Insights into Malware Distribution with Graph Analytics

Mingfei Gao[*] Doowon Kim[†] Tongyang Li[‡] Virinchi Srinivas[§]

February 29, 2016

Introduction

Malicious software (malware) destroys and steals access to users' private computer systems, which can lead to breaches of sensitive personal information. It has been rapidly growing, spreading and infecting computer systems; it continues to be an active threat. Currently, more than 200 million unique variants of malware exist. Anti-virus is a software tool that is used to protect against attacks from malware. Lots of works have been dedicated to detecting malware which broadly focused on better understanding the properties of malware such as malware network behaviors. However, this approach is not always able to help detecting new malware because the malware continuously keep evolving by re-packing and obfuscating themselves. Therefore, a new approach, called content-agnostic techniques, has come into the limelight. Content agnostic techniques do not solely rely on the content of various files (benign/malware) and stresses for the need to focus on the relationship between these different files (downloaders). The need for using content agnostic techniques can be motivated with an example that follows.

Executable files can download a variety of auxiliary malware, called payloads, which are the main sources of malware distribution. For example, assume that a user downloads Chrome (web browser). Although the web browser is benign, it can download malicious programs. Moreover, the malicious programs downloads supplementary malware from the Internet. The detection of this kind of the attack is not trivial because downloading software from the Internet is not a clue of malicious intents.

Methodology and Results

This work detects malware downloads by graph analytics. To be more specific, *downloader graphs* are generated by detecting the relationships among downloaded executable files on a

[*] mingfeigaobupt@gmail.com
[†] doowon@outlook.com
[‡] tongyang@cs.umd.edu
[§] virinchimnm@gmail.com

host. This notion can provide a better mechanism to understand and capture the relationship between different files during the software download process.

An example of a downloader graph is shown in Figure 1. From this figure, we can see the basic idea of how the malicious nodes might distribute in the downloader graph. The Internet searching starts at node A which represents a benign Web browser. The node A further downloaded two files(represented by nodes B and D). We don't know if the two files are malicious because we do not have the labels (benign or malicious) of these Nodes. Then Node B downloaded nodes C and F which are labeled as malware. Since node B is connected directly to known malware, we can suggest that node B and all the nodes reachable from node B in the graph are probably involved in malware distribution. For conciseness, these kinds of malware distributions are called influence graphs as shown in the figure. If we could identify the properties of benign and malicious influence graphs in advance, we would have be able to detect malware and abort it from dropping malicious softwares in the machine.

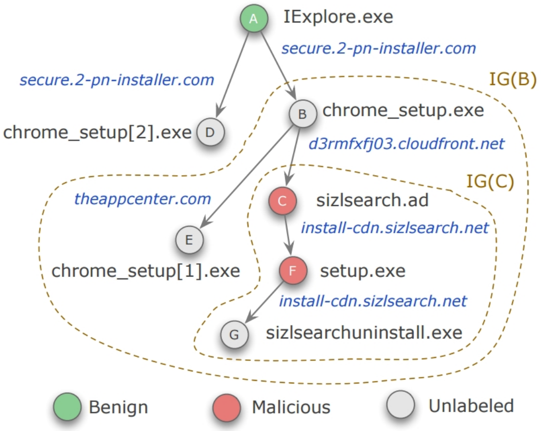

Figure 1: Example of a real downloader graph and the influence graphs of two selected downloaders [1]

Next, *influence graphs* are formulated as the subgraph of the downloader graph on a given host. Interestingly, malicious influence graphs have many distinct properties compared to those benign ones as shown in Figure 2: their diameters are larger, their growth rates are slower, they tend to downed fewer files per domain, and their URL access are quite different from benign ones. Based on these features, a malware detection classifier is built by employing supervised machine learning techniques to separate malicious and benign influence graphs.

Experiments have been done on 19 million downloader graphs from 5 million real hosts using data on a platform for data intensive experiments in cyber security, where the data were known to be malicious or benign. The result shows that the proposed classifier achieves a 96.0% true positive rate, with a 1.0% false positive rate.

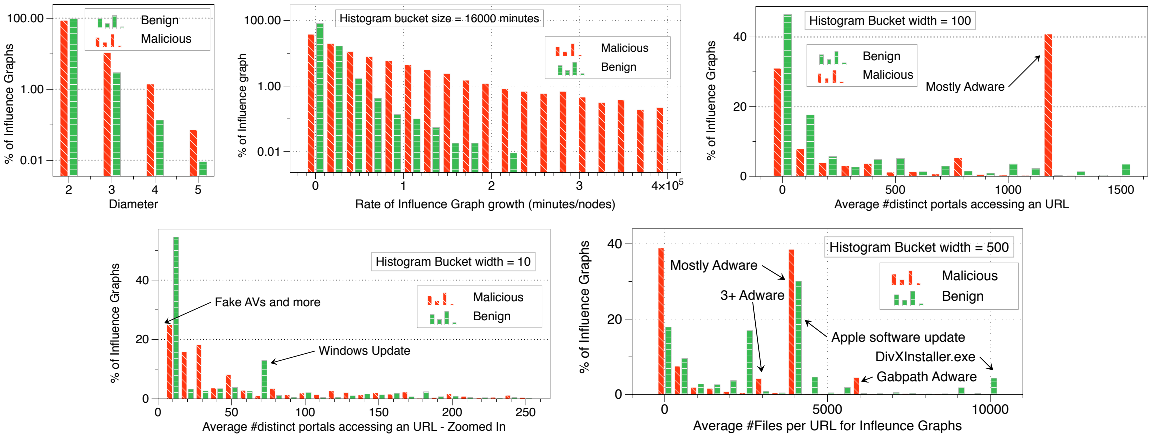

Figure 2: Malicious influence graph vs benign influence graph [1]

Future Work

Given this being the first work in this direction, one could extend this system to identify malicious campaigns in the wild. We could use similar graph abstractions to understand the relation between downloaders, executables and domains they access and how they actively coexist in a given time interval. The system proposed in this work, considers static data setting and an obvious extension would be to build a malware detection system when data is presented in a streamed fashion. This would give rise to a complete end-to-end malware detection system.

References

[1] Kwon, Bum Jun, et al. ”The Dropper Effect: Insights into Malware Distribution with Downloader Graph Analytics.” *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015.

[2] Mavrommatis, Niels Provos Panayiotis, and Moheeb Abu Rajab Fabian Monrose. ”All your iframes point to us.” *USENIX Security Symposium*. 2008.

[3] Zou, Cliff C., and Ryan Cunningham. ”Honeypot-aware advanced botnet construction and maintenance.” *Dependable Systems and Networks, 2006. DSN 2006. International Conference on*. IEEE, 2006.