1990's, much effort has been expended in trying to thwart their unprecedented growth. However, with robust and advanced capabilities, it is very difficult for average users to avoid or prevent infection by BotNet malware. Moreover, whilst BotNets have increased in scale, scope and sophistication, the dearth of standardized and effective investigative procedures poses huge challenges to digital investigators in trying to probe such cases. In this paper we present a practical (and repeatable) host-based investigative methodology to the collection of evidentiary information from a Bot-infected machine. Our approach collects digital traces from both the network and physical memory of the infected local host, and correlates this information to identify the resident BotNet malware involved.

Keywords: BotNet, memory forensics, network investigation, malware.

1 Introduction

A BotNet normally refers to a cluster of machines that have been infected by a particular type of malware. These machines, or Bots, are controlled remotely by a Bot-herder and have the capability to perform a number of malicious activities autonomously and automatically. Such activities can include hacking, email spamming or even a Distributed Denial of Service (DDoS) attack against a specific target or targets [2, 3, 11]. The Bots are controlled and managed by the Bot-herder via one or more Command and Control (C&C) servers using specific network protocols.

A BotNet investigation may be initiated at the network level to search for possible C&C servers and other networked infected computers. Alternately, it may be initiated at the local level by investigating the infected host and tracing back upstream to other connected machines. The borderless nature of the Internet means that the BotNet infrastructure almost inevitably spans multiple countries and jurisdictions. A successful BotNet investigation and prosecution therefore requires synergy and cooperation between various jurisdictions and parties.

Knowledge and expertise of investigators in dealing with BotNets is known to vary from jurisdiction to jurisdiction. Some agencies utilise advanced techniques, whereas others lack the fundamental knowledge and tools by which to pursue a BotNet case. Additionally, whilst some agencies and organizations may have developed their own standard operating procedures for BotNet investigations, such methodologies have not

S. Goel (Ed.): ICDF2C 2009, LNICST 31, pp. 161–170, 2010.

been subject to scientific or peer review and validation. Such factors reinforce the need for harmonization of competencies between digital investigators and jurisdictions.

The prevalence of BotNets, and the desire to improve both the speed and quality of BotNet investigations, has prompted calls to develop a systematic and common investigative approach. Thus far most researchers have proposed investigating BotNets at the network level, but this neglects the importance and potential advantages of examining an infected host at the local level. It should also be noted that BotNets are constantly evolving and changing, e.g. from a centralized to distributed C&C structure, thereby increasing the complexity of network level only investigations.

It is observed that the BotNet infection model and the control mechanism at the infected host are quite similar, straightforward and stable in nature. We therefore propose a host-based approach to BotNet forensic investigations, wherein relevant digital traces from a local machine are collected to supplement any subsequent network level investigation.

Unlike those methodologies which are confined to internal use only by individual organizations, our approach is suited and intended use in the field by all practitioners. We believe that our proposed approach, through peer review and validation, can point towards a standardized method for all organizations involved in tackling BotNets, thereby enhancing the global fight against such malware.

2 Literature Review

Much BotNet research has focused on the analysis of BotNet behaviours, propagation methods, and ways by which to detect and stop their proliferation. There have been suggestions to identify IRC-based BotNets by passively monitoring network traffic to look for suspicious IRC-related traffic in a specific network [3, 4]. Honeypot approaches are also widely used to collect Bot samples and to study the behaviour of BotNets [25].

Dagon et al [5] analyzed the BotNet traffic of various regions and time zones and successfully created a diurnal propagation model to predict BotNet population growth. They also proposed a BotNet detection methodology by analyzing “rally” DNS traffic [6]. Other researchers have posited similar network traffic analyses [7, 8, 9, 10]. We call these approaches network-based BotNet investigation techniques.

Schiller et al [25] suggested investigating the infected host by scrutinizing event and firewall logs to determine the payload and functions of the Bot. They also suggested looking for suspicious start-up processes so as to identify the location of the malware. Barford [2] studied the overall architecture and implementation of BotNets via an in-depth analysis of the source code of Bot malware programs collected from an infected host.

We have already seen how network-based BotNet investigation techniques focus on the detection of BotNets and identification of Bot-infected machines at the network level. As the name suggests, these techniques use data collected from the network to steer the investigation, but seldom discuss ways to examine the Bot-infected host also to collect additional pertinent data.

Obviously network-based investigation is based on communication protocol information obtained from Bot-infected machines. This highlights the significance of host-based investigation and the fact that these two approaches are closely related. In comparison with network-based techniques however, host-based investigation is considered simpler and easier for the average digital investigator to apply and use.

