Call of Duty: Can Turkey Benefit from Crowd-Sourced Serious Games to Strengthen Its Cyber Security Capabilities?

Ümit Tellioğlu, Bilal Kartal, Mehmet Şimşek, Ömer Eryılmaz

***Abstract*—The software verification process is cumbersome and currently requires highly trained experts and considerable time. This process is critical to eliminate bugs and to strengthen software systems. Military software systems are of utmost importance for national security. One way to improve this verification phase is to use a crowdsourcing paradigm to strengthen software systems. Turkey has an increasing number of the Internet users and online game players who spend more time per game session than the average user. In this paper, we propose using crowd-sourced serious games (CSSGs) to improve Turkey's national cyber security capabilities and examine the stakeholders of Turkey needed for a CSSG project.**

***Index Terms*—Crowd-Sourced Serious Games, Cyber Security, Software Formal Verification**

I. Introduction

Playing is an essential part of the human experience. It results in learning the required activities of life at an early age, and it is enjoyable. Humans have used games throughout history as an instrument of play activity. With the improvement of technology, the game-playing process has digitalized and become more immersive. Humans have created digital machines that run virtual software to interact, play, and enjoy. After the Internet boom, people had the opportunity to connect with others throughout the world. As a result, people have spent billions of hours dedicated to online computer games. Trends show that the time spent playing online games will continue to increase. Consequently, L. Von Ahn [1] developed the idea of *games with a purpose,* claiming that it may be beneficial for humanity if billions of game playing hours can be harvested to solve problems.

The aim of crowd-sourced serious games (CSSGs) is to harness non-expert citizens to solve scientific problems for which scientists do not have adequate resources as we will present many existing software systems proving the insufficiency of experts. Considering the aim of generating a side product from game playing, the term ”citizen science” is introduced that aims to direct non-expert people into real-life scientific problem solving [2]. At that point, however, the issues arise as to how we benefit from the crowd's power for a subject that is completely unfamiliar to them. The solution is to place a converter from the game to an algorithm between non-expert citizens and science, which will attract regular people, transforming them into citizen scientists [2]. There have been applications of CSSGs with different purposes such as *EyeWire*, *Phylo* and *Foldit*. For instance, non-expert *Foldit* players helped to solve the structure of an AIDS related enzyme within three weeks [3]. Scientists had sought to find a solution to this problem for ten years. In addition, *EyeWire* players have contributed to a complex neuroscience research on a problem remaining unsolved for 50 years [4].

As a developing country, Turkey has produced national technological systems that are highly software dependent, and CSSGs can be very beneficial as a tool to reduce software-caused vulnerabilities. The growth of technology usage is fast all over the world, and the situation is similar for Turkey as well. Technology usage and the ratio of people connecting to the Internet have been increasing in the country. With this progress, technological systems such as e-government and National Judiciary Informatics Systems (UYAP in Turkish) have been developed and are now being actively utilized. Moreover, Turkey has begun producing critical national military vehicles such as T129 ATAK Helicopter, Altay Tank, ANKA UAV and national warship (Milgem Project). All of these critical systems are potential targets for cyber-attacks. One of the important processes for vulnerability prevention of these critical systems is performed through software verification which is a possible application area for crowdsourcing.

Under these circumstances, a comprehensive strategy to strengthen the national-cyber-security by using national resources is imperative for Turkey. As an initial step in this direction, we will hypothesize two relational questions within this paper:

1) Is it possible to use CSSGs to improve cyber security?

2) Does Turkey have enough resources or stakeholders to harness CSSGs aimed at cyber security?

Organization The rest of this paper is organized as follows. Section 2 highlights the related work on this research topic and Section 3 presents existing software systems with discussion of military software usage. In Section 4, we show Turkey's stakeholders of a CSSG project. Section 5 concludes the paper, and we present future research directions in the final section.

Ümit Tellioğlu, Turkish Military Academy, The Defense Sciences Institute, Bakanlıklar, Ankara-Turkey, email: umittelli@gmail.com

Bilal Kartal, Turkish Military Academy, Computer Engineering Department, Bakanlıklar, Ankara-Turkey, email: bllkrtl.cs@gmail.com

Mehmet Şimşek, War Colleges Command, Army War College, Yenilevent, Istanbul-Turkey, email: msimsekfg@gmail.com

Ömer Eryılmaz, War Colleges Command, Army War College, Yenilevent, Istanbul-Turkey, email: omereryilmaz83@gmail.com

II. Related work

In this section, we will present some discussion of crowdsourcing and software verification. Additionally we will show how crowdsourcing can advance cyber security through different applications.

