Journal of Advanced Computer Science & Technology, 5 (1) (2016) 14-22

Journal of Advanced Computer Science & Technology

Website: www.sciencepubco.com/index.php/JACST
doi: 10.14419/jacst.v5i1.6181
Research paper

Cybersecurity Snapshot: Google, Twitter, and Other Online Databases

Bharat S Rawal *, Gabrielle Eberhardt , Jaein Lee

IST Department, Penn State Abington, PA 19001 USA
**Corresponding author E-mail: brawal@psu.edu*

Abstract

Every day, millions of attacks are carried out on the networks and computer systems. In recent years, these numbers have increased dramatically. All it requires is one success for a hacker to gain unauthorized access and data, but for administrators, it is a constant battle to protect what is rightfully theirs. In this paper, we look into how these attacks have increased, what the studies of various databases and reports say on how and what types of data are being breached, who is breaching them, and how they are breaching the systems. Also, we propose various unconventional ways to prevent these attacks from happening in the future. Furthermore, this paper lists the top 26 bug-fix times reported in the Google Security Research Project (GSRP). This article brings to light reoccurring cyber threats, challenges associated with these threats, and emerging trends in the domain of cyber security.

Keywords: *Cyber-Attack; Availability; Vulnerability; Mandelbug; Heisenbug.*

1. Introduction

The cybersecurity research that has been the subject of much attention in recent years is that of cyber-attack detection systems. Cyber-attacks are activities that try to circumvent the security tools of computer systems. Cyber-attacks are any set of activities that impend the reliability, accessibility, and privacy of a network and system resource. Patricia et.al. describes a cyber-attack as a combination of devices, procedures, security notions, security protections, guidelines, contingency command procedures, activities, training, conventional practices, assurance, and technologies that can be used to guard the networked environment and system and user's assets [48]. In recent years, organizations are becoming more and more susceptible to possible cyber threats such as system cyber-attacks. There is an increasing need to preserve the security and safety of information systems using various security implementation. However, for most systems, complete attack prevention is not realistically attainable due to system complexity [46]. The reality is that there is no easy or perfect answer to a cybersecurity threat. Cyber security as an issue is too comprehensive; there are too many variables in security and too many known and unknown vulnerabilities in hardware and software [47]. In this paper, we provide a comprehensive overview of cyber security by describing both attacks and vulnerabilities. A failure is an inability for anything, computer (technology) or person, to perform a normal function. This applies to systems as well [1]. When it comes to software, a failure occurs when the person or thing using the computer or program believes that it can do something that it may actually be incapable of doing or performing. The function is expected, so the person then describes it as a failure [2]. A fault is a problem or mistake that prevents something from being perfect [1]. Faults, about computers, can be described as bugs or "hiccups" in the system [3]. The vulnerability can be defined as something that is susceptible to attacks or weakness that somebody can take advantage of [4]. When it comes to computers, vulnerabilities occur with input errors, buffer overflow, software or hardware malfunctions. Across all business industries, about 96 percent of systems have been breached [5]. In America alone, our enterprises and industries are attacked on average 100 million times a day [6]. With computer technology increasing every day, hackers and their strategies are by far outreaching our strategic efforts to protect this technology. The numerous vulnerabilities that computers have and the fact that hacker knowledge is increasing increases the constant struggle and breaches of technological security in this day and age. From statistics so far, it can only get worse. The cyber security market will be worth approximately 170.21 billion dollars by the year 2020 [7], [47]. Various studies have proposed models for future cyber-security threats and are based on the time series and moving average, Hidden Markov Model, and the State Based Stochastic Model.

The remainder of the paper is organized as follows. Section II presents related work; Section III describes vulnerability classification; Section IV provides difference between intrusion tolerance v/s fault tolerance; Section V provides attack taxonomy, comprehensive classification; Section VI discusses security objectives; Section VII shows past data breaches reported from Jan 2005 to Feb 2016; Section VIII discusses on google security research; Section IX produces various software patching time for bugs in the google security research project; Section X discusses a study of recent tweets; Section XI presents anticipation for cyber threats; Section XII suggests the additional solutions for cyber threat. Section XIII concludes the paper.

2. Related work

Many classifications of faults have been seen and documented over the years and can be considered in the article [17]. Orthogonal Defect Classification [18] identifies weaknesses in programs throughout the system that contain faults and remove them [19-22]. The past few years, types of attack of this classification have risen

tremendously, and much effort is going into trying to find a fix. The head company in this project is Verizon with help from seventy other organizations from all around the world [23]. Because the data breaches are becoming very common, this group started to investigate. One common question that Ponemon has asked executives is how companies prepare for data breaches, in that they are inevitable [24]. Many organizations document breaches affecting groups of many people to see if any of the data collected can help in the future. One group, Breach Portal Notice to the Secretary of HHS Breach of the Unsecured Protected Health Information U.S. Department of Health and Human Services Office for the Civil Rights report, documents all data affecting more than 500 people involved with health services. One work cited, Ocrportal, accurately describes the types of breaches and the devices that are used to accomplish violations [25].

In 2014, the percentage of phishing in firms with more than two thousand five hundred people went up to 5 out of 6. That is a 40% increase [26]. In this day and age, the threat to security on the internet is real, and numerous documents have been made to stress this importance [27], [28]. In [27] the paper describes that there needs to be an answer to end all threats to the computer. The world needs better security training, software, and more people need to get involved. Technology is the basis for everything today, without it, our country and the world, may not be able to process or function [29].

3. Vulnerability classifications

Vulnerabilities that result from mistakes in human error are not faults. Vulnerabilities are not always bugs, and not all faults are weaknesses [3]. The most significant difference between the two is that vulnerabilities are exploitable, while faults are not [8]. When it pertains to cyber-security, vulnerability is a fault that allows an invader or a hacker to attack that system and decrease its information and cyber-security assurance. Vulnerabilities consist of three components: system weakness or defect, access to the fault, and the capability to exploit the error or flaw [9].

