The Classic Cyber Defense Methods Have Failed – What Comes Next?

Amir Averbuch and Gabi Siboni

Introduction

The classic defense methods employed throughout the world in recent decades are proving unsuccessful in halting modern malware attacks that exploit unknown (and therefore still unsolved) security breaches called "zero-day vulnerabilities." Viruses, worms, backdoor, and Trojan horses (remote management/access tools – RATs) are some examples of these attacks on the computers and communications networks of large enterprises and providers of essential and critical infrastructure and services.

The classic defense methods, which include firewall-based software and hardware tools, signatures and rules, antivirus software, content filters, intruder detection systems (IDS), and the like, have completely failed to defend against unknown threats such as those based on zero-day vulnerabilities or new threats. These sophisticated and stealth threats impersonate reliable and legal information and data in the system, and as a result, the classic defense methods do not provide the necessary defense solution. The current defensive systems usually protect against known attacks, creating heuristic solutions based on known signatures and analysis that are already known attacks,[1] but they are useless against the increasing number of unfamiliar attacks that lack any signature. Solving this problem requires different thinking and solutions. This article proposes an up-to-date approach, based on an analysis of sensitive information that

Prof. Amir Averbuch is a faculty member at the Blavatnik School of Computer Science at Tel Aviv University and a researcher in the INSS Cyber Warfare program sponsored by the Neubauer Foundation.
Dr. Gabi Siboni is the head of the INSS Military and Strategic Affairs Program and head of the INSS Cyber Warfare Program.

must be protected, for the purpose of identifying anomalous behavior.[2] The analyzed information includes an organization's data silos as a means of understanding unusual (anomalous) activity that in most cases indicates the presence of malware in the system. The article further proposes relying on the data to be protected as a source of knowledge for developing the defense system. An analytical analysis of massive data (big data analytics) will make it possible to identify such malware, while constructing a model that will provide a high degree of reliability in identifying and minimizing false positives, which pose a challenge to every defense system.

Development of Threats and the Limitations of the Traditional Defense Systems

The first cyber attacks on computer systems were based on viruses or worms that reproduced themselves and spread rapidly. Antivirus technology, however, completely failed to detect Trojan horses, whose behavior was entirely different than that of viruses. Traditionally, defense systems were developed to protect against known viruses, because it is quite difficult to identify such viruses by their behavior rather than their signatures. In this way, it became possible to create a database of virus signatures, and to compare files and communications reaching computers with these signatures. This approach required manufacturers of defensive software to continually monitor the development of viruses in order to create their signatures and distribute updates to their customers for the purpose of enabling them to update as quickly as possible the systems on which the protective software based on these signatures was installed. The burgeoning development of various forms of viruses and malware and the enormous growth in their number rendered this process virtually impossible, because major investments of resources in the continual updating of signature data for antivirus software were required.

The cyber attack hazards can be roughly divided into the following families: malware, spyware, worms, and Trojan horses (which open "backdoors"[3]). A classification that relates more to the object of an attack includes advanced persistent threats (APTs), which began with countries launching cyber attacks against other countries' military networks and the networks of government agencies, and in recent years developed into an attack by one country directed at another's organizational network of critical civilian infrastructure, and attacks against computer-operated industrial

supervisory control and data acquisition (SCADA) systems – such as the Stuxnet attack. Essential infrastructure systems controlled by industrial control systems in which control is exercised by the SCADA protocol are therefore exposed to attacks that are liable to paralyze the essential services, and could even suffer physical damage. Other possibilities include attacks against wireless systems and mobile broadcasting stations, the use of social networks for the purpose of spreading spyware and malware, and an attack against storage and cloud computing services.

The realm of attack in cyberspace can be divided into two types of attacks that exploit numerous weaknesses, including zero-day vulnerabilities:

a. *Broadcast attacks* are attacks that try to damage computers indiscriminately. They also feature extensive infection of software agents in order to create an entire network of computers (Botnet), with the aim of making these computers execute independent commands at a later stage or retrieve commands from a control server. As noted above, when information about new threats reaches the antivirus companies, they identify the signature or investigate them heuristically. By means of regular updates, the computers can be protected against these attacks. Given the extensive target community, the information about such threats will undoubtedly reach the relevant companies rapidly and be inserted into future versions of their products. In some cases, the goal of an attack of this kind is to reach a large number of computers – for example, employees (in the case of an attack against an organizational network) or customers (in the case of an attack against a financial institution, an attempt to steal credit cards via the internet, and so on). After the computer is infected, a Trojan horse is installed on it, making it possible to steal information or access the computer from a remote location. These attacks include various types of malicious code, even codes that vary from one infection to another in order to render identification through a signature more difficult (polymorphic viruses). There is still no complete defense since Trojan horse developers regularly check whether the antivirus software programs have already identified the hostile code and created the signature or group of heuristic rules to intercept it. In most cases, if the detection systems manage to identify the hostile code, the developers change the way it spreads or the way it operates in order to prevent