When we started looking at BotNet investigations in depth we could find little previous academic study on host-based investigative approaches. This sparked our interest and subsequent research into this particular area. Our proposed approach differs from existing work by emphasizing the importance of digital traces that might be recovered from an infected host at the local level. Our approach also provides a reliable and repeatable method to trace C&C servers, and to recover Bot malware in the overall pursuit of a BotNet investigation.

3 The Methodology

In this section we discuss the digital traces that may be obtained from a Bot-infected machine and derive a methodology for investigation.

Under the simple hierarchy of a typical BotNet, the Bot-herder usually utilizes a layering approach to command his Bots and prevent possible detection of his location. Investigators often come across Bots at the bottom layer and these infected machines will normally contain information that might assist a BotNet investigation in identifying the next highest layer.

In brief, the goals of investigating a Bot-infected host are four-fold:

1. To reveal the numbers and locations of the C&C servers, and thereby estimate the size of the BotNet and derive a strategy for its disinfection;
2. To obtain the BotNet's command and control data to assist in the analysis of its hierarchy and functionality;
3. To recover the Bot malware to help understand its potential threat and methods of propagation, and to create a malware signature to enable its detection and disinfection; and
4. To derive an appropriate investigation strategy to trace the Bot-herder, based on the information collected from multiple machines.

In comparison to collecting BotNet information directly from the higher network level layer through the use of data flow monitoring, the host-based approach is more focused and direct, and the amount of data to be collated is far less. For example, network-based investigation often requires the capture of huge volumes of data traffic [4] to detect the location of the C&C server. Host-based investigation, meanwhile, requires less than several hundreds of megabytes of data traffic to achieve the same result.

According to the known and accepted order of volatility [12], network information will diminish much faster than other information existing on the machine. Memory data will dissipate next, whilst the BotNet malware program data will be last. To conduct a proper live investigation of a Bot-infected machine therefore, it becomes necessary to follow this same order of volatility.

In order to perform a live investigation of a BotNet infected machine, we propose to carry out the investigation in two phases. In phase one we collect digital evidence from the infected machine according to the aforementioned order of volatility:

1. Collection of network traces;
2. Collection of memory traces; and
3. Collection of malware traces.

The above digital traces help to understand the behaviour of the Bot and where it resides on the infected machine. However, since most Bots will automatically connect to the C&C server and their neighbouring Bots when the computer is started, certain pertinent information cannot be captured during phase one.

We therefore propose to reboot the machine at phase two, so that the Bot will initiate another round of network traffic in order to communicate with the C&C server. The information collected during phase two will assist the investigator to understand the behaviour of the Bot during initialization.

3.1 Phase One – Collection of Digital Traces

This section will discuss the practicalities of collecting relevant digital traces from a Bot-infected machine, whilst also attempting to minimize potential changes to system data.

In order to capture the required data traffic we propose setting up a simple network environment comprising the Bot-infected target machine and an investigator's machine with external hard disk. The two machines are connected to each other via LAN cable through a network hub. The network hub also serves as the gateway to the Internet (and therefore the BotNet). The proposed physical network topology is shown in the diagram at Figure 1.

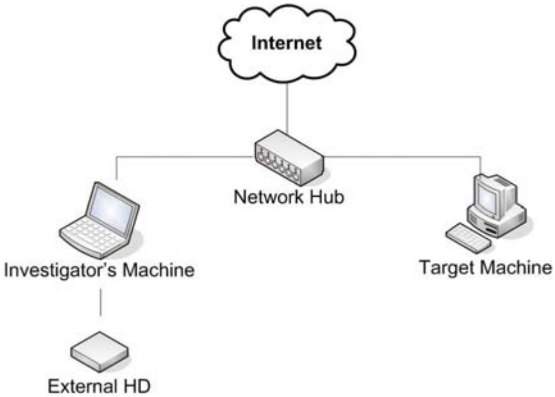
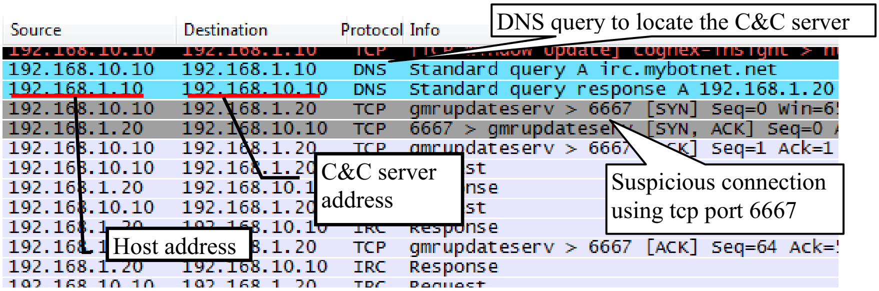