A. Crowdsourcing

Crowdsourcing has been a game changer for solving difficult problems with a mass of ordinary, untrained people. Howe used "crowdsourcing" as a term with different application perspectives [5]. The range of application possibilities is huge, extending even to capabilities of changing how the Web works by pushing content creation and management activities to its ordinary users in Web 2.0 [5].

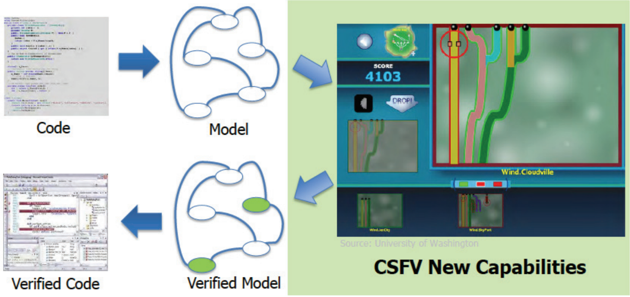


Fig. 1. *An example of a converter model from game input to software verification process using a CSSG.* [6]

The crowdsourcing approach employs a large number of agents, i.e., untrained people, with different reliability to accomplish a specific task. Therefore, statistical reliability requires a large number of agents in the system so that the impact of the outliers will be reduced. For most of the CSSGs, there will be a middle transformer layer that takes untrained ordinary people's input and converts it to a solution, e.g., see *Foldit*, [3]. Yet, there are many other application domains of crowdsourcing. For example, it can be used for emergency response after disasters by creating a collective system [7]. However, there are also harmful applications of this technique when applied in a different fashion. For example, a study [8] analyzing the contents on the web portals in China revealed the existence of an application called *Human Flesh Search*. In this application [8], participating web users perform a collaborative cyber surveillance to find other people, or to obtain specific information about them through the portal. It is reported that this search generally results in harmful effects on the community.

One other application of crowdsourcing is specifically for business purposes. Companies can take advantage of the diverse perspective of the crowd to collect new ideas for their business strategies [9].

Finally, Mechanical Turk is a very widely used tool that is a great proof of concept example for crowdsourcing. In this case, individuals and businesses are able to push tasks to a task pool of compensated crowd workers. This way, businesses have their tasks completed and users are paid for the assistance they provide.

B. Software Verification

The software verification aims to guarantee correctness of the program as a disciplined phase within software engineering. This process ensures that the software performs only the tasks for which it is designed with no unintended tasks, which might be malicious [10]. In terms of cyber defense, the latter case, i.e. making sure that the software does not perform any unintended tasks is crucial. Non-developers should test software during verification phase [10], i.e. the exact step that we propose to be pushed to crowd. This improves the verification.

To have an overview of widely used software, let's examine an operating system. Linux is one of the best-known and most popular open source operating systems and it has many derivatives with different distributions. According to the report shown in [11], five-computer science researchers examined 5.7 Million lines of *Linux* source code in four years. They concluded that the *Linux 6* kernel code was better and more secure than that of most propriety software [11]. Throughout the study, the researchers worked on *2.6 Linux* production kernel, which was used by Red Hat, Novell, and other popular vendors, and found 985 bugs in 5.7 million lines of code. On the other side, according to Carnegie Mellon University's CyLab Sustainable Computing Consortium, commercial closed source software have 20 to 30 bugs for every 1,000 lines, resulting in 114,000 to 171,000 possible bugs in the 40 million lines of code in *Windows XP* [11].

Software bugs can cause serious problems. First, systems using the software may stop working or fail to achieve its intended results, which historically has caused space shuttle crashes, millions dollars of financial loss for companies, fatal mistreatment of patients, power outages in cities and other serious incidences [12] [13]. If military systems fail, the results can be more tragic. For example, during the First Gulf War, the Patriot air defense system failed to prevent an incoming missile because of a bug in the software and caused the death of 28 soldiers [14]. Another example of a software bug resulted in death of 29 people in a Chinook helicopter crash [15].