3.1. Popular vulnerability and attack types

Howard et al. described vulnerability classifications and they can be useful to find out what exactly went wrong [10]. Weber S, Karger PA, and Paradkar believe that classifications are necessary no matter if they are useful or not [11]. Described below are several classification techniques:

1) Classification by Software Development Cycle: Vulnerabilities classified by when the software life cycle was created [12].
2) Classification by Genesis: Flaws can be split into intentional or unintentional faults. Intentional defects are divided into malicious and non-malicious [11], [13].
3) Classification by Location in Object Models: Vulnerabilities are grouped based on which entity they belong to [14].
4) Errors or Mistakes: The errors are triggered by vulnerabilities; the effect and alteration are made to get rid of the error [15].
5) Classification by Enabled Attack Scenario: Vulnerabilities that are descriptive and precise enough to let a set attack happen. Cross Site Scripting allows an injection of a virus or malicious content into your data or information [16].

3.2. Classification of software systems faults

Grottke et al. tried to define software systems faults more precisely based on manifestation and complexity. They classify defects into four classes; “Bohrbugs,” “Heisenbugs,” “Mandelbugs,” and “aging-related bugs.” [49-51].

1) Bohrbug: An effortlessly quarantined fault that regularly manifests. It is easily reproduced because its activation and error propagation lack “complexity”.
2) Mandelbug: A fault whose stimulation or error spread is complex. "Complexity" can be caused by the influence of indirect factors, such as the time lag between the fault initiation and the manifestation of a failure. It can also be triggered by interactions with the software application with its system-internal environment or influence of the timing of inputs and processes.
3) Aging-related bug: The initiation or error spread of the fault is subjected to the total time the system has been running. The aging-related bug is capable of causing an increasing failure rate and/or degraded performance.
4) Heisenbug: In computer software design, Heisenbug is a grouping of an infrequent programming bug that vanishes or changes its behavior when an effort to quarantine it is made. Mandelbugs are soft [52] or vague [53] bugs, which Gray [52] also consider as Heisenbugs.

3.3. Tolerant system

An intrusion-tolerant scheme permits for a limited probability that the system’s security may be breached. An intrusion tolerant system will be capable of sensing either the onset of a security outbreak into a system or the inevitable security failure because of such an attack and consequently respond to such an occurrence in a manner that invalidates the hostile effects of an explosion. The strengthening and protection mechanisms use various approaches such as authentication, access control, encryption, firewalls, proxy servers, etc. If the scheme fails during the penetration and exploration phases, the system needs to reset from a real state G into the vulnerable state V [30]. If a vulnerability is exploited successfully, the system will enter the active attack state A. Intrusion tolerance is activated where intrusion resistance fails. For fault tolerance techniques, there are four phases, namely error discovery, damage calculation, error recovery, and fault management [30]. Some strategies for recognizing the outbreaks and assessment of loss comprise of intrusion, logging, discovery and assessing.

4. Intrusion tolerance v/s fault tolerance

In today’s complex software systems, despite employing the best software engineering practices, it is practically impossible to eradicate all faults or bugs that will ultimately cause the software to fail in its operational phase. An alternative to making complete fault free software is to build a fault-tolerant software that guarantees total recovery from failures. Intrusion tolerance is to increase fault tolerance and may be a real-world substitute to constructing secure software systems [30]. The progress of a software system near catastrophe due to accidental errors or security weaknesses is shown in Fig 1.

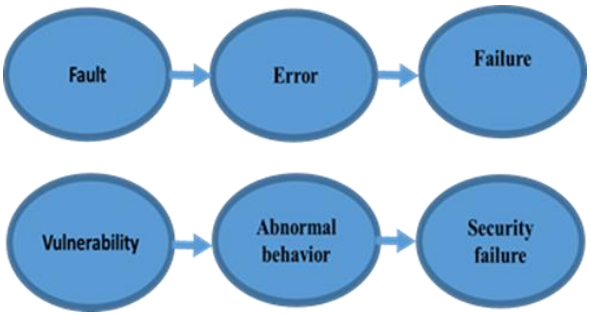

Fig. 1: Failure Progression [30].

5. Attack taxonomy comprehensive classification

Dorottya, Ma, and Buttyan describe that if a device is attacked, one or more of the following must be true [31]:

Internet Accessible: A remote attacker can exploit vulnerabilities in the CVE if the device can be accessed via the Internet. The attacker merely has to discover the device and send packets to it.
Some form of remote access to the device: The attacker has some form of privilege to some of the device's services. The level of access required can be normal user privileges, not even requiring administrator privileges.
Direct physical access to the device: An attacker, given direct access to the device, will often not require any form of rights to perform an attack on the instrument.
Miscellaneous: An attacker takes advantage of certain software vulnerabilities in the device or exploits a particular device configuration.
Unknown: Some devices do not provide a sufficient amount of information to determine the characteristics of the attack.
Papp, Dorottya, Zhendong Ma, and Levente Buttyan systematized the vulnerabilities in the following ways [31].
Programming errors: Programming errors can lead to vulnerabilities being created on the input device that can be exploited by attackers. Such vulnerabilities include buffer overflow issues and pointers that were never freed.
Web-based vulnerability: Devices can be attacked through unpatched web server applications.
Weak access control or authentication: Devices have weak passwords or default passwords that are easily cracked. Furthermore, some devices possess hard-coded passwords that provide administrative access.
Improper use of cryptography: Some devices utilize weak cryptography improperly. They employ weak random numbers as keys or primitive types that are easily obtained.
Unknown: Some devices do not reveal information about the vulnerability used to attack the device. The methods by which cyber-attacks are perpetrated are [31]:
Control hijacking attacks: Attackers send commands executed remotely that are performed by the computer. The user may be using the computer and never notice the attack happening in the background.
Reverse engineering: Some attackers take a device and analyze it thoroughly. Such analysis can reveal vulnerabilities that can be used as a vector for an attack.
Malware: An assailant can try to infect a fixed device with malicious software (malware). Malware that affects an embedded device may modify the behavior of the device, which can have consequences in the physical world.
Injecting crafted packets or input: Attacks can consist of injections of crafted packets or inputs into a program embedded into a device.
Eavesdropping: Eavesdropping is a passive attack method where the "attacker" intercepts and analyzes packets sent to and from the device.
Brute-force search attacks: Attackers can utilize brute-force search attacks to guess a password or vulnerability by trying every single possibility. This is often used when the desired result has a limited range it could be in.
Routine use: Some devices do not have any protection. Thus, an attacker can use the device as if he was a regular user; allowing for any of the previous attack methods to be used.
Unknown: Some devices describe vulnerabilities but do not identify any particular attack method that would exploit those identified weaknesses.
Denial-of-Service (DoS): Some devices fail to work after an attack, preventing analysis of what caused the breakdown.
Code execution: Some devices may run malicious code on the instrument even after the attack was carried out.
Integrity violation: The device may have altered firmware or data. The integrity of the data is called into question.
Information leakage: Information on the device can be obtained by the attacker.
Illegitimate access: The attacker gains additional privileges on the devices, allowing for a deeper form of attack.
Financial loss: Some devices have access to premium services that can be the goal of an attacker.
The degraded level of protection: Another effect of attack can be degraded software protections on the device. Future attacks may increase the effectiveness of the instrument.
Miscellaneous: Some attacks can cause random website redirects. The exact mechanism for the redirects is unclear.
Unknown: In some cyber-attacks, no method is identified.