its detection. In this way, many Trojan horses consistently succeed in evading detection by the leading defensive software.

b. *Targeted attacks* are planned especially for a specific need, and exploit unknown weaknesses in the operating systems or widely known software packages while independently spotting new weaknesses. The vast majority of antivirus software, which is by nature based on signature defense, is incapable of identifying and preventing this type of attack, and the limited target community enables such attacks to evade the "radar" of antivirus manufacturers. It should be noted that threats are rapidly developing in the direction of focused attacks on high caliber targets.

The volume of data transmitted on a modern communications network is very large, owing to the need to provide many services to various kinds of end stations, including PCs, work stations, servers, switches and communications equipment, and many other diverse units. Such networks have many users, most of whom have no security awareness at all. As a result, APT attacks focus on people as well as on machines − via social networks, for example. The attack on the RSA company, which targeted the people in the organization, succeeded in penetrating the most secure systems.[4]

In recent years, we have seen a dramatic rise in the volume of new, undocumented, sophisticated attacks of a stealth nature. This is reflected both in the group of general attacks and in focused attacks. These attacks are overcoming all the classic standard defenses of the companies currently leading the protection sector. Major investments by countries and organized crime are responsible for the development of these attack methods, and the resulting damage is extensive.[5] The quantity of malware successfully penetrating all the existing defense systems and overcoming all the signature and rule-based classic defenses is increasing by leaps and bounds. The rate of increase has been in the three-digit percentages from 2011 until the present time.[6]

The existing systems are based mainly on preventing and thwarting known threats through the use of signatures and rules that are known in advance. Having no known signature at any given moment, these systems cannot detect zero-day attacks. They also find it difficult to identify Trojan horses and backdoors, and many sophisticated stealth attacks have no known signatures. Because they appear to be legal data and code, and do

not look like malware, they can penetrate almost any computer system. The attacks succeed in penetrating organizational networks and end-user computers despite all the defense systems; this is attributable to the fact that the initial appearance and behavior of the malware appears to be legal and proper. Furthermore, most of today's operating systems are built to handle a certain kind of attack, and are unable to deal with a broad range of attacks with mutations and secondary attacks.

In conventional software, one way of detecting unfamiliar and unsigned attacks is by identifying abnormal behavior of codes residing in the organizational systems, which differs from the way most normal data behave. This different behavior is what betrays hostile codes. The notion of the irregular behavior of a software element attempting to conduct unauthorized activity could serve as a possible basis for identifying and preventing attacks. Software producers worldwide understand the challenge and are taking steps to furnish such identification capabilities. This, however, is where the most significant challenge lies, namely, the difficulty in providing a reliable tool that will not produce false alarms or affect the user experience in an extremely negative manner. False alarms, which constitute one of the most significant challenges in defense systems, are created when the system issues a warning for a legal code with normal behavior and defines it as a hostile or suspicious code. If the load of such false alarms is too heavy, it will significantly harm the working capability of the computer systems, and is liable to cause the user to lose confidence in the defense system.

The second challenge is finding a solution for malicious code that evades the defense system. This phenomenon is called a false negative – when a result is obtained that appears negative, but is actually positive (comparable to a bearer of a serious virus who receives a negative test result from a laboratory when the virus is actually present in his body). These two challenges lie at the heart of defense systems in general, particularly in the use of analysis of the anomalous behavior of hostile code in an information system.

Identifying Anomalies as an Approach to an Operative Solution

This article focuses on the protection-based detection of anomalies in communications networks at various levels. The problem is broader, however, and includes the need to identify anomalies of hostile codes that

have penetrated weak points in software programs and applications. This approach is not discussed in the present article, unless the hostile code is exposed in the organizational communications. Regardless of the above, one can assume that some of the ideas mentioned are also suitable for detecting anomalies in software and applications.

