Game theory in cyber security defence

Conference Paper · June 2013

DOI: 10.1109/ECAI.2013.6636210

CITATIONS
2

READS
879

2 authors:

Alecsandru Patrascu
Technical Military Academy of Bucharest
30 PUBLICATIONS **138** CITATIONS

SEE PROFILE

Emil Simion
Polytechnic University of Bucharest
86 PUBLICATIONS **133** CITATIONS

SEE PROFILE

Game Theory in Cyber Security Defence

Alecsandru PĂTRAŞCU
Advanced Technologies Institute
Bucharest, Romania
ati@dcti.ro, alecsandru.patrascu@gmail.com

Emil SIMION
Advanced Technologies Institute
Bucharest, Romania
ati@dcti.ro, esimion@fmi.unibuc.ro

***Abstract* – Cyber threats and the field of computer cyber defense are gaining more and more an increased importance in our lives. Starting from our regular personal computers and ending with thin clients such as netbooks or smartphones we find ourselves bombarded with constant malware attacks. In this paper we will present a new and novel way in which we can detect these kind of attacks by using elements of modern game theory. We will present the effects and benefits of game theory and we will talk about a defense exercise model that can be used to train cyber response specialists.**

Keywords-cyber security, cryptography, steganography, critical infrastructure, industrial control systems, car viruses, information warfare, malware

I. INTRODUCTION

Internet today has faced an explosive increase in number of user. If we compare it with other technologies existing in our lives, we can see that it has known the fastest growth rate. An estimated 1.6 billion people regularly access the web, and while most people log in, log out and harm no one some of them do among them criminals, malicious hackers and terrorist. The threats they pose are huge and multiple. Today online is the new frontline. Some statistical records are showing that:

- the number of malware attacks for Android systems have increased in 2012 with 700%;
- the average time between infection and detection is 170 days;
- the average time to solve malicious insider attacks is 57 days;
- the average time for solving a cyber attack is 24 days, with a cost of 600.000 USD per day;

Thus, for **optimizing the cost** of detection, investigation, incident mitigation, containment, recovery, post incident activities we need to know and anticipate the moves of the cyber enemy.

We can simulate this situation by strategic games in which two players *A* and *B*, which can select individually their actions *a* respectively *b* and will get a *payoff L(a,b)*. The players *A* and B are supposed to be antagonistic: player *A* want to maximize the minimum income and player *B* want to minimize the maximum loss. In game theory the concept of "dominance and equilibrium" play a central role. A decision a_1 is dominated by a_2 if the payoff for a_2 is greater than the payoff for a_1 whatever decision *b* will select the adversary. Players *A* and *B* are in *Nash equilibrium* if *A* is making the best decision he can, taking into account *B* decision, and *B* is making the best decision he can, taking into account *A* decision.

Each game is characterized by the inferior game value $\alpha = \max \min L(a;b)$ and the superior game value $\beta = \min \max L(a;b)$, where min and max are performed by pure strategies of *A* and *B*. Generally, $\alpha \leq \beta$ and if have equality then the players *A* and *B* are in equilibrium. The game theory tells us that exist mixed strategies (set of pure strategies played with some probabilities) for which $\alpha = \beta$. Simulation of the situation can be made using several software applications such as Gambit: Software tools for game theory [1].

For cyber network defense we may to consider the following extended game model:

- **Players:** cyber attackers and network defense system are two players of (possible Markov game);
- **States:** all the possible states of involved network nodes consist of the state space;
- **Action Space:** at every time step, each team chooses targets with associated actions based on its observed network information;
- **Transition Rules:** updating the states of the game depending of input action;
- **Payoff Functions**: the gain/loss of each player;
- **Strategies and COA**: strategies are set of moves performed by each player and can be deterministic or mixed strategies (in probability). Course of actions are sequences moves that *A* or *B* may follow. As an e**xample of strategies of A** is SQL injection and **examples of strategies of B** is Patch applications/use the latest version of applications.

The extended game model allow us to learn from several situations and improve our defending strategies.