6. Security objectives

In the following, we cite three high-level security goals as described by Wenye and Lu [29, 32].
Availability: Guaranteeing timely and secure access to the use of information is of the most importance in the information system. This is because a loss of availability is the disturbance of access to the use of information, which may further weaken the expected service delivery.
Integrity: Guarding against improper information modification or destruction is to ensure information nonrepudiation and authenticity. A loss of honesty is the unauthorized alteration or damage of information and can further cause incorrect decision concerning expected service management.
Confidentiality: Certified protective restrictions on information access and disclosure are mainly to protect personal privacy and proprietary information. This is, in particular, necessary to prevent unauthorized disclosure of information that is not open to the public and individuals. From the perspective of system reliability, availability and integrity are the most important security objectives in information systems (databases). Confidentiality is the least critical for system reliability; however, it is becoming more important, particularly in systems involving interactions with customers, such as demand response and AMI networks.

7. Historical threat analysis

We have used multivariate forecasting methods based on modern statistical models, and we attempted to generalize the extrapolation methods to the multivariate case, using time-series models or techniques that are structural or theory-based. There are 896, 258, 345 records of all types breached, and 4753 of those breaches have been made public. There have been 626,832,481 records breached due to malware attack, and 1249 of those data breaches were made public since 2005[33], [47].

Fig 2 represents the average two-month data breaches from 2005 to the year 2016. We can notice that 2016 is the lowest (191260), and data breaches in 2009 are the highest (22839880) reported in the United States.

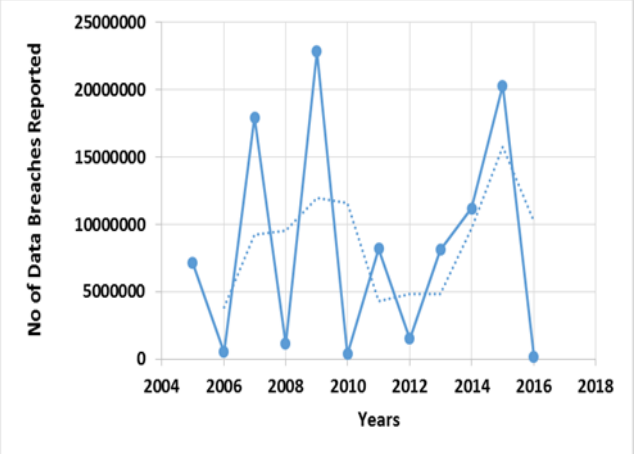

Fig. 2: The average two months' data breaches reported from Jan 2005 to Feb 2016

8. Google security research

Google Security Research Project (GSRP) is an open source database about existing issues in security holes. This paper analyzes said database based on reported security vulnerabilities encountered by users over a period of time. Our analysis is entirely based on comments published in each encountered issue in the database. As of Feb 23, 2016, there were 522 issues reported. Out of the 522, 494 were fixed, five are new, seven are duplicate, six were invalid, and 15 bugs are non-fixable. This paper reports those unknown bugs as OTHER [34].

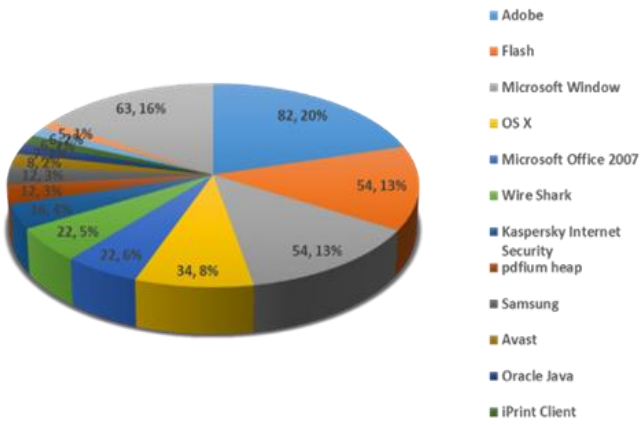

Fig. 3: Bug distribution in various software in the GSRP

Fig 4. shows the distribution of various types of bugs in the Google Security Research database. Bohrbugs represent 85% of the bugs reported. Nonage related MAN bugs represent 7%, Aging-related bugs represent 4%, and the remaining 4% are represented by other (the unknown) types of bugs.

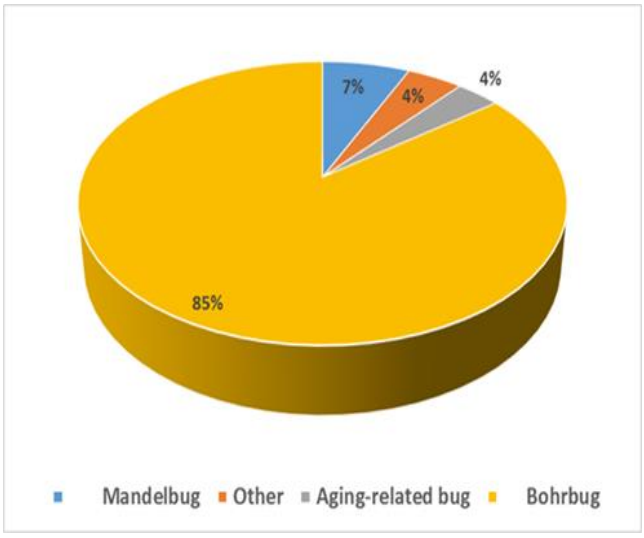

Fig. 4: The distribution of Mandelbugs and Bohrbugs in the GSRP.