Fig. 1. Illustration of set up for network traffic collection

A network hub is used here because it will automatically broadcast all network traffic generated at the Bot-infected machine to the investigator's machine. The investigator's machine, meanwhile, is installed with packet sniffer software such as Wireshark [13] in order to capture the content of the network data traffic generated by the target machine.

Apart from the purposes of collecting network traffic data, the above topology also establishes a network bridge between the two computers to make possible the transport of data from the Bot-infected machine to the investigator's machine in the latter stages of an investigation. We further suggest attaching an external hard drive to the investigator's machine so as to facilitate the collection of all evidentiary data thereby generated.

Figure 2 gives a sample output of a network capture from a machine with IP address 192.168.10.10, and which is known to be infected with an IRC Bot. With early IRC type C&C, we could identify the IP address of the C&C server by simply looking for suspicious communications through TCP ports 135, 139, 445 or 6667 (ports commonly used by IRC Bots). In Figure 2 we find a suspicious connection using TCP port 6667 connected from IP address 192.168.1.20, which is the suspected C&C server. However, for BotNets using HTTP and P2P protocols, the C&C structures are different and we need therefore to identify the location of C&C server by observing suspicious connections through other ports [26, 27].



```
Source          Destination     Protocol Info
192.168.10.10   192.168.1.10    TCP      [TCP Window Update] cognex-insight > n
192.168.10.10   192.168.1.10    DNS      Standard query A irc.mybotnet.net
192.168.1.10    192.168.10.10   DNS      Standard query response A 192.168.1.20
192.168.10.10   192.168.1.20    TCP      gmrupdateserv > 6667 [SYN] Seq=0 Win=6
192.168.1.20    192.168.10.10   TCP      6667 > gmrupdateserv [SYN, ACK] Seq=0 A
192.168.10.10   192.168.1.20    TCP      gmrupdateserv > 6667 [ACK] Seq=1 Ack=1
192.168.10.10   192.168.1.20             st
192.168.1.20    192.168.10.1             nse
192.168.10.10   192.168.1.20             st
192.168.1.20    192.168.10.10   IRC      Response
192.168.1       8.1.20          TCP      gmrupdateserv > 6667 [ACK] Seq=64 Ack=
192.168.1.20    192.168.10.10   IRC      Response
192.168.10.10   192.168.1.20    IRC      Request
```

DNS query to locate the C&C server

C&C server address

Suspicious connection using tcp port 6667

Host address

Fig. 2. Sample output of a network capture from a machine infected with an IRC-Bot

By examining the TCP stream of the identified suspicious connection, we may be able to locate the IP address of the C&C server and even the content of the communication itself, if not encrypted. However, should encryption be employed to protect the communication, then the encryption algorithm needs to be known before analysis can take place.

One method is to exploit "Server Addressing" and "Server Authentication" tradeoffs to assist in the decryption of encrypted Bot communication content [1]. Apart from any C&C related information, the communication also hints as to the memory process that launched it in the first place.

After collecting network traces, the next step is to acquire memory information. In doing so we aim to identify the suspicious memory process that establishes the C&C connection, and to find the location of the Bot malware on the host machine.

Tools such as "fport" [14] and "openports" [15] may be able to reveal the suspicious memory process and its related network connection. However, these tools need to be run on the Bot-infected machine itself and can be intrusive in nature. Furthermore, if the Bot is actively hiding its process and network connection from memory, then these tools (and others like them) may fail in capturing the desired information.

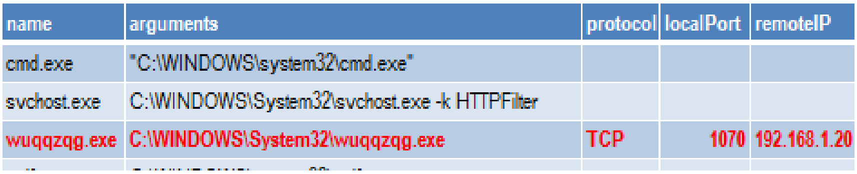
One way to overcome this problem is to acquire a memory snapshot from the target machine and analyze its content offline. This approach is better suited to handling hiding techniques of advanced BotNets, plus it avoids making potential changes to the system.