In this paper, along with other specific tools and techniques that are currently used in traditional computer networks we are going to present how can we use the same instruments for this specific field. In this case, the use of scenarios is the main way in which people involved in incident response can be trained. Often these scenarios are presented to the regular people in a masked form, and a good example in this field is represented by an international challenge called "Cyber MITRE", organized by the Federal Bureau of Investigation and the Fordham University. We will present in this case seven basic scenarios that

can be used in real life transportation network security breach: identification of encrypted data in a file, decrypting a piece of encrypted data, identification of steganography and revealing the hidden data, identification of a suspect communication between two computers and reveal the stolen data, identification of the incorrect usage of cryptographic algorithms usage and finding the private key used for signing, identification of host to host wireless communication, interception of host to host communication

The paper is organized as follows. Section 2 is dedicated entirely for the newer botnets and malware threats that are intended for desktops and industrial computer networks. During section 3 we briefly talk about some of the most used tools in case of cyber security investigations and we present some practical scenarios used in cyber trainings. Finally section 4 of the paper contains conclusions and an outline of the directions for future research.

II. CYBER THREATS THREATHENING COMPUTER NETWORKS

In this following section we will talk about the most important and most often encountered cyber threats. We will present the concept of zero-day attacks, that stands the ground for every great cyber attack, and continue with botnets and malware.

A. Zero day attacks

A "zero-day" threat represents a computer or network oriented software menace that aims to exploit weaknesses or other possible points of failure that can grant full access to them. We call them zero-day because they are used by attackers before they are patched. and the developers of the affected application had no days available for patching it.

These attacks come in a large number. Malware, viruses and Trojans all represent attacks vectors that target modern software and delivery networks. In this equation the web browsers and the operating systems on top, which they are running, represent the most widely targets because they are widespread on all devices, starting with mobile phones and ending with desktops. Mail delivery networks are also targeted because they can carry to a potential victim an infected e-mail attachment. To cope with these threats organizations like US-CERT [2] and Zero Day Initiative [3] dedicate their work in providing users cyber security.

Since the vulnerabilities haven't been yet reported and fixed there is no way to protect ourselves from it before it happens. Of course, methods and procedures for early detection exists, like: the use of VLAN's with IPsec to protect the content of an individual transmission, the use of Intrusion Detection Systems, the use of network access control to protect from rogue machines that connect to a certain network.

B. Botnets

We call "botnet", a series on interlinked computers, or even an entire computer network, that are used by attackers in their own personal interests. The most important fact to notice here is that the affected computers are used without the legit users knowledge. The term stands for the fact that the affected computer becomes a "zombie", or "robot". Reports from well knows security companies like Symantec and Kaspersky Labs reach a common conclusion: botnets currently are the biggest threat to the Internet.

Computers that are used inside a botnet are those whose owners fail to provide effective firewalls or other safeguards from the Internet. Furthermore, we see that an increasing number of home computers benefit from high speed Internet connections, thus aiding the efforts of the attackers. A bot is a program attached to one of the computer ports that is left open and through this port a remote program can connect to it.

One example is the usage of a botnet to redirect HTTP traffic to another specific computer or website, in a Distributed Denial-Of-Service (DDoS) attack. The remote website will be closed down because it cannot handle all the traffic.

Another example is the DNSChanger bot. This is a Domain Name System (DNS) hijacking Trojan and it was distributed over the Internet as a download claiming to be a video codec needed to view video content on bait pornography sites. Once installed it modifies the target DNS configuration to point to bogus servers over the Internet operated by an Estonian company called Rove Digital and its hosting subsidiary Esthost. Until today it is estimated that the number of infected computers around the world reaches a large number – over four million workstations. And since it is implemented in a cross-operating system manner, it affects Microsoft Windows, Linux distributions and Apple MacOS. What it does is very simple. It installs itself on the target computer and redirects traffic incoming traffic to spoofed websites and servers across the Internet. This traffic was made to IP addresses falling into the following ranges: 85.255.112.0 through 85.255.127.255, 67.210.0.0 through 67.210.15.255, 93.188.160.0 through 93.188.167.255, 77.67.83.0 through 77.67.83.255, 213.109.64.0 through 213.109.79.255, 64.28.176.0 through 64.28.191.255.

C. Malware

Malware represents the software used or created by hackers to alter computer and system operations. The goal is to gather sensitive information or to gain access to private computer systems. Its form varies from a full software program to a script. It is a general term that is used to refer to all forms of hostile and intrusive software, like computer viruses, Internet worms, Trojan horses, spyware, adware, and rootkits.