9. Repair time for bugs in the Google security research project

Bug fix time is beneficial in numerous areas of software development and evolution, such as expecting software quality or synchronizing progress effort during bug testing. From a particular GSRP, this paper explores the characteristics of the fix time for a bug. We did not consider the significance or severity of particular bugs. Attributes such as the priority and the severity of a bug, have an effect on the fixed time for bugs in big software systems. We have studied various mathematical representations for bug fix times. Fig 5. represents the distribution of bugfix times for reported bugs in GSRP. The highest repair time reported was 191 days for issue id 38("Flash leak of uninitialized data).

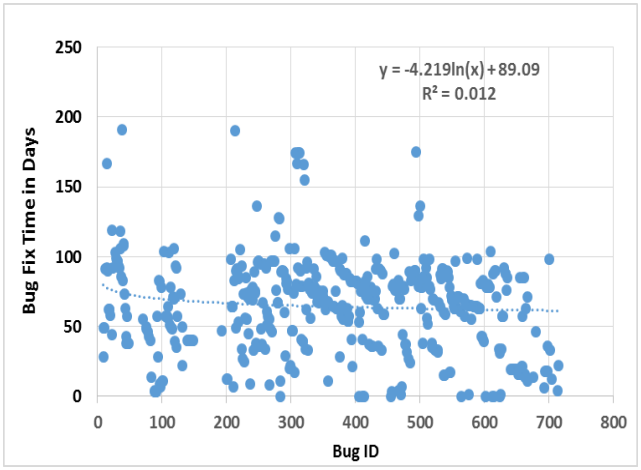

Fig. 5: The distribution of bug fixes time for reported bugs in the GSRP

The lowest fix time was zero days for 11 issues from issue Id 284 (Rowhammer: NaCl sandbox escape PoC),405 Linux: fuse privilege escalation), 411 Linux: privilege escalations via crash analysis frameworks (apport, abrt),412 Linux: a network manager authorization problem with modem config files and arbitrary file read),413 (Linux: missing authentication check in USB-creator leads to local privilege escalation),455 (Placeholder: PoC for cupsd exploit of string reference count over decrement),564 (Kaspersky Internet Security: Network Attack Blocker Design Flaw),602 (FreeType 2.6.1 TrueType parsing, heap-based out-of-bounds reads in "tt_cmap14_validate"),612 (pdfium stack-based buffer overflow in CPDF_Function::Call),614(FreeType 2.6.1 TrueType parsing heap-based out-of-bounds read in "tt_sbit_decoder_load_bit_aligned"[34]),and 623(pdfium heap-based out-of-bounds read inCPDF_TextObject::CalcPositionData). The average fix time for 492 bugs was 66.14 days [34].

From Figs 5. and 6, we can notice that the majority of fix times lie in the interval of 60-89 days. The Mode fix time is 78 days, the Median fix time is 69 days, and Standard Deviation for fix time is 33.16421424. The R^2 value is equal to 0.012. R^2 is a degree of the goodness of fit of the trend line to the data. A value of 1 is considered a perfect fit. The R^2 value of 0.012 indicates that the line does not fit the data at all. This can be because the data is more non-linear than the curve allows, or because it is random [34].

From Fig 6. we can notice that 75 issues were resolved within an interval of 0-29 days. Only 113 problems were fixed within the interval of 30-59, and the highest 203 bugs were fixed in the range of 60-89 days. Two of the lowest numbers of bugs were fixed in the interval of 180-209 days.

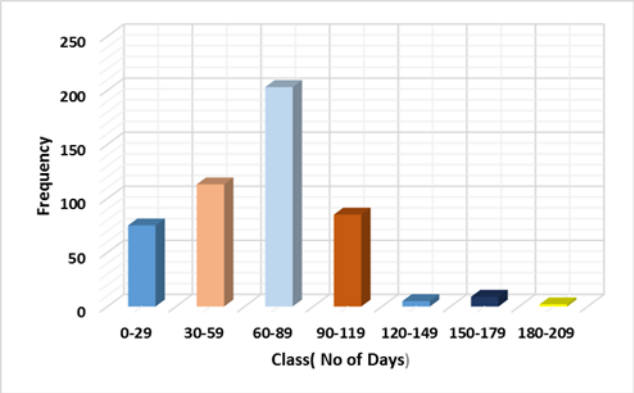

Fig. 6: Frequency distribution for a various range of a bug fix period in days (GSRP).

Fig 7. depicts the cumulative frequency of various classes. In this figure, 494 out of 476 (96.75%) issues were resolved in the interval of 90-119 days.

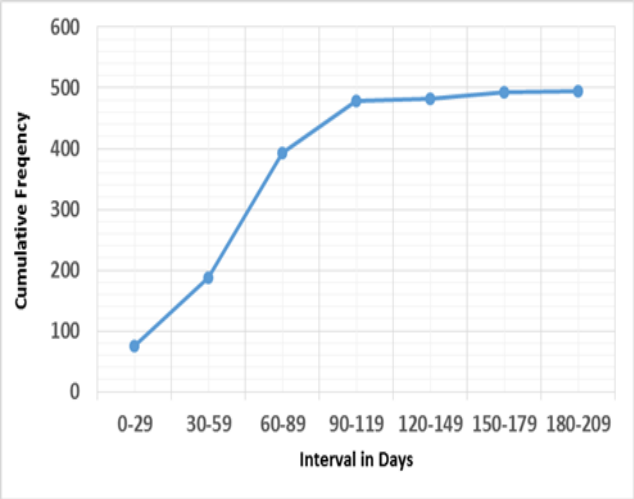

Fig. 7: Cumulative frequencies for various intervals of time periods (GSRP).

Fig 8. shows the pie chart for the distribution of bugs that could be fixed or not fixed. We can see that 94% of reported bugs were fixed within 191 days. The newly reported issues that are not resolved are 1% of the amount of bugs that were recently published in GSRP. 1% were duplicates, 1% if the items were Invalid and 3% bugs were not fixed.