Anomalies first proposed in 1987[7] are deviations from the expected behavior, which is the normal behavior. The basic assumption for any system seeking anomalies posits that malicious data have characteristics that are not found in the normal behavior specified during the learning phase. Since 1987, additional theories and methodologies have been developed, based on machine learning approaches and on the theory of information,[8] such as nervous systems,[9] a support vector machine,[10] genetic algorithms,[11] and many others. There are also numerous approaches that utilize data mining in order to find hostile code.[12] A general review of finding anomalies appears in an article by Chandola and Banerjee,[13] and there is a study of methods for spotting hostile code.[14]

One approach to detecting attacks on data from communications networks entails monitoring anomalies in network activity by finding the deviation from a normal profile learned from benign (proper non-malware) data. This methodology is based on tools retrieved from studies in machine learning,[15] mathematical and stochastic analysis,[16] statistics, data mining, graph theory, information theory, geometry, probability theory and random processes, and so on. Machine learning and data mining tools, combined with the above methodologies, are used successfully in many other fields, such as systems for recommending Amazon products,[17] Netflix,[18] optical character recognition,[19] translation of a natural language,[20] and identifying junk e-mail (spam).[21] Machine learning deals with the development of algorithms that enable a computer to learn, based on examples. Supervised learning of data known in advance, in which the correct significance of the parameters is known ahead of time, namely, labeled data, already exists. In unsupervised learning, the goal of the algorithms is to find a simple representation of the data without labels. Supervised learning is more limited with respect to the data content being learned. On the other hand, the results are more reliable, and it is therefore preferable.

Learning first takes place with a "healthy" group of data, which presumably contains no malware at all. This is called the "training set." It is usually best for the learning method to be able to detect whether part of

the training set contains malware up to a given percentage of all the data. Obviously, if most of the training set contains malware, it will be identified as normal data. As part of the filtering process, a process called "outlier removal" is used, which removes data that appear to be noise or infected from the training set.

The training set is analyzed by a variety of existing mathematical methods combined with innovative methods. The normal characteristics of the examined data can be identified through this process. This type of learning is called "one class." Another method, in which the characteristics are learned through comparison with a training set containing both clean and unclean data (e-mail with and without spam, for example) is called "binary class." The training set is derived from a mass of data accumulated and protected in an organization, together with continually guarded new data. For this purpose, methods of learning the data characteristic of normal behavior have been developed. While understanding the geometry of the learned data is one of the analysis methods, other methods also exist. For example, the following process describes a possible general structure of algorithms used as well as the processors of the training set in order to find the characteristics of normal (proper) behavior:

a. Breaking down each basic unit of communications or event data into characteristics (features, parameters).
b. Quantifying the relationships among the characteristics. There are a number of methods of characterizing such relationships. The kernel method[22] is one of the most common methodologies for defining them. Mathematical distance functions are usually used to define these relationships, which are near/far relationships with a range of characteristics existing between them. After this stage, the relationships between the communications data or events are guarded.
c. Lowering the dimension of the data. The dimension of the data is usually high, and is determined according to the number of characteristics making up a basic communications unit or basic event unit. The dimension of the data[23] is therefore lowered (from ten dimensions to two, for example), while preserving the relationships and coherence among the characteristics that were identified at the preceding stage. This is similar to sampling, in which only a small, reliably representative part of the original data is logically selected. Mathematical, algorithmic, and conceptual innovation is required in order to process data from

a high dimension that will suit a computer and reliably represent the original data. The sampling, which is aimed at reducing the volume of data, can be random, and it can be proved that the coherence of the data is maintained. There are many mathematical methods for achieving this objective. One of the methods for streamlining the computations in order to construct a compact representative of multi-dimensional data is the construction of dictionaries in order to speed up calculations while maintaining the relationships and features identified before the dimension was lowered. Other methods for speeding up computations facilitate sparsification of the data. The goal of these approaches is to specify a normal profile for the data from the training set while overcoming heavy computational problems in processing the training set. The learning action is usually computationally heavy. This action is conducted offline, and need not take place in real time. Common methods include PCE,[24] LLE,[25] ISOMAP,[26] and so forth.