Second, adversaries can exploit vulnerabilities caused by bugs. Hackers mostly use software bugs and *zero* day bugs to exploit systems. One of the most dangerous incidences from a global perspective is the "heart bleed bug" [16]. In this case, hackers were able to reach encrypted data by using this bug, which had impacted the *OpenSSL* cryptographic software library, i.e. the main security provider for quite a while. Hacking of military systems and vehicles is also possible. In particular, hackers have targeted unmanned vehicles (UVs) as they hacked a UAV in Afghanistan [17]. One crucial step to detect these vulnerabilities in software is through formal verification.

Considering the increase in use of technology in daily life and number of bugs in these technology systems, the crucial need to improve verification process to make these systems safer is very clear.

C. *Cyber Security through Crowdsourcing*

As e-services are increasingly used for government applications, the need to protect the services also increases as each bug is a possible chance for exploitation by malicious users. Apart from conventional testing processes to strengthen software, each country can use crowd-sourced applications in order to improve security of e-service software. For example, Paulhamus et al. [18] proposed a proof of concept game called *Flux Hunter* which is played in a network and utilizes crowdsourcing to identify malicious web-sites for cyber security. The game aims to harness the employees' play experience to identify cyber threats. As we described earlier, in this type of application, there is a transformer layer that takes game playing input and feeds it into the problem-solving algorithm. From this point, *Flux Hunter* presents pictures to players that in fact represent some camouflaged network information, and the players must choose images based on their perception of threat. Also, the players are trained for the game with supervised images and sample answers beforehand.

Another application that uses crowdsourcing along with machine learning techniques is presented in [19], which allows personal web users to identify scam web sites to prevent any phishing type of attack. Through a custom module, users can post and share information about the websites and rate them, and ultimately logistic regression is applied to classify the websites. There are several other applications where crowdsourcing can enhance the security of software to defend against cyber threats.

III. Background on existing software systems

Software sold on the market can contain up to 5 bugs per thousand line of code as reported in [20]. To reduce the number of bugs in software, verification needs to be performed in an improved and faster fashion. However, verification is a complex process that can be performed by a limited number of highly trained experts, leading to insufficient resources to verify a large number of software [21]. Also, while the line of code produced in the world has been increasing very rapidly, the number of experts for verification phase has not followed the same trend [21]. One study [22] suggests interesting code production figures, i.e. there are 6 million software developers in the world and they produce 300 millions lines of code weekly or 15 billion lines of code yearly. Thus, if all verification experts in the U.S. worked only on *Windows 8*'s source code to formally verify and find even 25 predefined vulnerabilities, it would take them more than 30 years [21]. This demonstrates just how time consuming the process is and insufficient supply to solve the problem in a traditional way. Additionally, there are several types of software, including operating systems, commercial off-the-shelf (COTS), and others, which further complicates the issue. When we look at the Table I, we can see that the size of the software needed to be verified is huge.

Verification is also critically important for military systems. Table I shows that military systems such as jet fighters, missiles, and future combat systems have software systems made of millions of lines of source code. Table II also supports this by demonstrating the increased reliance on software of jet fighters over the years. It is obvious from the trend in Tables I and II that this reliance will keep increasing due to the shift to both unmanned air vehicles and usage of artificial intelligence supported systems in battlefield. Thus, as the produced code continues to increase in volume, the need for alternative ways to handle verification will be more urgent.

TABLE I
LOCs of a group of software [23], [6]

Name	Lines of Code (LOC)
Windows XP	40 M
Windows 7	40 M
Linux 3.1	15 M
Mac OS X 10.4	86 M
Debian 5.0 (all software in package)	324 M
Android OS	12 M
Microsoft Office (2013)	45 M
F-22 Raptor Jet Fighter	1.7 M
F-35 Fighter	24 M
Patriot PAC-3 Missiles	Close to 2 M
US Army's Future Combat System	63.8 M
Hubble Space Telescope	2 M
Google Chrome (2011)	5.4 M
Boeing 787 Dreamliner	6.1 M
FireFox	9.7 M
Chevrolet Volt (Electric Car)	10 M
Apache Open Office	23 M
MySQL	12.5 M
Software in typical new car, 2013	100 M
Healthcare.gov	500 M

TABLE II
Software dependence of military airplanes by years [24]

Military Aircrafts	Year	Percentage of Functions Performed in Software
F-4	1960	8
A-7	1964	10
F-111	1970	20
F-15	1975	35
F-16	1982	45
B-2	1990	65
F-22	2000	80

Defense Advanced Research Projects Agency (DARPA) has been conducting the Crowd Sourced Formal Verification (CSFV) project since 2011, and developed five CSSGs to initially verify some small open source software [20]. The overall aim of the project is to find a way to formally verify software used in complex military systems without heavily relying on the insufficient amount of software verification experts. The games are hosted in a single portal called Verigames [20]. A software company named Topcoder, which develops software by using the crowdsourcing concept is forming this portal.