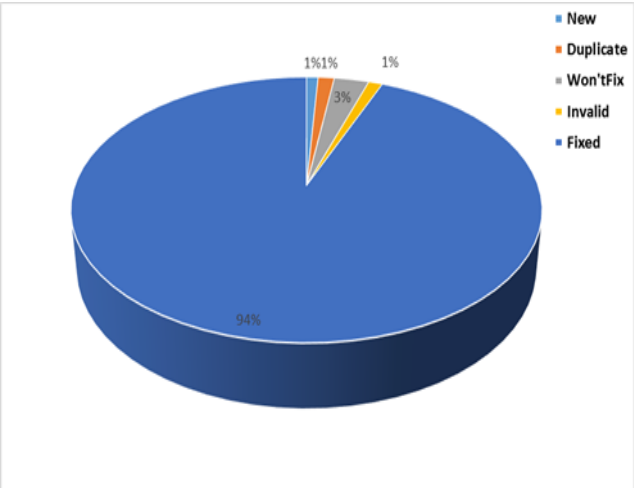

Fig. 8: Distribution of fixable and non-fixable bugs (GSRP)

Fig 9. describes the distribution of 75 bugs those were fixed within 30 first days. Wireshark and Flash related bugs were 29% and 15% of the total bugs respectively.

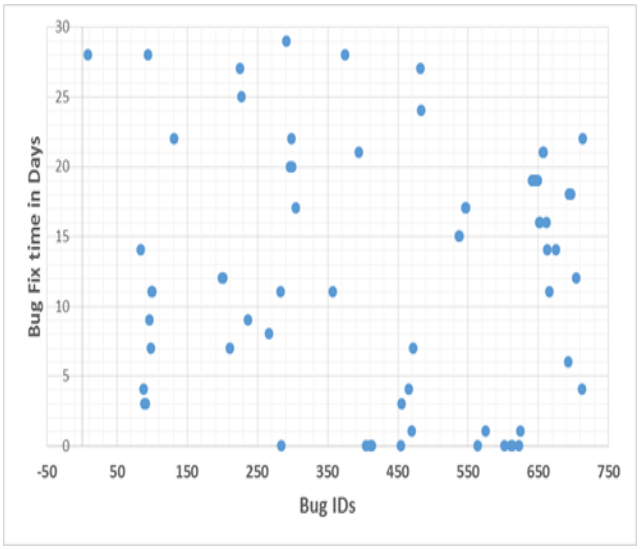

Fig. 9: Bugs which were fixed in 30 days in the GSRP Database.

Fig 10 lists the bug fix times for the top 26 issues reported in the interval of 106-191 days in the GSRP. The average repair time for the top 26 items was 141 days, and the standard deviation was 29.57 days. Moreover, 73% of the top bugs were related to Wireshark

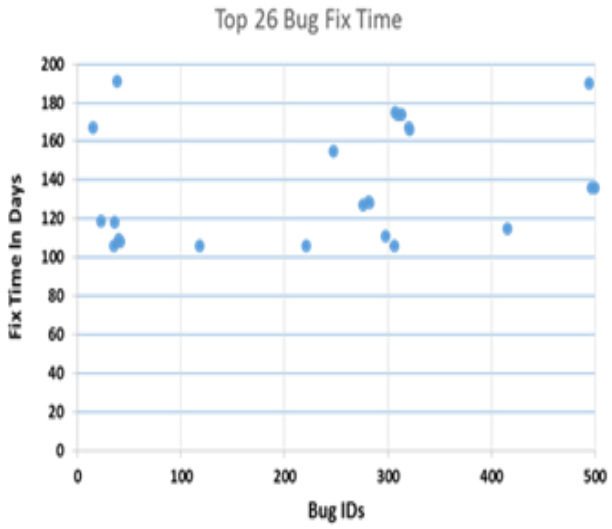

Fig. 10: The bug fix time for Top 26 bugs in GSRP

Fig 11. represents the speed test of the various servers for performance for the First Byte Time (back-end processing) for various servers. The higher the server's CPU utilization, the slower the response time. Moreover, servers are more vulnerable when they are operating at higher CPU utilization. We can see that split-servers offer the fastest response time of 126 MS. The slowest response time is 815MS for Google.

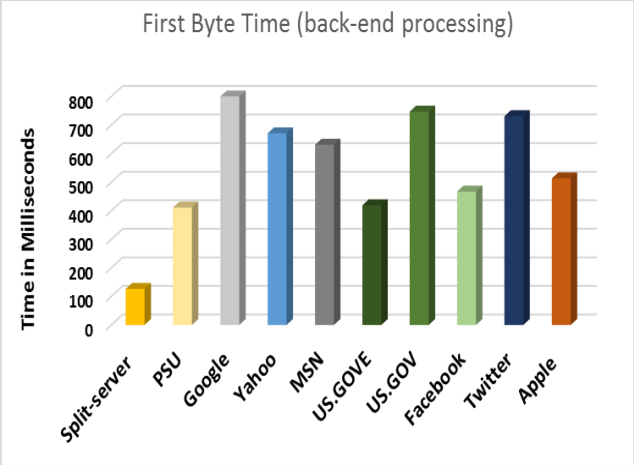

Fig. 11: Seed performance for First Byte Time (back-end processing) for various servers

10. Twitter

Twitter is an online social interacting website and blogging service that allows customers to read and write text-based messages of up to 140 characters, known as "tweets." It was started by Jack Dorsey in July of 2006. Twitter is now in the top ten most visited internet sites [35]. According to Statistic Brain as of September 25th, 2015, there are 58 million tweets per day (672/sec) [35]. In March 2010, cyber criminals used Twitter to disseminate malware using festive-themed messages [36, 37]. In September 2010, thousands of Twitter users, including the wife of former British Prime Minister and White House Press Secretary, were compromised by Twitter cybercriminals [37]. Cyber crooks have exploited Twitter as a new juncture to conduct their malicious actions by comprising, transmitting spam, using phishing scams [38], spreading malware [39, 40], and launching other underground illicit activities. Most of the current methods [41, and 42] focus on disclosing Twitter criminal accounts personally; we still comprehend far less around the properties of those illegitimate accounts' social associations on Twitter.

We have reproduced some mining from recent tweets posted on Twitter's database and compared that to data sets collected from the past fifteen years [47]. As shown in Fig 12, we can see sample data collected from the Twitter in Potomac, MD for 8.00 PM- 9.00 PM from Nov 18 to Nov 24, 2015. If we consider the uninterrupted cases only, we found that the average size of the raw data was 1331.8 MB, general messages data was only 40393.8KB and, the data with security keyword messages was 36KB [47].