The methods described above make it possible to effectively process the training set, which is "heavy" and liable to make calculations impossible. The goal of processing the training set is to specify the training data's ordinary (normal) behavior, based on an examination of the training set and the relationships defined between the characteristics of the data and the events of the training set. This assumes that the learning and the conclusions derived from it will reflect the normal behavior of all the future new data that are not part of the training set. As the volume of data in the training set increases and its characteristics become more numerous and diverse, the normal behavioral characteristics derived from the training set become more reliable. The calculation is more complicated, however, and it is therefore necessary to invest a great deal of effort in producing algorithms that are computationally effective and can handle large volumes of data.

The process described above specifies a possible learning model that generates a specification of the normative behavior of future data with the help of the training set's normal profile. From there on, the characteristics of all new information arriving, or of a new event, are examined. These characteristics are processed in order to see whether they deviate from the normative profile learned and determined during the learning (an anomaly). Deviations from the normal profile make it necessary to identify the attacks characterized as zero-day attacks. The method described thus

far does not use signatures; it finds behavioral deviations from the normal profile generated by processing the training set.

Figure 1 is a procedural description of the learning process described above. The chart also presents the range of sources from which the information has been retrieved for the purposes of the initial learning.

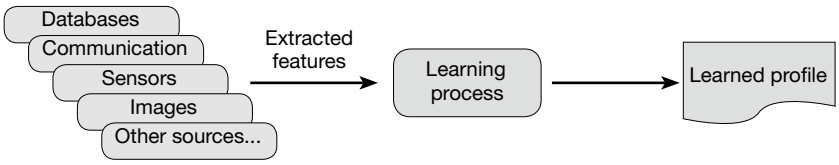

Figure 1. The Learning Process Chart

These methods and their derivatives for finding malware by monitoring the behavior of the data can be used in two different and complementary ways. The common denominator in these two ways consists of offline learning of the communications data from the protocol through which the data reach the organization (for example, port 443 [HTTPS], UDP port 53 [DNS], TCP, and TCP port 80 [HTTP], which are also web protocols) and constructing a profile that describes the normative behavior of the data of a given protocol that must be checked, according to the training set.[27]

a. *Operation in real time*. The algorithm for finding anomalies in communications data (accomplished in software or hardware) is located at the entrance to the organization. After data pass through the ordinary IPS Firewalls and IDS defense tools (signatures and rules allow them to enter), the algorithm checks each communications unit – whether its behavior matches the normal profile learned from the training set. If it proves to be an anomaly, its path into the organization is blocked. Since signatures are not used, the analysis of the substance of the anomaly can be performed either automatically or manually.
b. *Offline operation* – finding malware offline. Communications data that entered the organization through all the defense systems appear to be legal data, and subsequently begin to operate. An example of this is a spyware network absorbed into the environment with the aim of operating in the future. For this purpose, logs and events that occurred previously and are occurring now should be processed. In order to process information from both preserved and newly arrived

logs, security information and event management (SIEM) technology is used. SIEM, an information security monitoring system commonly used in organizational networks, serves as a central location for preserving and decoding logs and events of communications data. SIEM, an archive of all the communications data and events, helps conduct forensic analysis in order to find anomalies.

The above-mentioned methods of finding anomalies can be applied to the data collected by SIEM. Other data mining tools can also be applied to the SIEM data. SIEM contains two functions for security management: security information management (SIM) and security event management (SEM). The method that employs SIEM data should constantly apply the methodology for finding anomalies in order to identify the operation of malware when it is activated at some future date.

Figure 2 describes processes for checking information, given the results of the learning analysis:

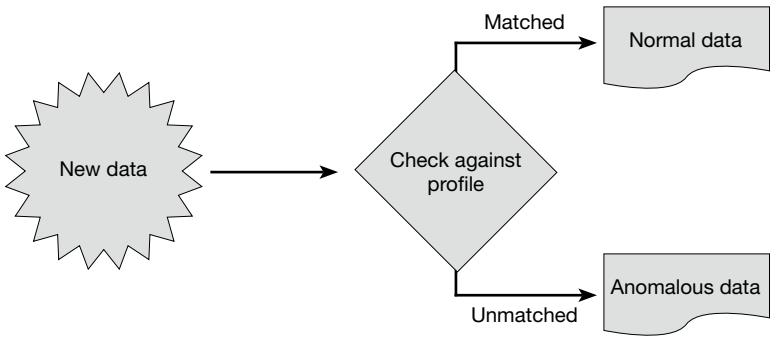

Figure 2. The Identification Process Chart

The Use of Big Data to Find Anomalies: The Data and Events Dictate the Identification Method