Any memory acquisition tool which does not cause adverse impact to the infected system can be used to obtain a memory snapshot for investigation. This can be done by putting the tool on a CD and executing it on the infected machine. The acquired memory snapshot can then be transported to the investigator's machine as shown in Figure 1.

By enumerating the memory process information [24] from the snapshot, it is possible to obtain network connection information that is related with the Bot process. Tools like "Memoryze" [16] or "Volatility framework" [17] are capable of performing offline analysis on the memory snapshot and obtaining necessary information for further investigation.

Figure 3 shows the result of analyzing a sample memory snapshot of a Bot-infected machine in which a hidden Bot process “wuqqzqg.exe” was found to be connected to a C&C server, (i.e. 192.168.1.20 - previously identified from the network traces in Figure 2). Apart from the memory process, the output also reveals that the malware was hiding inside the directory “c:\windows\system32”.

Armed with the above information, we should be able to discover the location of the malware. We can then preserve the malware file using common file extraction tools [19, 20, 21, 22] for later analysis. Similarly, these tools could be run from a CD and the malware copied to the investigator’s machine as in Figure 1 for further analysis.



name	arguments	protocol	localPort	remoteIP
cmd.exe	"C:\WINDOWS\system32\cmd.exe"			
svchost.exe	C:\WINDOWS\System32\svchost.exe -k HTTPFilter			
wuqqzqg.exe	C:\WINDOWS\System32\wuqqzqg.exe	TCP	1070	192.168.1.20

Fig. 3. Using the Memoryze tool to extract the memory process "wuqqzqg.exe" information from the memory snapshot

3.2 Phase 2 – Reboot and Recapture

Phase one is complete at this point. As we have already discussed however, most Bots are programmed to automatically connect to the C&C server and their neighboring Bots upon boot-up. This information is considered crucial to the investigation but may not be captured during phase one, since the machine is already booted up when phase one is initialised. We therefore recommend rebooting the machine so that the Bot will initiate another round of network traffic to communicate with the C&C server.

As the investigator's machine has already been set up to collect the Bot-infected machine's network traffic, we can continue using the network packet sniffer at the investigator’s machine to collect this data upon reboot.

Once the above actions have been completed, an initial assessment can then be made to decide if the evidentiary data collected from the Bot-infected machine is sufficient for prosecution purposes. If possible, consideration should be given to seizing or cloning the Bot-infected machine using established forensic procedures for later court proceedings.

Tables 1 and 2 summarize the steps for this host-based investigation.

Table 1. Summary of phase 1 investigation

Phase I		Steps
Network traces	i)	Install a network hub to establish a network bridge between the Bot-infected machine and the investigator's machine;
	ii)	Use a network sniffer to collect suspicious network data traffic from the Bot-infected machine and store onto to the investigator's machine;
Memory traces	iii)	Use memory acquisition tools to obtain memory snapshot(s) from the Bot-infected machine and store onto the investigator's machine;
	iv)	Analyse the memory snapshot(s) on the investigator's machine to reveal suspicious memory processes, network connections and the location of the Bot program; and
Malware traces	v)	Use file extraction tools to extract the Bot program to the investigator's machine.

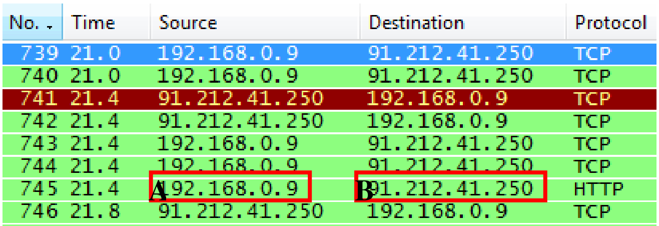
Table 2. Summary of Phase 2 investigation

Phase II		Steps
Network traces	i)	Reboot the Bot-infected machine to generate another round of network traffic;
	ii)	Again, use the network sniffer to collect suspicious network data traffic from the Bot-infected machine and store onto the investigator's machine; and
Malware traces	iii)	If necessary, seize or clone the Bot-infected machine for more in-depth computer forensic analysis.

4 Case Analysis

To test if our proposed approach was feasible and workable under real-life conditions, we invited law enforcements in Hong Kong to act as pilot tester and reviewer. Our approach was used on an actual BotNet investigation being carried out by the unit.