Twitter Streamed Data

Date	Raw Data (MB)	Data with only twitter messages (KB)	Messages with only key words (KB)
11/18/2015	596*	16890	21
11/19/2015	1381	40074	46
11/20/2015	1403	44997	40
11/21/2015	1380	43722	30
11/22/2015	1349	36452	28
11/23/2015	770*	22265	30
11/24/2015	1146	36724	36

Note:
1) *During sampling hour, the program sometimes stopped while collecting the data due to a network issue.
2) Location for data collection: Potomac, MD
3) Time: from 8 pm to 9 pm every day during Nov. 18 to Nov. 24, 2015
4) Desktop: Optiplex960 with Intel Core Duo CPU 3.00GHz and RAM 8.00GB as well as Fios network connection

Fig. 12: A typical sample of data streamed from the Twitter [47].

Fig 13. shows the distribution of various cybersecurity related keywords in Twitter streamed data. We can notice that 38% is the majority of the pie slice, and its keyword was related to China. The second highest keyword was 2600, (2600: "The Hacker Quarterly") at 31% of the pie. The rest of keywords resulted in 11%, 6%, 5%, 4% and 2% of the pie. These keywords were on Social media, Worm, Hacker, Virus, and Trojan respectively.

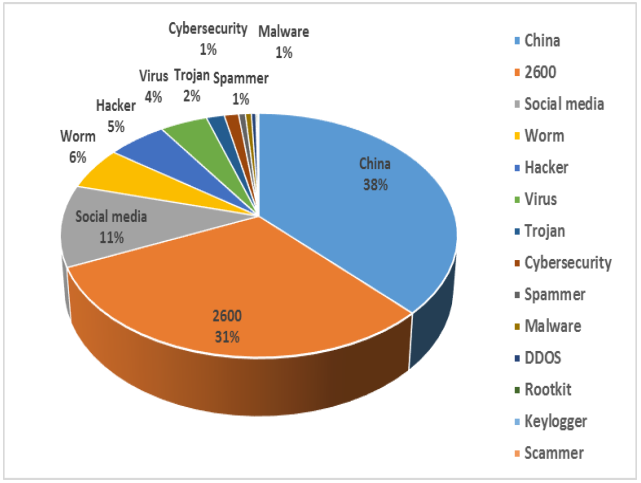

Fig. 13: Distribution bugs in 13763KB Twitter streamed data.

11. Anticipation for cyber threats

Table 1. illustrates the forecast for 2016 of various industry segments. The Monte Carlo method is used for the 2016 prediction; we can notice that all individual industry is showing a drastic decrease in cyber threats compared to the year 2015. The biggest (60%) percentage drop is in MED, and lowest percentage (33%) drop is in the GOV segment. The NGO and EDU's data were not available for the year 2015. We believe that the average of Monte Carlo and Extrapolated values will produce the most accurate prediction [47].

Fig 14. shows typical numbers of attacks for 12 hours in the duration 2/2/2016 -2/11/2016 on FireEye [43]. The average daily count was 349868 attacks. The highest and the lowest attacks were 66346.52 and 18707.07 respectively.

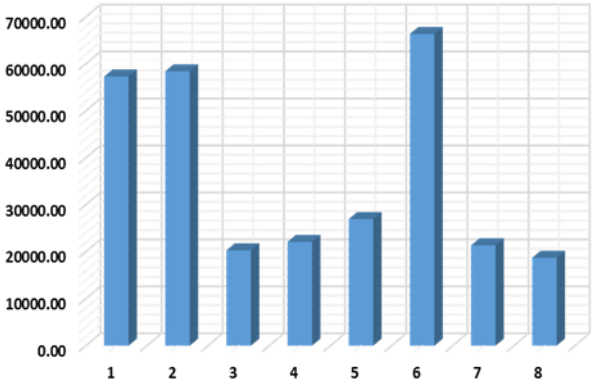

Fig. 14: A typical attack trend for eight days

Table 1: Cyber threat prediction for the year 2016

	Nominal	Min	Max	Number of Cyber Attack in 2015	Stochastic (Monte Carlo) Prediction for 2016	Extrapolation	Mean of Monte Carlo and Extrapolation
BSO	3479462	2274	108685	847200	458375	733727	596051
FIS	5518610	51236	135000550	91000000	83011517	10657338	93668855
GOV	2150000	48045	21500000	21500000	12225994	1758873	13984867
NGO	36243	1272	110000	N/A	52824	36587	44705
MED	1687119	15967	8551626	8551626	3627097	3910632	378865
EDU	755848	915	611130	611130	611130	875761	743446

Table 2. describes the highest attacking, the highest targeted country, and the different attack types on a typical day. We looked at dates 02/04/16 and 02/08/16. We can notice that the USA is the highest targeted country. However, attackers rank differently based on the day and time of the attacks. On the day of 02/04/16, the USA has the most outgoing attacks, as well as incoming attacks. China attacks other countries more frequently as seen on dates 02/04/16 through 02/08/16 respectively. Moreover, most of the attacks were generated using SSL, SSH, TELNET, SIP, and ntis-ROUTER

Table 2: The highest attacking targeted country and attack types on 02/04/16 and 02/08/16[44].

Date	Time	Highest Attack Origin	Highest Attacks	Highest Attack Type	Highest Types	Highest Attack Target	Highest Target
2/4/2016	5:20 PM	US	482	ssl	752	US	1,126
2/4/2016	6:09 PM	Singapore	620	ssh	630	US	1,838
2/4/2016	6:14 PM	Singapore	638	ssh	638	US	2,351
2/4/2016	6:20 PM	China	433	ssl	212	US	1,084
2/4/2016	6:55 PM	US	20	ssl	17	US	24
2/4/2016	5:21 PM	US	471	ssl	732	US	1,231
2/4/2016	5:24 PM	US	532	ssl	735	US	1,330
2/4/2016	5:24 PM	US	532	ssl	735	US	1,341
2/4/2016	5:33 PM	US	412	telnet	531	US	1,617
2/4/2016	5:40 PM	US	538	ssl	264	US	1,387
2/4/2016	5:52 PM	US	104	unknown	68	US	240
2/4/2016	5:59 PM	US	421	telnet	183	US	1,247
2/4/2016	6:05 PM	US	481	unknown	270	US	1,340
2/8/2016	7:34 PM	China	402	netis-router	285	US	1,065
2/8/2016	8:11 PM	China	317	netis-router	136	US	718
2/8/2016	8:14 PM	US	665	telnet	494	US	1,564
2/8/2016	8:17 PM	Turkey	687	netis-router	917	US	2,252
2/8/2016	8:18 PM	Turkey	738	netis-router	991	US	2,693
2/8/2016	8:21 PM	Turkey	745	netis-router	1010	US	2,022
2/8/2016	8:26 PM	China	396	sip	167	US	883
2/8/2016	8:28 PM	China	438	sip	262	US	1,319
2/8/2016	8:30 PM	China	649	ssh	368	US	1,453
2/8/2016	8:09 PM	US	369	netis-router	420	US	1,116

12. Additional solutions

DoS / DDoS and Data breaches are major cyber threats that are causing maximum loss to organizations. Specifically, to address the above cyber-attacks, we have earlier proposed the deployment of split-protocol and split-encoding techniques on top of the important security protocols and encryption methods [49, 55, and 56].