As described above, the main idea on which finding anomalies is based is specifying the behavior of the data in the training set and drawing conclusions from it with regard to the behavior of the data that did not participate in the training set, that is, characterizing the newly arrived data. In other words, the data dictate the processing, as reflected in the algorithms whose task was to learn the data as they are, and to adapt to them. This is in contrast to all the existing defenses against malware,

which seek patterns of already familiar malware and are unrelated to the behavior of the data. In the case of communications data, the data from each information unit of the protocol being monitored are analyzed. The relationships between the data are found by using the kernel method, and they are stationed in non-linear fashion in spaces with a lower dimension. The dimension of the data, which is usually high, is lowered in this way, thereby creating an effective way of finding anomalies.

Today, the data in which we look for anomalies are referred to as "big data," that is, a huge volume of data collected from all the information sources available on the organizational network. In many organizations, they are guarded by SIEM methodology. According to former Google CEO Eric Schmidt, the quantity of data created between the dawn of civilization and 2003 was five exabytes.[28] Schmidt asserts that this quantity is now created every two days. The following are a number of examples of the creation of big data every single day: the New York Stock Exchange (NYSE) creates one terabyte of data, Facebook creates 20 terabytes of compressed data, and the CERN particle accelerator in Switzerland creates 40 terabytes of data. According to a published report,[29] the volume of data doubles every year, and at least half of all businesses keep their data for at least three years for analytic purposes. Some of them are legally required to keep these data for a number of years. New sources of enormous quantities of data are constantly emerging in various businesses such as utilities. The bulk (80 percent) of these data is unstructured, which means that the organization is therefore unable to use them effectively. Big data have become a source of data mining that facilitates the identification of malware. Many well known companies such as Facebook, Google, Amazon, LiveJournal, and Wikipedia possess quotidian big data, and this list is far from complete. Today, big data are kept in the cloud. The quantity of data stored in each organization is huge, and is constantly growing. In order to handle large data silos, tools have been developed for processing big data that are unrelated to data mining or finding anomalies, such as Hadoop,[30] MapReduce,[31] and Memcached[32] – enormous parallel databases[33] that facilitate rapid data queries. In addition, many communications "pipelines" are being developed (by the Mellanox company for instance) for high speed transmission of these quantities of data. A great deal of effort is being expended on developing advanced tools for effective processing of big

data. Big data can therefore serve as a source for finding a broad range of sophisticated behavioral anomalies of different varieties of malware.

Conclusion

In order to process big data and effectively identify "high quality" malware, it is necessary to combine all the methods listed above. Tools – most of which are non-linear – were mentioned for reducing the volume of multi-dimensional big data without affecting the coherence of the data, at the same time maintaining the efficiency of the algorithms, for the purpose of handling huge volumes of data. The methods mentioned in this article that should be added are: learning from a small group of data; and using the kernel method on data, thereby determining the relationships (distances) between the sample points and reducing the dimension of the data by means of discrete or random sampling. This thins out the data, thereby obtaining an effective "housing project" of multidimensional big data in a significantly lower dimensional space in which anomalies are identified. Constructing dictionaries and using sophisticated and effective algorithms, together with big data processing tools, create many possibilities for finding malware in any organization by specifying the normative behavior and identifying deviations from it.

The proposed approach is a combination of computationally effective big data analysis and advanced tools for finding anomalies that are malware of zero-day attacks that do not yet have known signatures and behavior patterns. The methodology discussed here requires finding a needle in a haystack of data.[34] The point of departure states that the proposed algorithms adapt themselves and become accustomed to the data themselves. The data dictate how the algorithm operates. The methodology proposed in the article combines an understanding of the data structure by learning from a small group and drawing conclusions about the future behavior of the data that were not included in the learning set. This methodology is capable of detecting both malware whose activity is immediate, and malware, such as Trojan horses, that has entered the organization and will become operational at a later date.

Notes

1 "Heuristically" means through rules that help detect the harmful code.

2 Anomalous behavior of software code or information is unusual (uncharacteristic) behavior that arouses suspicion of malware in a system.