IV. Stakeholders of a CSSG Project in Turkey

As we stated in our hypothesis questions, harnessing crowdsourcing models embedded in computer games in order to solve different problems is a new research trend similar to many applications, e.g., *EyeWire*, *Phylo*, *Foldit*. Moreover, a recent study [25] shows that Turkish game players consume more time for online games than world average (see Table III). This supports our hypothesis in the sense that Turkey can gain a great deal of benefits from CSSGs.

Three stakeholders are required to successfully create a project that formally verifies software by using CSSGs. They are game players, developers, and software verification experts. In this section, we will examine Turkey's current status related to these three stakeholders.

A. Game Players

The role of the gamers is crucial for CSSGs because without gamers, crowdsourcing projects cannot realize their goals. In addition, each player has different motivation to play games [27]. Recently, it is shown in [27] that CSSGs often have difficulty finding enough players, since the nature of the game is linked to specific problem solving rather than the players' personal pleasure. Therefore, in order to incentivize many Turkish players, there is need to find a common motive such as patriotism to support Turkish cyber defense capability. In the current study, we will examine Turkish game players. However, future research should define the possible motivational factors among Turkish players. Also, if the games are enjoyable, they will attract players from all over the world.

Figure 3 shows the data of Turkish game players based on the game platforms. CSSGs have been developed for various platforms. For example, *Foldit* is downloadable for PCs, *Xylem* is mobile (for iPads), and *EyeWire* and *Stormbound* are browser based online games. Also, one important factor here is to recognize the leading medium for game playing, e.g. social networks, high connectivity between the users provided by social networks should be utilized to increase preference for CSSGs . On the other hand, the ratio ranges from 12.8% to 22.3% of the overall population across the different platforms for Turkish players and Table III clearly shows that Turkish players tend to spend on average more time playing social, casual, and mobile games than any other countries' players in the world.

TABLE III
Players data of a list of countries and the world average [25]

	Session Time Avg.	Avg. Session Count	Avg. Total Play Time
Turkey	45.9 mins	15.4	705 mins
Japan	30.4 mins	22.5	683 mins
South Korea	28.9 mins	17	492 mins
U.S.	21.9 mins	6.5	142.8 mins
Russia	23.2 mins	3.3	76.2 mins
China	2.7 mins	1.3	16.8 mins
World Avg.	26.7 mins	5.9	157.6 mins

In addition, CSSG projects can benefit from the increasing popularity of mobile devices in Turkey. Mobile games' complexity and quality get closer to PC or console games with the improved capabilities of the mobile devices (smart phones and tablet computers). Figure 2 briefly shows the popularity of mobile devices in Turkey. A report in 2013 showed that 19% of all phones sold in the country are smart phones [28]. In addition, 38% of smart phone owners use applications in their smart phones, and 32% of those phone owners play mobile games via those devices [28]. Moreover, tablet computers connected to the Internet have exceeded 1.5 million.

The Ministry of National Education (MONE) plans to gift tablet computers to all students and teachers below university level and has already distributed more than 700,000 of them [29]. Therefore, the number of tablet computers in Turkey is increasing. Although MONE currently restricted usage of game type applications from given tablets [29], if developers can produce education purposed games such as *Xylem*, this would be beneficial for both the students and the CSSG project. Of course, this is a future project that educators and pedagogues would need to discuss.

