The Emerging Technology that Attacks and Fixes Software

Matthew Molyett

UMUC CSEC 670

Feb 19, 2017

Introduction

Emerging technologies is sort of the primary mission of the Defense Advanced Research Projects Agency, or DARPA. In words of the agency itself, that mission is “to make pivotal investments in breakthrough technologies for national security”(DARPA, 2015a, pg 1). Recent major pushes to advance technologies have occurred in the form of so-called DARPA Grand Challenges. The Grand Challenge plan began with Self-Driving cars in 2004/2005, even though no car was able to finish. (Borgolte, 2016) A second Grand Challenge supported the growth of Robot research. Thirteen years later we are seeing significant progress toward commercial self-driving cars. Thus, the technologies being driven by DARPA Grand Challenges definitely qualify as emerging technology.

This past August I was an attendee at DEFCON 24 at Paris/Bally’s in Las Vegas. (DEF CON 24 Hacking Conference, 2016) This allowed me the opportunity to sit in attendance at the DARPA Cyber Grand Challenge Final Event. A gigantic ballroom was filled with an audience facing a stage with eight supercomputer systems, the finalists of a two year Grand Challenge seeking to bolster the development of automated systems that can replace human vulnerability analysts. Just like self-driving cars allows a computer

to replace a human driver with human reflexes and human errors, the systems running on those supercomputers could identify and fix exploitable weaknesses at computer speed with unwavering attention. Following the timeline from the self-driving car Challenge, 2032 could see commercial offerings which make human software analysis a thing of the past. Compilers could upload newly created binaries to a server for analysis and then download a report with vulnerabilities found and a vulnerability proof of concept.

Background

In June 2014 the Cyber Grand Challenge, CGC, was announced by DARPA as a driver to creating systems and technology leading to automated vulnerability analysis and exploit defense within software. (Song & Alves-Foss, 2015) The format was to be similar to a common test of human teams: the Capture the Flag game, or CTF. After the gauntlet was laid down by the agency, there was grant funding provided to seven teams through DARPA contracts. (Song & Alves-Foss, 2015) In addition to the seven funded-track teams, other teams could join in through self-funding.

One of the finalists, Shellphish, was a self-funded entrant known in the CTF community as a team that originated at UC Santa Barbara. (Borgolte, 2016) DARPA had recognized the need for “machine-speed, scalable cyber defense”(OUTREACH, 2016), according to the program manager behind the project launch in 2013. With severe exploits like the Heartbleed attack on TLS/SSL software revealed in 2014 and the ever-expanding number of internet connected devices, or Internet of Things, the importance of detecting vulnerabilities as fast as vendors and developers release them. Since severe bugs typically remaining unpatched for 10 months(OUTREACH, 2016), that is a lot of vulnerabilities that are undiscovered with more being released every month.

Two events were established to carry out this challenge: a CGC Qualifying Event on June 3, 2015 and the CGC Final Event, which was what I witnessed in August 2016. One set of rules was used for the Qualifying Event and another for the Final Event. For the Qualifying Event, vulnerabilities were submitted as Proof of Vulnerability and patches were submitted for the Challenge Binaries. Patches that degraded performance scored less well. For the Final Event, patches were applied to challenge

daemons that the system was defending and opponents threw live exploits at their patched binaries. (Song & Alves-Foss, 2015)

Technology

The goal was “fully automated software vulnerability analysis and repair” (Song & Alves-Foss, 2015, pg 1) and so that leaves two problems needing solved, namely detecting vulnerabilities and creating patches for them. For the Qualifying Event vulnerabilities just had to be proven, but the Final Event required them to be exploited. Thus, the technological problem starts with detecting vulnerabilities, but then diverges into automated exploit development and automated patch development.

In their paper titled Automatic Exploit Generation, the authors describe the history of tracking execution symbolically, a prerequisite to automatic exploit generation, or AEG. (Avgerinos et al, 2014) Furthermore, they describe the Mayhem system developed from 2012 for performing AEG. Testing Mayhem had revealed that AEG techniques reveal two types of bugs: sound and complete. The important difference between the two is that “a sound AEG technique says a bug is exploitable if it really is exploitable, while a complete technique reports all exploitable bugs”

(Avgerinos et al, 2014, pg 82) and that Mayhem is not complete, though it is sound. The second major insight from Mayhem was that it discoverers exploitable bugs, but not necessarily attack vectors. A naïve approach to the subject equates the two, but the authors describe that an attack vector must involve an exploit on an attack surface: untrusted input, network protocols, process interaction, or media players. (Avgerinos et al, 2014) This distinction allows the paper to discuss frankly an exploitable bug that was discovered through Mayhem but does not cause much damage due to the example not being on an attack surface. Mayhem was the AEG used in the system “Mayhem” that ForAllSecure built to win the Cyber Grand Challenge. (OUTREACH, 2016)