In the evolution of malware we can establish two big periods: before 2010 and after 2010. Malware before 2010 was mainly targeted to single computers or medium-sized computer networks. Since 2010 we can find the so-called "modern malware", which now has migrated from personal computers to large systems, even critical industrial systems.

We will present on the following paragraphs one of the main treats in this field, responsible with industrial systems malfunction called Flame.

Flame, also known as Flamer or sKyWiper [4] is a computer malware that attacks computer and industrial systems running Microsoft Windows operating system. It was used for regular and critical infrastructure penetration in several countries around the world [5].

The spreading mechanisms are in great number and they include local area network infection and Bluetooth enabled devices. The data recorded is also various and it includes information such as audio files, screenshots, instant messaging conversations, and basically all that can be recorded from a computer. This data was later sent to different servers through the Internet.

An interested feature of Flame was its capacity to remove itself from the infected computers. This command was caught by the Symantec Security Company using computers set up to watch the malware's actions. More exactly, when it received that command, Flame located every file existing on the victim PC, deleted it and then overwrote its memory location with random data to prevent a forensic examination.

According to cryptographic experts [6] [7], Flame was the first malware to use a rather obscure cryptographic technique called "prefix collision attack". This allowed it to fake digital credentials that had helped it to spread. The exact method for this kind of attack was demonstrated in 2008, but the creators of Flame implemented their own variant. This determined Marc Stevens to state that "the design of this new variant required world-class cryptanalysis". All these findings give support to claims that Flame must have been built by a nation state rather than cybercriminals due to the large amount of time, effort and resources that have been put into its creation.

III. SCENARIOS AND TOOLS USED IN CYBER FORENSIC

Attacks on IT systems can be handled using tools and methods common to the field of classic computer networks and regular computing systems. As stated in the first section, it is common in this field to use of scenarios. These scenarios represent the main way in which people involved in incident response can be trained. Often these scenarios are presented to the regular people in a masked form. This means that an expert handling a modern transport system can use in case of incident handling and response the same tools as an expert in computer networks.

In this part we will present some of these tools, along with their capabilities. Also, in order to be more relevant to the reader, we present these tools in action, applied to the "Cyber MITRE" international challenge mentioned before.

A. Tools

In general, the tools used in incident response tools are designed as a general application, which can have multiple other uses. We are going to briefly talk about three main representative categories: stand-alone tools, statistical tools and security oriented distributions.

1) Stand-alone tools

Stand-alone tools represent tools that can be used independently over an operating system. Representative to this field is CrypTool. This is a software package dedicated to cryptographic simulation, analysis and cracking which has a user graphic interface.

CrypTool has been developed in cooperation with prestigious universities and thus has become excellent educational software and also a tool for learning cryptology. CrypTool covers both branches of cryptology: cryptography and cryptanalysis. Thus, the product has implement facilities of each field, such as classic cryptography (Caesar and Vigenère ciphers, monoalphabetic substitution, etc.), symmetric cryptography (IDEA, RC2, AES, etc.), asymmetric cryptography (RSA and elliptic curves, etc.), hash functions (MD2, MD5, SHA-1, etc.), cypher text attacks, plaintext attacks, adaptive attacks, side channel attacks.

2) Statistical tools

In order to test the degree of randomness for input or output for such transportation systems, we need a set of different tools, tests and theoretical models – we need statistical tools. Using this kind of tools we can test for true randomness of functions that are part of the software implementation of these systems. Good examples in this direction are:

- "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications" [8], is a publication of sixteen statistical tests. The authors provide also an implementation for it.
- "The Art of Computer Programming, Seminumerical algorithms, Volume 2", by Donald Knuth [9] contains the theoretical description for some of these tools, that are based on permutations, birthday spacing, serial correlations, etc.
- The Crypt-XS suite developed by the researchers from Queensland University of Technology, Australia. This suite contains the implementation for some tools described by Knuth, along with other custom ones for binary derivative, sequence and linear complexity measurement, etc.
- The DIEHARD suite [10] developed by George Marsaglia adds to the tools mentioned before, tests such as random spheres, overlapping sums, etc.

3) Security oriented distributions

Security oriented distributions, such as BackTrack Linux (BTL) [11], [12] represent a collection of great tools used for network penetrations testing and security audit in general. BTL is a distribution built on top of Debian Linux and it comes with all its

advantages and disadvantages. Among many tools we can find even well known applications, such as wireshark or zenmap.