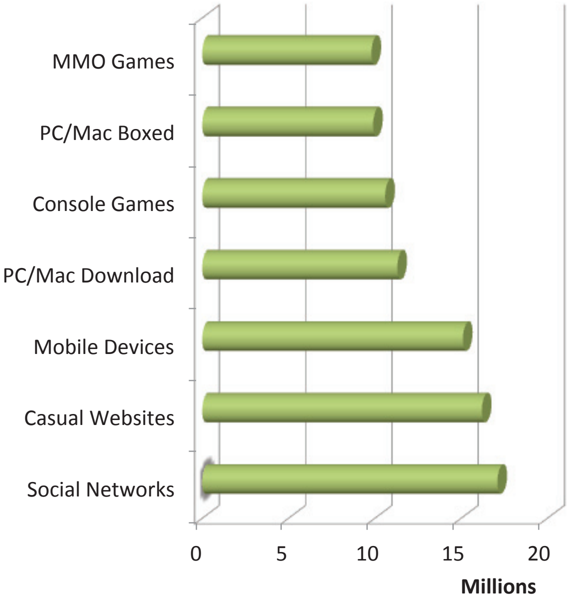
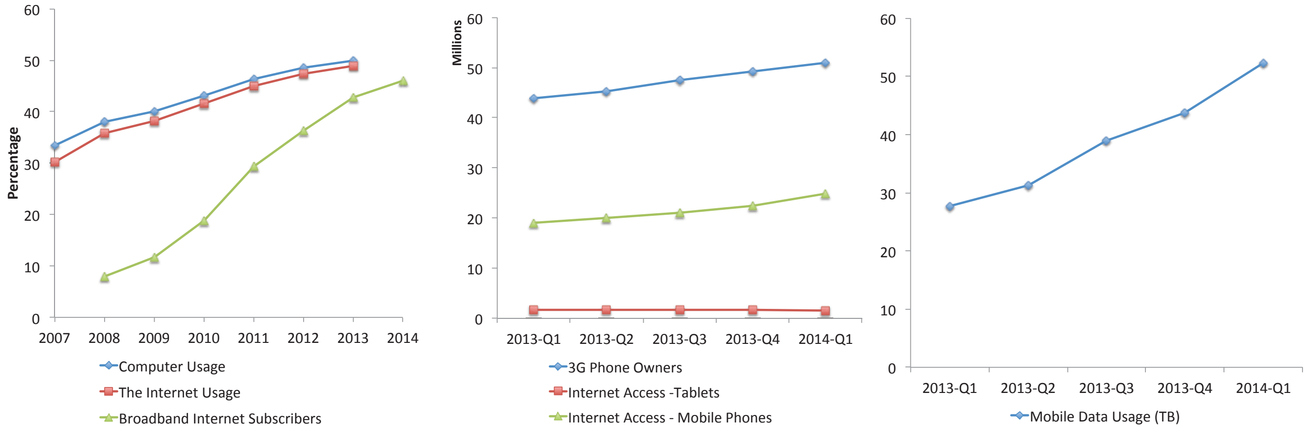


Fig. 3. Gamers in Turkey by game type [30]

B. Game Developers

The second stakeholders are game developers, so Turkey needs experts in game creation to achieve the ultimate purpose. Upon searching the literature, we saw that a Turkish gaming company, Peak Games, has developed social, mobile, and online browser games. It is a leading company not only in the Middle East, but globally. Table IV compares the number of players for Peak Games with the world's most popular game developers [31]. In addition, mostly young entrepreneurs in Turkey are building start-up firms related to gaming each year. For example, in *e-tohum* database, which is a portal that



(a) (b) (c)

Fig. 2. *(a) Computer usage and Internet access have steadily increased in Turkey in the last decade. Currently half of the population accesses it. (b) Similar to the trend worldwide, smart phone usage has also increased and tablet usage has exceeded 1.5 million, which justifies one of the reasons why games should be utilized for more important problems in addition to the joy of playing. (c) Mobile data usage also increases as it gets cheaper and more accessible by the public.* [26]

gathers Turkish entrepreneurs, almost 20 electronic gaming start-ups have been recorded since 2010 [32]. Lastly, METU Animation Technologies and Game Development Center supports game developers, and young developers have created more than 200 games in Turkey since 2008 [33]. These recent events demonstrate that Turkey has also a young population who can contribute to the developing process.

TABLE IV
DAILY AND MONTHLY ACTIVE USERS (DAU, MAU) OF SOME LEADING MOBILE AND SOCIAL ONLINE GAME DEVELOPERS [27]

	DAU	MAU
Peak Games	11.8M	25M
Zyanga	11.1M	292M
Storm8	4M	-
Glu Mobile	3.4M	29M
DeNa	-	16.9M
GREE	-	13.9M

C. Verification Experts

The third stakeholders are verification experts. Turkey currently has private sector firms working in software verification such as BEAM Technology and Altay Software. Universities can also support the private sector. For example, Koc University Multi-Core Software Engineering research center and METU Software Management Research Group both work on software formal verification. In addition, Turkey has individual academic experts on software verification.

