Research Project:
Time-domain visualization

Sponsor
Andy White

Description of Research Problem

Large software systems refer to systems with humongous lines of code, volumes of data, numbers of users and hardware. Systems of such scale produce problems of all sorts as it is governed by developers of multiple organizations, constructed with heterogeneous parts involving complex dependencies and with multiple stakeholders involved, such systems will always evolve over time with human error being the norm. Quite a lot of research is being done in securing these large systems. However, visual analysis of evolution of large software systems with the help of visualization has not been much attention. The crux of the project would involve **developing ways to visualize CMS data that a security analyst could use to figure out where the source code may start to have bugs.**
Tools of such kind could prove immense use to the security analysts as it can lead global as well as local insights about bugs that might creep in at a particular version/time. As humans comprehend visuals more easily than pure numbers, the best way to present CMS (Content Management System) data is through a cohesive picture of the large-scale software check-in/check-out data using visualization.
There are several operations that could be applied such as filtering to hide uninteresting aspects of the source code (libraries that are not used in different versions, but used in earlier versions), a hierarchical approach in determining the frequently contributed local source code and overlaying CVE (common vulnerability exposure) for the CMS data.

Similar Work:
Kiran Lakkaraju et al. designed a system that allow analysts to increase their "situational awareness" by visualizing their current network configuration. The tool is successful in determining various security-related scenarios that it can detect [2]. Xiaoxin Yin et al. utilized parallel coordinates, a well-established concept in visualization to detect anomalous traffic between the local network and external domains [1]. Although the tools do not really cater to our use-case, but visualization techniques have had a certain impact on the way analysts look at security problems.

Statement of Interest

Turning data into picture is something that I enjoy doing, my primary interest in this project is the because of the same. Since last year, I have been working on various visualization and networking projects. It will be interesting to convert long and tedious CMS data into pictures that can decipher the interesting parts of the system. A motivating factor, in pursuing this project, is that analysts and researchers could rely on the tool/methods that we will develop to figure out bugs in software.

Expected Outcomes

The expected outcome of the project would answer these questions:

1. Are source-code updates (check-in, check-out) a good indication of where bugs might occur?
2. What and how certain dependencies of the software are associated bugs are that found with the software?

The contribution for the project in the form of a final report would be:

1. Literature review of security analysis of large software
2. Possibly, if time permits, an extensive visualization tool that is specific to test the above hypothesis
3. Results of the findings and insights that I encounter upon with regard to the evaluation of large systems at a security point of view

Qualifications

From the past summer of 2014, I have been working with the data analysis and visualization group at Lawrence Berkeley National Lab. While working on visualization projects, I was intrigued by the ability of software to convert data into highly informative as well as artistic pictures and animations. This led me to steer my research in the direction of visualization and data analysis. I also have a broad background in networking under my belt. In my bachelor's degree, I have worked on several projects that analyze packets across networks to infer some properties of the data. My skills to apply standard visualization techniques to the security domain, specifically, this problem, makes me a perfect fit for the project. Additionally, if possible I would like to be the scribe of the team that I would be put it.

Bibliography

[1] Yin, Xiaoxin, et al. "VisFlowConnect: netflow visualizations of link relationships for security situational awareness." *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*. ACM, 2004.

[2] Lakkaraju, Kiran, William Yurcik, and Adam J. Lee. "NVisionIP: netflow visualizations of system state for security situational awareness." *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*. ACM, 2004.

[3] Livieri, Simone, et al. "Very-large scale code clone analysis and visualization of open source programs using distributed CCFinder: D-CCFinder." *Proceedings of the 29th international conference on Software Engineering*. IEEE Computer Society, 2007.

[4] Teoh, Soon Tee, et al. "Case study: Interactive visualization for internet security." *Proceedings of the conference on Visualization'02*. IEEE Computer Society, 2002.

[5] Ogawa, Michael, and Kwan-Liu Ma. "Software evolution storylines."*Proceedings of the 5th international symposium on Software visualization*. ACM, 2010.