In addition to data compression, improving privacy and security is an inherent benefit of the proposed method. It is possible to encode data recursively up to N times and to use a unique combination of b, c and N values to generate encryption keys. For example, a Split-encoding mechanism, in addition to Transport Layer Security (TLS) would en

-hance the overall security of the data being transferred between web browsers and servers (HTTPS). Using a unique b, c, and N combination within an SSH tunnel would add more security to the data on the wire. Servers may be configurable in a similar manner as in existing security protocols [55]. On the other hand, the proposed encoding opens doors to overcome the scalability issue of network security monitoring as it reduces the amount of data on the wire immensely. Scalability is considered to be one of significant security challenges at this present time [45]. With the Split-protocol, one can transmit data using a single or a combination of encoding techniques for faster and more secure communication.

13. Conclusion

In this paper, we analyzed Twitter data, Cyber threat maps, and Google Security Research to discover the classifications of attacks that are occurring and how the fight to stop them is happening. Classification of the vulnerabilities, types of attacks, and the vectors were done based on information collected from various databases. We found that there was a broad range of attack types that are commonly performed and that the largest target of Cyber Attacks was and still is, the US. Most of the vulnerabilities resolved by Google in its Security Research took an average of 2 months to patch the vulnerabilities, and most bugs were repaired within 3-4 months. The use of our mining information can help optimize the vulnerability fixing process and provide tools to detect and fix future vulnerabilities before they can be exploited. We presented a detailed study of past and ongoing cyber-attacks to help organizations better prepare for future threats. The unconventional techniques which have never been exploited such as BMC and Split-protocol techniques offer effective ways to defeat and prevent existing cyber threats.

For example, a Split-encoding mechanism in addition to Transport Layer Security (TLS) would enhance the overall security of the data being transferred between web browsers and servers (HTTPS). Using a unique b, c, and N combination within an SSH tunnel would add more security to the data on the wire [55].

As a follow-up to this study, an analysis of the various types of bugs based on the degree of complexity and their relative impact on software systems would be conducted. Considering the effect of bugs in software systems, it is imperative that software developers incorporate secure coding practices in the SDLC and not be reactive regarding potential security flaws.

References

[1] O. H. Alhazmi and Y. K. Malaiya. Quantitative vulnerability assessment of systems software. In Proceedings of the IEEE Reliability and Maintainability Symposium (RAMS'05), pages 615–620, Alexandria, VA, USA, 2005

[2] M. R. Lyu. Introduction. In M. R. Lyu, editor, Handbook of Software Reliability Engineering, chapter 1, pages 3–22. McGraw-Hill, 1996.

[3] Ozment, Andy. "Improving vulnerability discovery models." Proceedings of the 2007 ACM workshop on Quality of protection. ACM, 2007.

[4] Computer Science and Telecommunications Board. Computers at Risk: Safe Computing In the Information Age. National Academy Press, Washington, DC, 2001.

[5] https://www.fireeye.com/cyber-map/threat-map.html accessed on 28 Nov 2015

[6] Taylor, Robert W., Eric J. Fritsch, and John Liederbach. Digital crime and digital terrorism. Prentice Hall Press, 2014.

[7] http://www.lyncmigration.com/news/215/10/28/8268137.html accessed on 28 Nov 2015

[8] Weber S, Karger PA and Partaker A (2005) A software flaw taxonomy: Aiming tools at security. Software Engineering for Secure Systems (SESS'05).

[9] Prakash, Atul, and Rudrapatna Shyamasundar, eds. Information Systems Security: 10th International Conference, ICISS 2014, Hyderabad, India, December 16-20, 2014. Proceedings. Vol. 8880. Springer, 2014.

[10] Howard, M, LeBlanc, D and Viega, J (2005) 19 Deadly Sins of Software Security. Emeryville, C A: McGraw-Hill/Osborne.

[11] Weber S, Karger PA, and Partaker A (2005) software flaw taxonomy: Aiming tools at security. Software Engineering for Secure Systems (SESS'05).

[12] Dowd, M, McDonald, J and Schuh, J (2006) the Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities. Boston, MA: Addison Wesley Professional

[13] Landwehr CE, Bull AR, McDermott JP and Choi WS (1994) A taxonomy of computer program security flaws. ACM Computer Surveys 26(3):211–254.

[14] ISO 7498:1984 Open Systems Interconnection - Basic Reference Model.

[15] Du, W, Mathur, AP (1998) Categorization of Software Errors that led to Security Breaches, In Proceeding of the 21st National Information Systems Security Conference (NISSC'98), Crystal City, VA.

[16] Meunier, Pascal. "Classes of vulnerabilities and attacks." Wiley Handbook of Science and Technology for Homeland Security (2008).

[17] S. Wagner, "Defect classification and defect types revisited," in Proceedings of the 2008 workshop on Defects in large software systems. ACM, 2008, pp. 39–40.

[18] R. Chillarege, I. S. Bhandari, J. K. Chaar, M. J. Halliday, D. S. Moebus, B. K. Ray, and M.-Y. Wong, "Orthogonal defect classification-a concept for in-process measurements," Software Engineering, IEEE Transactions on, vol. 18, no. 11, pp. 943–956, 1992.

[19] R. Chillarege and K. A. Bassin, "Software Triggers as a Function of Time - ODC on Field Faults," in Dependable Computing and Fault-Tolerant Systems, vol. 10. IEEE Computer Society, 1995.