V. CONCLUSION

Software bugs can cause huge risks for governmental, military (ATAK, MILGEM, ALTAY, etc.), and civilian systems (UYAP etc.) [20]. ATAK helicopters, Altay Tanks, Milgem, ANKA UAVs have been nationally developed. Each of these systems uses technology and inevitably run many lines of software. Moreover, Turkey as a developing country is trying to build national military systems. According to the Turkish Department of Defense Strategy document, the country has also planned to increase the nationality of the current weapons systems and to produce high technology (meaning software dependent) systems such as air defense systems, unmanned sea-air-land vehicles, autonomous systems, satellites and cyber defense systems in medium and long-term [34]. All those indicate that software formal verification is critically important for Turkey.

In order to achieve these goals, National Cyberspace Security Policy, another cyber security document prepared in 2008 with the collaboration of government agencies, underlines the development of national cyber capabilities. It further stresses the importance of the protection of critical infrastructures and the cyber security of the public institutions.

In this research, we demonstrated that CSSGs are used to strengthen cyber security. One of them aims to formally verify software for which experts need source code. Obviously, source code is invaluable for critical systems. Therefore, to formally verify national software using domestic resources is invaluable, and CSSGs have the potential to support verification experts to achieve that goal. In addition, we examined Turkey's domestic assets that can provide resources for a crowd-sourced formal verification project. We propose that Turkey has capacity to conduct this project.

VI. FUTURE WORK

Crowd-sourced game projects have been conducted for less than a decade, and most of them have begun within the last two years. Consequently, we are unable to find a project that has completely achieved its goals, but some of them, such as *EyeWire* and *Foldit*, are progressing well. How to run a crowd-sourced game project, and how to incentivize the stakeholders for the project is subjects of future work.

Acknowledgment: We thank Yunus Küçük for his valuable feedbacks.

REFERENCES

[1] L. Von Ahn, "Games with a purpose," *Computer*, vol. 39, no. 6, pp. 92–94, 2006.

[2] A. Wiggins and K. Crowston, "From conservation to crowdsourcing: A typology of citizen science," in *System Sciences (HICSS), 2011 44th Hawaii International Conference on*. IEEE, 2011, pp. 1–10.

[3] M. J. Coren and F. Company, "Foldit gamers solve riddle of hiv enzyme within 3 weeks." [Online]. Available: http://www.scientificamerican.com/article/foldit-gamers-solve-riddle/

[4] J. S. Kim, M. J. Greene, A. Zlateski, K. Lee, M. Richardson, S. C. Turaga, M. Purcaro, M. Balkam, A. Robinson, B. F. Behabadi *et al.*, "Space-time wiring specificity supports direction selectivity in the retina," *Nature*, vol. 509, no. 7500, pp. 331–336, 2014.

[5] J. Howe, "The rise of crowdsourcing," *Wired magazine*, vol. 14, no. 6, pp. 1–4, 2006.

[6] D. Dean, "Proceed and crowd-sourced formal verification," Nov. 2011. [Online]. Available: http://www.dtic.mil/dtic/tr/fulltext/u2/a551949.pdf

[7] M. F. Goodchild and J. A. Glennon, "Crowdsourcing geographic information for disaster response: a research frontier," *International Journal of Digital Earth*, vol. 3, no. 3, pp. 231–241, 2010.

[8] X. Pan, "Hunt by the crowd: an exploratory qualitative analysis on cyber surveillance in china," *Global Media Journal*, vol. 9, no. 16, pp. 1–19, 2010.

[9] B. L. Bayus, "Crowdsourcing new product ideas over time: An analysis of the dell ideastorm community," *Management Science*, vol. 59, no. 1, pp. 226–244, 2013.

[10] D. R. Wallace and R. U. Fujii, "Software verification and validation: an overview," *IEEE Software*, vol. 6, no. 3, pp. 10–17, 1989.