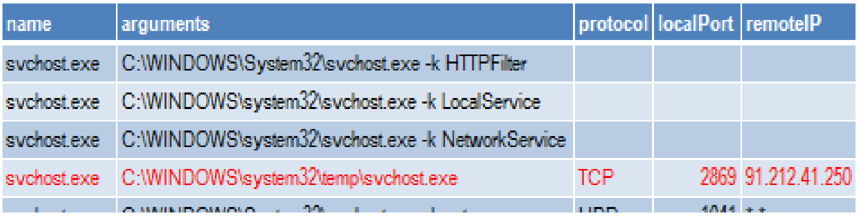
Acting on information, the unit identified a premises where a suspected Bot-infected machine was thought likely to be found. Upon entry, the digital investigator quickly set up the scene as stipulated in phase one and started to capture the traffic content of the Bot-infected machine using Wireshark [13]. Figure 4 is a Wireshark capture showing the Bot's connection to a C&C server via TCP port 80.



No.	Time	Source	Destination	Protocol
739	21.0	192.168.0.9	91.212.41.250	TCP
740	21.0	192.168.0.9	91.212.41.250	TCP
741	21.4	91.212.41.250	192.168.0.9	TCP
742	21.4	91.212.41.250	192.168.0.9	TCP
743	21.4	192.168.0.9	91.212.41.250	TCP
744	21.4	192.168.0.9	91.212.41.250	TCP
745	21.4	192.168.0.9	91.212.41.250	HTTP
746	21.8	91.212.41.250	192.168.0.9	TCP

Fig. 4. Captured communication between a Bot and C&C server showing: A) IP address of the host; and B) IP address of C&C server

The memory snapshot was collected by running the tool mdd.exe [18] from CD. The tool memoryze was then used to enumerate the network information and malware location. Figure 5 shows a screenshot of memoryze illustrating the suspicious malware process "svchost.exe" in the directory "c:\windows\system32\temp" utilizing TCP port 2869 to communicate with the C&C server at IP address 91.212.41.250.



name	arguments	protocol	localPort	remoteIP
svchost.exe	C:\WINDOWS\System32\svchost.exe -k HTTPFilter			
svchost.exe	C:\WINDOWS\system32\svchost.exe -k LocalService			
svchost.exe	C:\WINDOWS\system32\svchost.exe -k NetworkService			
svchost.exe	C:\WINDOWS\system32\temp\svchost.exe	TCP	2869	91.212.41.250

Fig. 5. The tool "memoryze" identified the suspicious process "svchost.exe"

By carefully examining the folder "c:\windows\system32\temp" on the infected machine the "svchost.exe" program was recovered. This malware program was then extracted by FTK imager running from CD and preserved onto the investigator's machine for further investigation.

Lastly, the infected machine was rebooted and another round of network traffic captured. The resultant information corroborated with the initial network capture and pointed to the same C&C server on the Internet.

It is worth noting that an anti-bot program had already been installed on the infected machine, but this had failed to detect the presence of the malware because it was a maiden or hitherto unknown version of the BotNet. The extracted program was subsequently sent to an anti-virus vendor to generate a suitable malware signature for disinfection purposes.

Because procedures had been pre-defined, the investigator in this case was fully conversant with the actions required at scene. In contrast to the traditional non-structured approach, this methodology was seen to be of practical benefit in enhancing the overall efficiency of the investigation and could successfully identify the BotNet C&C server being used to control the infected machines. Subject to further review, this methodology is considered competent in dealing with most other BotNet scenarios that an investigator is likely to come across in the real world.

5 Conclusions

Bot-herders are constantly evolving and adapting the structure of their BotNets at the network level to create ever more robust control mechanisms, and to avoid current detection techniques. Nonetheless, from our research we observe that the communication behaviours and characteristics of BotNets at the local machine level are stable, analogous and instructive.

Armed with this information, we propose a straightforward host-based investigation approach to collect relevant digital traces to detect and investigate BotNets. Our method is not novel, but rather a cocktail approach that leverages the observed similarities of Bot infected machines at the local level with established network based investigation techniques.