BTL contains more than 300 security tools and utilities that are all open source, grouped in major categories like: information gathering, vulnerability assessment, exploitation tools, privilege escalation, maintaining access, reverse engineering, RFID tools, stress testing, forensics, reporting tools, services.

B. *Practical scenarios*

Of course, to use such tool, the incident response experts must know how to use them, and especially what tool fits better in a context. In order to cross this border different sets of tests have been created. Next we will present five of the most important and most used one, that have also a great impact on today's intelligent transport systems: finding what kind of encryption is used over a system, identifying what data is leaked from or into our system computer network, identifying and intercepting eventually live transmissions between an infected host from our network and its malicious command and control servers over the Internet, identifying and recognizing fake or mangled signatures keys for public certificate access and finally, in case of using large transport system deployments with many nodes and a lot of intermediary wireless communication, identifying the weakest points that can be used by attacker to gain unauthorized access to the system.

1) *Identification of encryption system*

In this scenario the investigator finds a suspect file on the system or computer. This file represents an encrypted data with a classical encryption system. We need to recognize encryption system, decrypt the data and find the password hidden in the encrypted file. The investigator will need this password in solving the second task.

To solve this scenario we will use CrypTool. If we take a look at the encrypted file we see that this file contains only 26 characters A to Z. If we perform a statistics of these letters we see that the characters A-Z appears to be random. Thus, we can think that there is a classic encryption such as substitution (Playfair, Caeser, Vigenère etc.). Using cipher text only attack on a Vigenère cipher we find the encryption password <<SQUARE>>. If we take a look at the end of the decrypted file we find the text THEPASSWORDFORTOMMOROWISSTRONGPASSWORDSAREGOOD. Thus, we conclude that the password we are looking for is: STRONGPASSWORDSAREGOOD.

2) *Identification of hidden data inside other files*

Over the investigated system network have been intercepted some images. One of these images has a huge size reported to its format. We need to find the data founded in the image and decrypt it if necessary.

We start by analyzing the intercepted image. It is in gif format but has a huge size, approx. 13 MB. If we investigate this image with a hex editor like UltraEdit we can see at the end of the file a zip file header (PK). Thus, if we exchange the extension of gif file to zip file and open it with a supported archiver like WinZip we find an encrypted archive. After cracking the password, we find three gif files. Opening each file we can see that one of these one has the content: hollenger.dll. Thus, the malicious file name that the attacker are trying to access is hollenger.dll.

3) *Identifying a suspect communication between two computers and reveal the stolen data*

By analyzing our system network we find regular traffic between a node and another host over Internet. The local administrator gives to the investigator a traffic capture between the node and an outside unknown source. The task is to find what information is stolen.

Using the Wireshark application we are opening the target file. This tool is used for network traffic analysis or any other general network troubleshooting. Using Follow TCP stream option we locate within the capture a file found on the source node and dump it into a file. Opening the dumped file with a hex editor like UltraEdit we see the zip file header (PK); thus we change the extension of the dumped file into zip and open it. This archive contains a file and after a visual inspection with UltraEdit we find that it contains the magic header GIF. Thus, if we exchange the extension of the file into gif and open it with an image viewer we can see an image which has the content The Root Password is Pengu1nsR0ck.

4) *Identifying and recovering mangled signature keys*

In this scenario we are given two ECDSA (Elliptic Curve Digital Signature Algorithm) signatures. After a close inspection we find out that something about them looks strange. Using the known public key and its public parameters we must find a way to recover the private key used to generate the signatures.

Before presenting how an investigator can resolve the scenario, we will present briefly the concept behind ECDSA. Basically, ECDSA is a variant of DSA algorithms that use elliptic curve cryptography in order to be more reliable. The public parameters are the prime number p, an elliptic curve $E[F_p]$, a point $G \in E[F_p]$ with ord(G)=q, q prime number. The public key $V \in E[F_p]$ is derived from the signing key $1 \le d \le q-1$: $V=dG$. The signature of the hash h is computed using the ephemeral key k mod q is the pair $(r,s)=(x_{kG} \bmod q, (h+dr)k^{-1} \bmod q)$, where x_{eG} is the first component of the point $eG \in E[F_p]$. To verify the signature (r,s) of the hash h we need to check if $x_{v1G}+v_{2V} \bmod q=r$, where $v1=hs^{-1} \bmod q$ and $v_2=rs^{-1} \bmod q$. It is essential to have for different signatures (r_1,s_1) and (r_2,s_2) different ephemeral keys $k_1 \neq k_2$. If this two keys are equal then the signatures of the two hashes looks like (r,s_1) and (r,s_2). Thus, we can derive $s_1-s_2=k^{-1}(h_1-h_2) \bmod q$ and find the ephemeral key $k=(h_1-h_2)(s_1-s_2)^{-1} \bmod q$. Since $s_1=k^{-1}(h+dr) \bmod q$ we derive the private key $d=(s_1k-h_1)r^{-1} \bmod q$.