3 A security breach facilitates access to a computer without the need to verify an identity. This can result from a software error, a deliberate breach in the original code, or the installation of special software (such as a Trojan horse).
4 Gabi Siboni and Y. R., "What Lies Behind Chinese Cyber Warfare," *Military and Strategic Affairs* 4, no. 2 (2012): 43-56.
5 Symantec, "Internal Security Threat Report," *2011 Trends* 17, April 2012.
6 "FireEye Advanced Threat Report – 1H," *Source* 2012, http://www2.fireeye.com/advanced-threat-report-1h2012.html.
7 D. E. Denning, "An Intrusion-Detection Model, IEEE Trans," *Software Engl* SE-13, no. 2 (1987): 222-32.
8 W. Lee and D. Xiang, "Information-Theoretic Measures for Anomaly Detection," in *Proc. IEEE Symposium on Security and Privacy* (2001).
9 Z. Zhang, J. Li, C. Manikopoulos, J. Jorgenson, and J. Ucles, "HIDE: A Hierarchical Network Intrusion Detection System Using Statistical Preprocessing and Neural Network Classification," in *Proc. IEEE Workshop on Information Assurance and Security* (2001).
10 W. Hu, Y. Liao, and V. R. Vemuri, "Robust Anomaly Detection Using Support Vector Machines," in *Proc. International Conference on Machine Learning* (2003).
11 C. Sinclair, L. Pierce, and S. Matzner, "An Application of Machine Learning to Network Intrusion Detection," in *Proc. Computer Security Applications Conference* (1999).
12 M. A. Siddiqui, *Data Mining Methods for Malware Detection*, PhD dissertation, University of Central Florida (2008).
13 V. Chandola, A. Banerjee, and V. Kumar, "Anomaly Detection: A Survey," *ACM Computing Surveys (CSUR)* 41 no. 3, Article 15 (2009).
14 N. Idika and A. P. Mathur, "A Survey of Malware Detection Techniques," Department of Computer Science, Purdue University (2009).
15 R. Sommer and V. Paxon, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," in *Proc. IEEE Symposium on Security and Privacy* (May 2010).
16 Stochastic processes are processes whose development over time includes a certain element of randomness at any given moment.
17 G. Linden, B. Smith, and J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," *IEEE Internet Computing* 7, no. 1 (2003): 76-80.
18 J. Bennet, S. Lanning, and N. Netflix, "The Netflix Prize," in *Proc. KDD Cup and Workshop* (2007).
19 L. Vincent, "Google Book Search: Document Understanding on a Massive Scale," Proc. International Conference on Document Analysis and Recognition, 2007; R. Smith, "An Overview of the Tesseract OCR Engine," in *Proc. International Conference on Document Analysis and Recognition* (2007).
20 F. J. Och and H. Ney, "The Alignment Template Approach to Statistical Machine Translation," *Comput. Linguist* 30, no. 4 (2004): 417-49.

21 P. Graham, "A Plan for Spam," in *Hackers &* Painters: Big Ideas for the Computer Age (O'Reilly, 2004).

22 B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond* (Cambridge: MIT Press, 2002).

23 M. Elad, *Sparse Redundant Representations: From Theory to Applications in Signal and Image Processing* (New York: Springer, 2010).

24 I. T. Jolliffe, *Principal Component Analysis* (New York: Springer, 1986).

25 S. T. Rowels and L. K. Saul, "Nonlinear Dimensionality Reduction by Locally Linear Embedding," *Science* 290, no. 5500 (2000): 2323-26.

26 J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A Global Geometric Framework for Non-Linear Dimensionality Reduction," *Science* 290, no. 5500 (2000): 2319-23.

27 This approach also facilitates performance monitoring, an analysis of users' behavior, an analysis of man-machine relationships, and control of processes.

28 1 exabyte = 1 billion billion bytes.

29 M. G. Siegler, "Eric Schmidt: Every 2 Days We Create as Much Information as We Did up to 2003," *TechCrunch*, August 4, 2010, http://techcrunch.com/2010/08/04/schmidt-data/.

30 Web page: hadoop.apache.org.

31 J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," *OSDI* (2004).

32 L. Gavish, *New Caching Policies for MEMCACHED*, MSc Thesis, Tel Aviv University (2012); B. Fitzpatrick, "Distributed Caching with MEMCACHED," *Linux Journal*. 2004, no. 124 (2004): 5.

33 Hadapt, http://hadapt.com/.

34 M. Baker, D. Turnbull, and G. Kaszuba, "Finding Needles in Haystacks (the Size of Countries Blackhat)," Amsterdam, The Netherlands, March 14-16, 2012.