The evaluation of our approach by a law enforcement agency shows that it is capable of handling IRC and HTTP non-encrypted traffic from Bot-infected machines. Moreover our approach can swiftly retrieve digital traces to assist in the tracing of C&C servers and subsequent malware analysis. It is opined that such host-based investigation can supplement other network level investigative methods and provide a more thorough picture on how Bot-herders design and manage their BotNets

References

1. Ramsbrock, D.: Mitigating the Botnet Problem: From Victim to Botmaster, Master Thesis, George Mason University (2008), `http://mars.gmu.edu:8080/dspace/bitstream/1920/3136/1/Ramsbrock_Daniel.pdf`
2. Barford, P., Yegneswaran, V.: An inside look at BotNets. In: Proceedings of Special Workshop on Malware Detection. Advances in Information Security. Springer, Heidelberg (2006)
3. Cooke, E., Jahanian, F., McPherson, D.: The Zombie roundup: understanding, detecting, and disrupting botnets. In: Proceedings of the Steps to Reducing Unwanted Traffic on the Internet on Steps to Reducing Unwanted Traffic on the Internet Workshop, Cambridge, MA, July 07, p. 6 (2005)
4. Goebel, J., Holz, T.: Rishi: Identify bot contaminated hosts by irc nickname evaluation. In: Proceedings of USENIX Workshop on Hot Topics in Understanding Botnets, HotBots (2007)
5. Dagon, D., Zou, C., Lee, W.: Modeling botnet propagation using time zones. In: Proceedings of 13th Annual Network and Distributed System Security Symposium (NDSS), February 2006, pp. 235–249 (2006)
6. Dagon, D.: Botnet detection and response: The network is the infection. In: Proceedings of the Operations, Analysis, and Research Center Workshop, OARC (2005)
7. Choi, H., Lee, H., Lee, H., Kim, H.: Botnet Detection by Monitoring Group Activities in DNS Traffic. In: Proceedings of the 7th IEEE International Conference on Computer and Information Technology, Fukushima, Japan, October 16-19, pp. 715–720 (2007)
8. Romaña, D.A.L., Musashi, Y.: Entropy Based Analysis of DNS Query Traffic in the Campus Network. In: Proceedings for The 4th International Conference on Cybernetics and Information Technologies, System and Applications (CITSA 2007), Orlando, FL, USA, pp. 162–164 (2007)
9. Schonewille, A., Helmond, D.: The Domain Name Service as an IDS: How DNS can be used for detecting and monitoring badware in a network. University of Amsterdam (2006)
10. Karasaridis, A., Rexroad, B., Hoeflin, D.: Wide-scale botnet detection and characterization. In: USENIX Workshop on Hot Topics in Understanding Botnets, HotBots (2007)
11. Kristoff, J.: Botnets, detection and mitigation: DNS-based techniques. Information Security Day, Northwestern University (July 2005), `http://www.it.northwestern.edu/bin/docs/botskristoff_jul05.ppt`

12. Farmer, D., Venema, W.: Data Gathering and the Order of Volatility, Appendix B, Forensic Discovery. Addison-Wesley, Reading (2005), http://www.porcupine.org/forensics/forensic-discovery/appendixB.html
13. Wireshark, http://www.wireshark.org
14. Fport,
http://www.foundstone.com/us/resources/proddesc/fport.htm
15. Pslist,
http://technet.microsoft.com/en-us/sysinternals/bb896682.asp
16. Mandiant Memoryze v.1.2.18.0,
http://www.mandiant.com/software/memoryze.htm
17. Volatility Framework,
https://www.volatilesystems.com/default/volatility
18. Memory DD v1.3, http://www.mantech.com/msma/MDD.asp
19. X-Ways Capture v1.18, http://www.x-ways.net/capture/index-m.html
20. F-Response Field Kit Edition v1.18, http://www.f-response.com/
21. Encase Forensic Tool, http://www.guidancesoftware.com/
22. FTK imager, http://www.accessdata.com/
23. Helix Live CD, http://www.e-fense.com/products.php
24. Lee, R.: Memory Forensic Acquisition and Analysis 101, 2008-11-19,
http://sansforensics.wordpress.com/2008/11/19/memory-forensic-analysis-finding-hidden-processes/
25. Schiller, C., Binkley, J., Evron, G., Willems, C.: Botnets – The killer web app. Syngress, 179–208 (February 2007)
26. Grizzard, J., Sharma, V., Nunnery, C.: Peer-to-Peer Botnets: Overview and Case Study. In: Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets, Cambridge, MA, p. 1, April 10 (2007)
27. Taxonomy of Botnet Threats, A Trend Micro White Paper (November 2006),
http://us.trendmicro.com/imperia/md/content/us/pdf/threats/securitylibrary/botnettaxonomywhitepapernovember2006.pdf