The investigator receives three files. The first file contains the hash, in hex codification, of two messages h_1, h_2 and their ECDSA signatures (r_1,s_1) respectively (r_2,s_2) :

h_1=DE37B3145DB7359A0ACC13F0A4AFBD67EB496903

r_1=ACB2C1F5898E7578A8A861BDF1CA39E7EF41EAC0B6AAA49468DD70E2
s_1=BE4FA99C9D261C5F387A3ACE025702F6FB7884DD07CE18CAD48654B8
h_2=28469B02BF0D2CFC86FF43CB612EE8FC05A5DBAA
r_2=ACB2C1F5898E7578A8A861BDF1CA39E7EF41EAC0B6AAA49468DD70E2
s_2=D3540E2B13E51605F5FEB8C87EE8E176E59213F31EA8B8FFDAD077E2

The second file parameters.der contains, in der codification, the public parameters of the EC. This file can be interpreted using OpenSSL:

openssl ecparam -inform DER -in /cygdrive/e/parameters.der -outform PEM -out /cygdrive/e/parameters.pem

openssl ecparam -text -in /cygdrive/e/parameters.pem -noout

Field Type: prime-field
Prime:
00:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:ff:fe:ff:ff:e5:6d
A: 0
B: 5 (0x5)
Generator (uncompressed):
04:a1:45:5b:33:4d:f0:99:df:30:fc:28:a1:69:a4:67:e9:e4:70:75:a9:0f:7e:65:0e:b6:b7:a4:5c:7e:08:9f:ed:7f:ba:34:42:82:ca:fb:d6:f7:e3:19:f7:c0:b0:bd:59:e2:ca:4b:db:55:6d:61:a5
Order:
01:00:00:00:00:00:00:00:00:00:00:00:00:00:01:dc:e8:d2:ec:61:84:ca:f0:a9:71:76:9f:b1:f7
Cofactor: 1 (0x1)

The third file, public.oct, contains the public key:

X_V=85CEEE9C98EFDFDFCF64CB522A773F1435D568173677D1D28FC00643

Y_V=58A105CC1AB1A53D77B278850776E144197F3FA4E27AA676408DFE22

At this point, because the two signatures collide on the first half, we have all the elements to finalize the investigation. The only thing we need to do is to compute the private key using the formula:

$d=(s_1 k-h_1)r^{-1} \bmod q$,

where $k=(h_1-h_2)(s_1-s_2)^{-1} \bmod q$. We can perform these computations using, for example, MAPLE:

```
h1:=convert("DE37B3145DB7359A0ACC13F0A4AFBD67EB496903",decimal,hex);
h2:=convert("28469B02BF0D2CFC86FF43CB612EE8FC05A5DBAA",decimal,hex);
r:=convert("ACB2C1F5898E7578A8A861BDF1CA39E7EF41EAC0B6AAA49468DD70E2",decimal,hex)
s1:=convert("BE4FA99C9D261C5F387A3ACE025702F6FB7884DD07CE18CAD48654B8",decimal,hex);
s2:=convert("D3540E2B13E51605F5FEB8C87EE8E176E59213F31EA8B8FFDAD077E2",decimal,hex);
q:=convert("01000000000000000000000000001DCE8D2EC6184CAF0A971769FB1F7",decimal,hex);
d:=(-h2*s1+h1*s2)*(r*(s1-s2))^(-1) mod q;
convert(d,hex,decimal);
```

After the compilation of the program we find the private key:

d=8E88B0433C87D1269173487795C81553AD819A1123AE54854B3C0DA7

5) Identification of insecure wireless connection points

In this scenario the investigators are trying to find what wireless access points are unsafe in public use and they are trying to gain access to a private Wi-Fi network that is secured using the WPA (Wi-Fi Protected Access) protocol. WPA is a security protocol and security certification program developed by the Wi-Fi Alliance. It is known as the IEEE 802.11i standard. More exactly a Temporal Key Integrity Protocol (TKIP) is used, that involves using a dynamic 128 bit key for every packet transmitted. The newest version, WPA2 also includes Counter Cipher Mode with Block Chaining Message Authentication Code Protocol (CCMP), a new AES based encryption mode with strong security.

