SYNGRESS

How to Defeat Advanced Malware

New Tools for Protection and Forensics

Henry Dalziel

How to Defeat Advanced Malware

New Tools for Protection and Forensics

Page left intentionally blank

How to Defeat Advanced Malware

New Tools for Protection and Forensics

Henry Dalziel

AMSTERDAM • BOSTON • HEIDELBERG
LONDON • NEW YORK • OXFORD • PARIS
SAN DIEGO • SAN FRANCISCO
SINGAPORE • SYDNEY • TOKYO
Syngress is an Imprint of Elsevier

Syngress is an imprint of Elsevier
225 Wyman Street, Waltham, MA 02451, USA

Notices

Knowledge and best practice in this field are constantly changing. As new research and experience broaden our understanding, changes in research methods or professional practices, may become necessary. Practitioners and researchers must always rely on their own experience and knowledge in evaluating and using any information or methods described here in. In using such information or methods they should be mindful of their own safety and the safety of others, including parties for whom they have a professional responsibility.

To the fullest extent of the law, neither the Publisher nor the authors, contributors, or editors, assume any liability for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions, or ideas contained in the material herein.

British Library Cataloguing-in-Publication Data
A catalogue record for this book is available from the British Library.

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the Library of Congress.

ISBN: 978-0-12-802731-8

TABLE OF CONTENTS

AUTHOR BIOGRAPHY

Henry Dalziel is a serial education entrepreneur, founder of Concise Ac Ltd, online cybersecurity blogger, and e-book author. He writes for the blog "Concise-Courses.com" and has developed numerous cybersecurity continuing education courses and books. Concise Ac Ltd develops and distributes continuing education content (books and courses) for cybersecurity professionals seeking skill enhancement and career advancement. The company was recently accepted onto the UK Trade & Investment's (UKTI) Global Entrepreneur Programme (GEP).

Page left intentionally blank

CONTRIBUTING EDITOR BIOGRAPHY

Simon Crosby is cofounder and CTO at Bromium and The Bromium Labs. The Bromium Labs team of security analysts has extensive experience in building innovative technologies to counter and defend against advanced attacks. While Bromium has created an innovative new technology called microvirtualization to address the enterprise security problem and provide protection for end users against advanced malware.

Page left intentionally blank

CHAPTER 1

A Primer on Detection for Security

The security industry has relied for years on end point protection software that aims to detect specific behavioral patterns signatures of malware in order to protect a system under attack. Most signatures today attempt to capture key behavioral patterns of all variants of a particular exploit or class of malware. In fact, McAfee now reports identifying more than 75,000 unique variants of malware per day, most of which are slight variants on a few successful attacks, on a single vulnerability. If one can accurately capture the pattern, a single signature can deal with many variants. This approach is the key to success: The average ".dat" signature file measures 100 MB in size, and with thousands being added every day (Symantec[1] created more than 10 million unique signatures in 2010), the problem of distributing signatures to endpoints has become severe with the net result that PCs can remain unprotected for a long time.

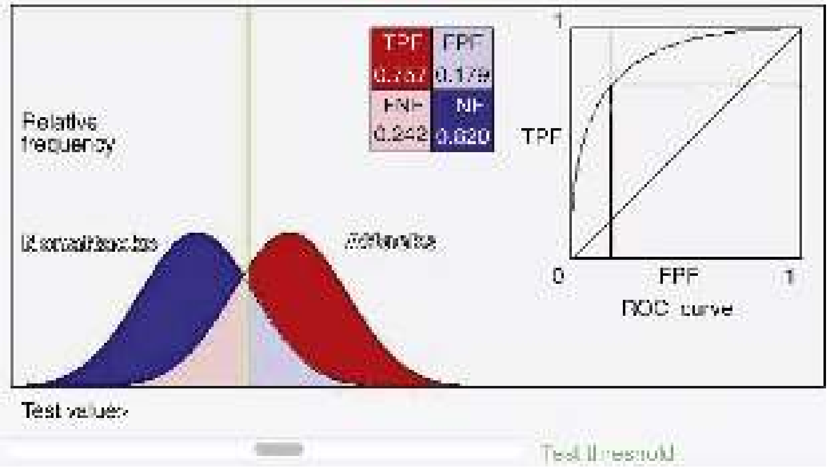
All detectors must be evaluated for accuracy against four key metrics, namely (for a given sample) the proportion of {True Positive, True Negative, False Positive, False Negative} results that the detector produces. The meaning of these is straightforward:

- *TPF*: The frequency of samples that contained attacks and that was correctly identified
- *TNF*: The frequency of samples that did not contain an attack and was not identified
- *FPF*: The frequency of samples that was incorrectly identified as containing an attack, and
- *FNF*: The frequency of samples that contained a real attack that was not identified.