[20] R. Chillarege and K. Ram Prasad, "Test and development process retrospective-a case study using ODC triggers," in Dependable Systems and Networks (DSN), 2002. Proceedings. International Conference on. IEEE, 2002, pp. 669–678.

[21] M. Butcher, H. Munro, and T. Kratschmer, "Improving software testing via ODC: Three case studies," IBM Systems Journal, vol. 41, no. 1, pp. 31–44, 2002.

[22] A. Dubey, "Towards adopting ODC in automation application development projects," in Proceedings of the 5th India Software Engineering Conference. ACM, 2012, pp. 153–156.

[23] http://www.experian.com/assets/data- breach/brochures/2014-ponemon-2nd-annual-preparedness.pdf

[24] https://ocrportal.hhs.gov/ocr/breach/breach_report.jsf

[25] http://www.symantec.com/security_response/publications/threatreport.jsp

[26] http://www.statisticbrain.com/Twitter-statistics/accessed January 02, 2016.

[27] G. W. Bush. A national strategy to secure cyberspace, the office of the president. 2003.

[28] President's Information Technology Advisory Committee, Cyber Security: A crisis of prioritization, 2005.

[29] The National Strategy for Homeland Security, http://www.dhs.gov/interweb/assetlibrary/nat-strat-hls.pdf, 2002.

[30] Madan, Bharat B., and et al. "A method for modeling and quantifying the security attributes of intrusion tolerant systems." Performance Evaluation 56.1 (2004): 167-186.

[31] Papp, Dorottya, Zhendong Ma, and Levente Buttyan. "Embedded systems security: Threats, vulnerabilities, and attack taxonomy." In Privacy, Security and Trust (PST), 2015 13th Annual Conference on, pp. 145-152. IEEE, 2015.)

[32] Wang, Wenye, and Zhuo Lu. "Cyber security in the Smart Grid: Survey and challenges." Computer Networks 57, no. 5 (2013): 1344-1371

[33] http://www.privacyrights.org/data-breach accessed on 01/03/2016

[34] Issues - google-security-research - Google Security Research https://code.google.com/p/google-security research/issues/list?Can=1&num=100 (accessed March 10, 2016).

[35] Twitter Statistics – Statistic Brain, http://www.statisticbrain.com/twitter-statistics/ (accessed March 10, 2016). Twitter Statistics – Statistic Brain, http://www.statisticbrain.com/twitter-statistics/accessed March 10, 2016.

[36] Yang, Chao, Robert Harkreader, Jialong Zhang, Seungwon Shin, and Guofei Gu. "Analyzing spammers' social networks for fun and profit: a case study of the cybercriminal ecosystem on twitter." In Proceedings of the 21st international conference on World Wide Web, pp. 71-80. ACM, 2012.

[37] Twitter vulnerability allows cyber criminals to spread spam. http://www.one.com/en/web-hosting-news/

[38] Website/twitter-vulnerability-allows-/ twitter-based Botnet Command Channel.
http://asert.arbornetworks.com/2009/08/twitter-based-botnet-command-channel/

[39] Twitter accounts spreading malicious code. http: //www.net- security.org/malware_news.php? id=1554

[40] KOOBFACE: Inside a Crimeware Network. http: //www.infowar-monitor.net/reports/iwm-koobface.pdf

[41] G. Stringhini, S. Barbara, C. Kruegel, and G. Vigna. Detecting Spammers on Social Networks. In Annual Computer Security Applications Conference, 2010.

[42] C. Yang, R. Harkreader, and G. Gu. Die Free or Live Hard? Empirical Evaluation and New Design for Fighting Evolving Twitter Spammers. In Proceedings of the 14th International Symposium on Recent Advances in Intrusion Detection (RAID'11), 2011

[43] https://www.fireeye.com/cyber-map/threat-map.html accessed on 28 Nov 2015
[44] http://map.norsecorp.com accessed on Jan 05, 2015.
[45] Kalutarage, Harsha K., Siraj A. Shaikh, Indika P. Wickramasinghe, Qin Zhou, and Anne E. James. "Detecting stealthy attacks: Efficient monitoring of suspicious activities on computer networks." Computers & Electrical Engineering 47 (2015): 327-344.
[46] Singh, Shailendra, and Sanjay Silakari. "A survey of cyber-attack detection systems." International Journal of Computer Science and Network Security 9, no. 5 (2009): 1-10.
[47] Bharat S. Rawal, Songjie Liang, Alae Loukili, Qiang Duan. (2016). Anticipatory Cyber Security Research: An Ultimate Technique for the First-Move Advantage.TEM Journal, 5(1), 3-14.
[48] Ralston, Patricia AS, James H. Graham, and Jefferey L. Hieb. "Cyber security risk assessment for SCADA and DCS networks." ISA transactions 46.4 (2007): 583-594.
[49] Grottke, Michael, Allen P. Nikora, and Kishor S. Trivedi. "An empirical investigation of fault types in space mission system software." In Dependable Systems and Networks (DSN), 2010 IEEE/IFIP International Conference on, pp. 447-456. IEEE, 2010.
[50] M. Grottke and K. S. Trivedi, Software faults, software aging and software rejuvenation, Journal of the Reliability Engineering Association of Japan 27(7):425–438, 2005.
[51] M. Grottke and K. S. Trivedi, A classification of software faults, in Supplemental Proc. Sixteenth International Symposium on Software Reliability Engineering, 2005, pp. 4.19-4.20.
[52] M. Grottke and K. S. Trivedi, fighting bugs: Remove, retry, replicate, and rejuvenate, IEEE Computer 40(2): 107–109, 2007.
[53] J. Gray, Why do computers stop and what can be done a``bout it? in Proc. Fifth Symposium on Reliability in Distributed Systems,1986, pp. 3–12.
[54] A. Avižienis, J.-C. Laprie, B. Randell, and C. Landwehr. Basic concepts and taxonomy of dependable and secure computing, IEEE Transactions on Dependable and Secure Computing 1(1):11–33, 2004.
[55] Bharat S. Rawal, Songjie Liang, Shiva Gautam, Harsha K. Kalutarage, and Pandi Vijayakum. “Nth Order Binary Encoding with Split-protocol” Not published.
[56] Bharat S. Rawal, Ramesh K. Karne, and Alexander L. Wijesinha. "Splitting HTTP requests on two servers." In Communication Systems and Networks (COMSNETS), 2011 Third International Conference on, pp. 1-8. IEEE, 2011.