In order to solve this scenario, the investigators will use a series of intercept nodes that will have a single goal: to continuously scan the entire wireless networks available and try to crack their password. For this, all the intercept nodes will run two tools called Reaver and Airmon-ng.

Reaver is a toot that uses a brute-force attack targeting Wi-Fi Protected Setup (WPS) routers. It targets the router PIN, because cracking the PIN gives it access to the router's WPA or WPA2 connection password. It is an attack that can be used to target even well known network solutions vendors, such as Cisco. Exact details of implementation can be found in http://sviehb.files.wordpress.com/2011/12/viehboeck_wps.pdf.

During its implementation, this tool was tested against various router models from different vendors and the authors came to the conclusion that is necessary for about four to ten hours in order to crack a router's PIN together with its WPA or WPA2 passphrase. As mentioned before, this tool takes advantage of a vulnerability existing in WPS. WPS is a feature that exists on many modern routers and its intention is to provide to the users an easy setup process. The problem is that it is tied to a PIN that is hard-coded into the device.

The first step is to set our node into a monitor mode using the command airmon-ng start wlan0. After this, we need to find the BSSID of the target node that we want to test. The BSSID is a unique series of letters and numbers that identifies a target router. We find it by using the command airodump-ng wlan0. In the list shown inside the terminal we copy the one for our network. We will assume the following made up BSSID: 8D:AE:9D:65:1F:B2.

Now, with the BSSID and the monitor interface name in hand we have everything we need to startup Reaver. Inside a terminal we issue the command reaver –i mon0 –b 8D:AE:9D:65:1F:B2 –vv After this, Reaver will try a series of PINs on the router in a brute force attack, one after another. After cracking is complete we will have the output of the total time

needed to crack the password, the router PIN and its password.

IV. Conclusions

As we can see from this paper the problem of cyber threats represents a current and menacing threat, which involves strong knowledge of computer communications techniques, secure programming techniques, algorithm and software implementation analysis, cryptography, steganography, probability and finally applied mathematics. The process of high level of assurance of cyber security must take into account all the above-specified domains.

Fast advances in cybercrime technology and techniques have resulted since the beginning of 2012 in an unprecedented rise in data breaches. We think that planning to ensure that our computer networks are trustworthy and secure so we need to consider the fundamental changes that are occurring in the cyberspace and try to adapt to them. In our opinion, looking forward into the future of more than 3 billion Internet users existing today we can see four big directions for resolving the cyber security issues: online users security education, cryptography, online data obfuscation and cloud services transparency and security.

References

[1] http://www.gambit-project.org/ [accessed on February 28, 2012]

[2] http://www.us-cert.gov/ [accessed on February 28, 2012]

[3] http://www.zerodayinitiative.com/ [accessed on February 28, 2012]

[4] http://www.crysys.hu/skywiper/skywiper.pdf [accessed on February 28, 2012]

[5] http://www.certcc.ir/index.php?name=news&file=article&sid=1894&newlang=eng [accessed on February 28, 2012]

[6] https://www.securelist.com/en/blog/208193522/The_Flame_Questions_and_Answers [accessed on February 28, 2012]

[7] http://www.bbc.co.uk/news/technology-18365844 [accessed on February 28, 2012]

[8] A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications, 800-22, NIST Special Publication, 2010

[9] Knuth, D; The Art of Computer Programming, Seminumerical Algorithms (3rd ed., Vol. 2). Addison Wesley, Reading, Massachusetts, 1998.

[10] Marsaglia, G.; DIEHARD Statistical Tests (accessed on February 28, 2012 on http://stst.fsu.edu/geo/diehard.html)

[11] Patrick Engebretson; The Basics of Hacking and Penetration Testing, Syngress, 2012

[12] http://www.backtrack-linux.org/ [accessed on February 28, 2012]