学位論文　博士（工学）

マルウェアの特徴的振る舞いの誘発によるマルウェア対策手法に関する研究

2013年度

慶應義塾大学大学院理工学研究科

糟谷 正樹

マルウェアの特徴的振る舞いの誘発による マルウェア対策手法に関する研究

糟谷 正樹

論文要旨

マルウェア対策はコンピュータセキュリティ上の重要な課題である．コンピュータがマルウェアに感染するとログインアカウントやパスワードなどの個人情報が盗まれたり，詐欺被害に遭うなど様々な被害を受ける．セキュリティベンダの報告によると，2012 年に新たに報告されたマルウェアは 4,000 万件以上に上る．そのため，防御側は増え続けるマルウェアに対して適切に対策を行う必要がある．

マルウェア作成の技術は向上し続けているため，精巧に作られたマルウェアに対処することは困難である．マルウェア作成者は polymorphism や metamorphism を利用してアンチウイルスソフトを回避する．アンチウイルスソフトが利用するシグネチャマッチングと呼ばれる方法は，既知のマルウェアを解析して得たシグネチャと検査対象のプログラムのバイナリ列が一致するかを調べる．そのため，僅かにバイナリを変更することによりマルウェアは容易にシグネチャマッチングを回避できる．一方，振る舞い検出による対策はマルウェア間に共通する振る舞いが検査対象のプログラムに含まれるかを検査する．ここでいう振る舞いとはシステムコールや API (Application Programmers Interface) の呼び出し列のことであり，静的解析や動的解析を利用して振る舞いを抽出する．しかし，マルウェアの多くは難読化を利用して，プログラムの振る舞いを正しく抽出することを困難にしている．動的解析は難読化の影響を受けないもののステルス性の高いマルウェアに対処できていない．ステルス性の高いマルウェアは頻繁な動作を避けて振る舞いを抽出することを困難にしたり，得られた振る舞いから悪意ある振る舞いを含むかどうかの判別を困難にするためである．

本論文では，マルウェアの特徴的な振る舞いを誘発するための作為的な環境を用意して，ステルス性の高いマルウェアに対処する方法を提案する．本手法はマルウェアの種類に応じた入力を作為的に与える環境を用意することにより，マルウェアが実行せざるを得ない状況を作り出す．その結果，頻繁な動作を避けるマルウェアを活性化させたり，得られた振る舞いが悪意ある動作を含むかどうかの

判別を困難にするマルウェアに作為的に特徴的な動作を誘発させることを可能とする．本論文では，提案手法の具体例としてアドウェア・スパイウェアと偽アンチウイルスソフトの 2 種類に対処する．これらは現在においても多くの影響を与えているマルウェアのためである．

アドウェアとスパイウェアは頻繁な動作を避けて自身の悪意ある振る舞いを隠す．そのため，効率良く詳細な解析を行うためには，アドウェア・スパイウェアを活性化させる刺激を外部から与えて，強制的に動作させる環境を用意することが好ましい．これを実現するために，本研究では Blayzard を提案する．Blayzard は Internet Explorer (IE) のアドオンである Browser Helper Object として実現されており，偽装した大量の IE イベントを作り出し，アドウェアやスパイウェアに挿入するシステムである．偽のイベントに騙されたアドウェアやスパイウェアはその挙動を活性化させるため，頻繁な動作を避けるアドウェア・スパイウェアであっても解析を行うことができる．Blayzard の有用性を確認するために，32 個のアドウェア・スパイウェアと 10 個の無害な BHO を利用した結果，Blayzard は全ての検体の挙動を活性化させて，その振る舞いを解析することができた．