On the other side of the coin, both the Qualifying Event and the Final Event required creating patches for the discovered vulnerabilities. It was even weighted to be a very significant portion of the score for the Qualifying Event, so much so that failure to create at least one patch was an automatic failure for that Event. (Song & Alves-Foss, 2016) Secure and Dependable Systems, CSDS, from Idaho performed generic patching to reduce vulnerabilities from stack buffer overflows without requiring the detection of specific, present vulnerabilities. (Song & Alves-Foss, 2016) By

adding stack canary code they could render stack buffer overflows non-exploitable. This is a defense that modern compilers add automatically, by default, and so would not be an effort carried out against production compiled code, but it does demonstrate the potential for automated binary patching.

CSDS minimized the overhead that was introduced by their generic patching by discovering what they termed “safe code”. (Song & Alves-Foss, 2016, pg 3) Safe code would be functions that the return address could not be manipulated, and so would not need to be protected by a stack canary. Detecting such a situation within a compiled binary which has been stripped of debugging assistance is not trivial, and so CSDS created, from scratch, heuristics for detecting function bounds and variables in use as well as detecting the presence of stack based structures or arrays. (Song & Alves-Foss, 2016) Based on my professional history of manually analyzing disassembled malware functions I can appreciate the depth of challenge that was being faced.

Federal Government and Support of Technology

By virtue of DARPA creating and hosting the Cyber Grand Challenge, the United States federal government is intimately involved with the development and nurturing of automated vulnerability analysis. DARPA is the advanced research are of the Department of Defense, the military of the United States. Civilian critical infrastructure and US military infrastructure is built upon commercial equipment, software, and protocols which in turn means that vulnerabilities in such systems are national security issues.

The potential drawback of DARPA's involvement is that the same national security issue that cyber vulnerabilities bring to the United States is shared by the potential adversaries of the United States. This means that the same vulnerabilities that endanger the security of the US government and infrastructure are enabling traits for espionage and preparing the battlefield for potential future cyberwar. As such, the technology discussed in this paper, in the hands of a military, may be used for developing exploits to weaponize rather than to patch. While discussing AES, Avgerinos et al states that "a conservative defensive security position must consider the possibility of real-world offensive AEG capabilities" (2014, pg 76).

Conclusion

Vulnerability discovery, on the defenders side, is something that can never happen too fast. From the moment a piece of software is fielded until a vulnerability is patched, their is a window that attacks could be occurring that the defender does not even know are threats. With Heartbleed that window was one where server private keys could be sucked off, which utterly destroys the web of trust underpinning the encryption that enables global ecommerce. The technology that was created to play in the Cyber Grand Challenge has the promise of bringing a window of months down to hours or even minutes. I, for one, would really like to see that.

References

Avgerinos, T., Cha, S. K., Rebert, A., Schwartz, E. J., Woo, M., & Brumley, D. (2014). Automatic exploit generation. *Communications of the ACM*, *57*(2), 74-84.

Borgolte, K. [Cooper]. (2016, Oct 18). Hack.lu 2016 Cyber Grand Shellphish: Shellphish and the DARPA Cyber Grand Challenge. [Video File]. Retrieved from https://youtu.be/nK9L4m0P7Wc

DARPA. (2015a). DARPA: Creating Breakthrough Technologies for National Security. Retrieved Feb 16, 2017 from http://www.darpa.mil/attachments/CreatingBreakthroughechnologiesforNationalSecurity%20Update.pdf

DARPA. (2015b). DARPA Accomplishments: Seminal Contributions to National Security. Retrieved Feb 16, 2017 from http://www.darpa.mil/attachments/DARPAAccomplishmentsSeminalContributionstoNationalSecurity.pdf

DARPA. (2016). DARPA | Cyber Grand Challenge. Retrieved Feb 16, 2017 from http://archive.darpa.mil/cybergrandchallenge/

DEF CON 24 Hacking Conference. (2016). Retrieved from https://www.defcon.org/html/defcon-24/dc-24-index.html

Song, J., & Alves-Foss, J. (2015). The DARPA Cyber Grand Challenge: A Competitor's Perspective. *IEEE Security & Privacy*, *13*(6), 72-76.

Song, J., & Alves-Foss, J. (2016). The DARPA Cyber Grand Challenge: A Competitor's Perspective, Part 2. *IEEE Security & Privacy*, *14*(1), 76-81.

OUTREACH. (2016). “Mayhem” Declared Preliminary Winner of Historic Cyber Grand Challenge. DARPA. Retrieved from http://www.darpa.mil/news-events/2016-08-04