[11] M. Delio, "Linux: Fewer bugs than rivals," 2004. [Online]. Available: http://archive.wired.com/software/coolapps/news/2004/12/66022

[12] M. Lake, "Epic failures: 11 infamous software bugs," Sep. 2010. [Online]. Available: http://www.computerworld.com/s/article/9183580/Epic_failures_11_infamous_software_bugs?taxonomyId=18&pageNumber=2

[13] K. Poulsen, "Software bug contributed to blackout." [Online]. Available: http://www.securityfocus.com/news/8016

[14] A. Lum, "Verification software bug report patriot missile software problem." [Online]. Available: http://sydney.edu.au/engineering/it/~alum/patriot_bug.html

[15] S. Rogerson, "The chinook helicopter disaster," *IMIS Journal*, vol. 12, no. 2, 2002.

[16] "The heartbleed bug." [Online]. Available: http://heartbleed.com/

[17] D. Cenciotti, "German heron drone hacked and crashed by taliban in afghanistan," Nov. 2013. [Online]. Available: http://theaviationist.com/2013/11/13/heron-hacked-afghanistan/

[18] B. Paulhamus, A. Ebaugh, C. Boylls, N. Bos, S. Hider, and S. Giguere, "Crowdsourced cyber defense: lessons from a large-scale, game-based approach to threat identification on a live network," in *Social Computing, Behavioral-Cultural Modeling and Prediction*. Springer, 2012, pp. 35–42.

[19] E. Fink, M. Sharifi, and J. G. Carbonell, "Application of machine learning and crowdsourcing to detection of cybersecurity threats," in *Proceedings of the US Department of Homeland Security Science Conference–Fifth Annual University Network Summit, Washington, DC*, 2011.

[20] "Darpa." [Online]. Available: http://www.darpa.mil/Our_Work/I2O/Programs/Crowd_Sourced_Formal_Verification_(CSFV).aspx

[21] D. E. Williams, "Gamers vs. hackers." [Online]. Available: http://www.verigames.com/Geek-May_2014-LEVEL_UP.pdf

[22] J. Johnson, "The new industrial revolution (infographic)." [Online]. Available: http://insights.wired.com/profiles/blogs/the-new-industrial-revolution

[23] D. McCandless, "Information is beautiful." [Online]. Available: http://www.informationisbeautiful.net/visualizations/million-lines-of-code/

[24] "Report of the defense task froce on defense software." [Online]. Available: http://www.acq.osd.mil/dsb/reports/ADA385923.pdf

[25] "Playnomics quarterly player engagement study." [Online]. Available: http://www.playnomics.com/assets/Playnomics-Q1-2013-Engagement-Report.pdf

[26] "Turkey electronic communication sector quarterly report on market data," May 2014. [Online]. Available: http://www.tk.gov.tr/kutuphane_ve_veribankasi/pazar_verileri/ucaylik14_1.pdf

[27] U. Tellioglu, G. Xie, J. Rohrer, and P. Charles, "Whale of a crowd: Quantifying the effectiveness of crowd-sourced serious games," to be published on the proceedings of 19th International Conference on Computer Games. [Online]. Available: http://faculty.nps.edu/xie/papers/whale-effect-games14.pdf

[28] "The mobile consumer a global snapshot," Feb. 2013. [Online]. Available: http://www.nielsen.com/content/dam/corporate/uk/en/documents/Mobile-Consumer-Report-2013.pdf

[29] "Fatih project." [Online]. Available: http://fatihprojesi.meb.gov.tr/tr/icerikincele.php?id=6

[30] "Number of gamers in turkey in 2012, by platform." [Online]. Available: http://www.statista.com/statistics/234793/number-of-gamers-in-turkey-by-platform/

[31] D. Takahashi, "Peak games dominates mobile social games in the middle east and turkey," Oct. 2013. [Online]. Available: http://venturebeat.com/2013/10/15/peak-games-dominates-mobile-social-games-in-the-middle-east-and-turkey/

[32] "etohum web portal." [Online]. Available: http://www.etohum.com/girisimler

[33] "Metutech animation technologies and game development center." [Online]. Available: http://odtuteknokent.com.tr/portal/faces/atom-navigation/about?_adf.ctrl-state=ph4cxo75l_122&_afrLoop=3417505009042893

[34] "Turkish undersecretariat for defense industries technology management strategy 2011-2016." [Online]. Available: http://www/ssm.gov.tr/anasayfa/kurumsal/Documents/201116_TYY.pdf