偽アンチウイルスソフトは正規のアンチウイルスソフトの挙動を考慮して動作するため，得られた振る舞いが悪意ある動作を含むかどうかを判別することは難しい．この問題を解決するために，正規のアンチウイルスソフトと偽アンチウイルスソフトの違いを誘発する方法と，両者の判別を行う指標を発見することは重要である．本論文では，マルウェアを利用して振る舞いの違いを誘発する．マルウェアのあり・なしの環境を用意して，両方のアンチウイルスソフトを動作させる．その後，環境の違いで振る舞いに差が生じるかどうかを調べ，正規か偽アンチウイルスソフトであるか判別する．実験的な調査から，振る舞いの違いを判別する際にメモリ使用量が最も精度よく判別できることを示す．その違いを機械的に判別するために，統計手法であるリーベン検定を利用する．提案手法の有効性を示すために 39 個の偽アンチウイルスソフトと 8 個の正規のアンチウイルスソフトを用いた結果，各々を正しく分類することができた．正規のアンチウイルスソフトはマルウェアが存在する環境でのみメモリを著しく使用したのに対して，偽アンチウイルスソフトは環境の違いに関わらず，メモリ使用量が変わることはなかった．

A Study on Countermeasures against Malware by Stimulating Behaviors Characteristic of Malware

Masaki Kasuya

Abstract

Tackling malware is an important challenge to computer security. Malware damages computer systems by stealing personal information such as login account and password and/or makes money by deceiving victim users. A security vendor reported that it found over 40 million samples of malware in 2012. To protect computer systems from malware, Defenders and researchers must reveal behaviors of malware, and develop practical countermeasure systems against it.

It is difficult to detect sophisticated malware instances. Malware developers frequently use the technique of polymorphism and metamorphism, and try to evade traditional malware detector, or antivirus software (AV). AV uses signature-based approach that identifies known malware instances by comparing the binary image of a number of uniquely characterizing signatures. It is purely syntactic and ignores semantics of instructions. In contrast to signature-based approach, behavior-based approach is easy to catch semantic information. The approach extracts system calls and API (Application Programmers Interface) calls by using static analysis or dynamic analysis. However, static analysis is evaded by obfuscation technique because it is difficult to make analysts extract correct behaviors. Dynamic analysis is resistant to obfuscation technique but cannot capture behaviors of stealthy malware because stealthy malware avoids showing malicious behaviors frequently and makes it difficult to infer whether the behaviors are benign or not.

This dissertation introduces a novel approach to deal with stealthy malware. To this end, an execution environment to inject "feed" inputs considering kinds of malware is prepared to intentionally stimulate and extract behaviors characteristic of malware. Even stealthy malware is activated and the behaviors are revealed. As a result, this approach can efficiently extract behaviors that stealthy malware executes occasionally and identify whether extracted behaviors are benign or malicious. To show the usefulness, this dissertation conducts two case studies, adware/spyware and fake AV, because

they are known as dominant threats in computer security. In principle, this approach is applicable to other kinds of malware.

Adware and spyware avoid frequent execution to hide their malicious behaviors. To perform the detail analysis, it is needed to extract their behaviors by giving adware and spyware "feed" to stimulate their behaviors. To this end, an environment to inject "feed" inputs is prepared. Since the environment is able to inject a large amount of feed inputs, it is capable to make adware and spyware activate their behaviors. As one example of the idea, Blayzard, a Browser Helper Object (BHO) of Internet Explorer (IE), is implemented. It generates bogus several IEs' events and injects them into adware and spyware instances. As a result, adware and spyware are deceived by the bogus events, and reveal the behaviors characteristic of them. To show the effectiveness, 32 adware/spyware instances and 10 benign BHOs have been used. In this experiment, Blayzard could extract the behaviors of 32 malicious instances and 10 benign samples.

Fake AV mimics behaviors of legitimate AV. Since it is difficult to judge whether AV-like behaviors are benign or not, it is important to find an indicator to distinguish legitimate AV and fake AV, and stimulate behaviors characteristic of fake AV. To this end, two environments, a clean environment including no malware and an infected environment including malware are prepared to bring out differences of behaviors between legitimate AV and fake AV. This approach uses malware as "feed". Experimental investigation shows memory usage is a good indicator. In this experiment, Levene Test, a statistical test, correctly identifies all fake AV samples (39 out of 39) as fake and all legitimate AV products (8 out of 8) as legitimate. All legitimate AV products significantly consume the memory usages in infected environments, but fake AV hardly changes the usages between clean and infected environments.