The ROC curve and the four fractions listed above can be shown graphically as the areas of intersection of two statistical distributions. The distributions plot the value of the detector (e.g., the degree of suspicion

[1] Wired Business Media, January 06, 2012 "Symantec Confirms Hackers Accessed Source Code of Two Enterprise Security Products."

of the detector that a particular event is a real attack) for both nonattack traffic and the actual attacks. An example ROC curve is shown below.



Every detector has a threshold at which it will trigger an alarm, and setting the threshold is critical to the utility of the detector in practice. What is the key is the ability of the detector to separate real attacks from normal traffic. A better detector separates the two curves more cleanly, leaving less overlap. The challenge is to accurately detect attacks given the enormous number of slight variations in malware that can be easily generated by an attacker, without increasing the False Positive or False Negative frequencies to the point that the detector is not useful.

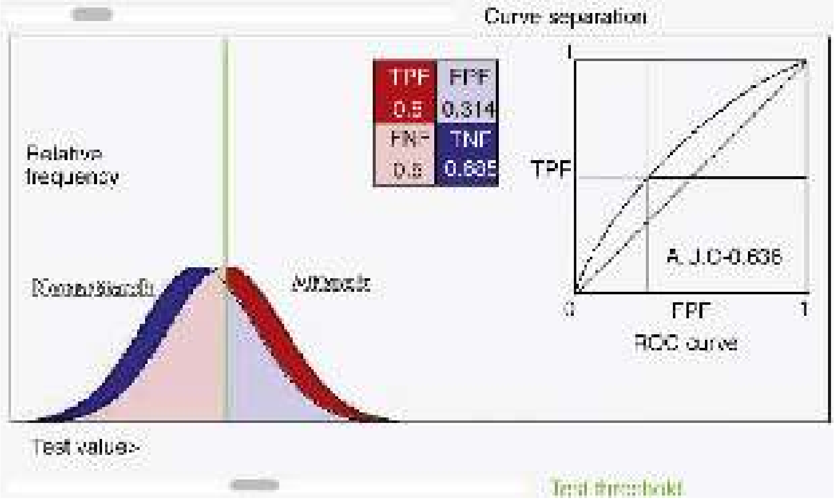
It is important to understand that:

1. No detector is perfect. When a detector fails (False Negative), the attacker will succeed.
2. Tuning a detector is a careful balance of trading off False Positives (which train users/IT teams to ignore alarms) against False Negatives (which in turn allow attackers to successfully avoid detection), and doing so requires careful analysis by experts, and a large, relevant data set to check against.
3. Unfortunately today's rapidly moving front of highly tailored malware adapts fast, leaves no time for human assessment, and makes historical attack data sets used to tune detectors significantly less useful.

4. It has been proven that it is impossible to build a useful signature-based detector for polymorphic malware: “The challenge of signature-based detection is to model a space on the order of O(28*n*) signatures to catch attacks hidden by polymorphism. To cover thirty-byte decoders requires O(2240) potential signatures; for comparison there exist an estimated 280 atoms in the universe.”[2]

1.1 TODAY’S APPROACH: “COMPROMISE-FIRST DETECTION”

The endpoint protection industry (EPP) today relies on classic signature-based attack detection. We call this “compromise-first detection” because the increasing difficulty of differentiating between normal and attacker behavior has resulted in both high False Positives and high False Negatives. This occurs when the detector is unable to sufficiently distinguish between attack and non-attack traffic, causing significant overlap of the two distributions measured by the detector, as shown further. The ratio of the TPF to FPF is sometimes called the signal to noise ratio (SNR). A low SNR loses True Positives in a sea of False Positives, training users, and administrators to ignore warnings, and wasting the time of security staff.



[2] On the Infeasibility of Modeling Polymorphic Shellcode, Columbia University.

As a result, the EPP industry has come to rely heavily on detectors that are sufficiently accurate only if they detect malware when it actually compromises the system, for example, when it overwrites a key Windows system dynamic-link library (DLL) or registry entry, or persists a file with a known-bad signature. Unfortunately, at this point, the system has already been compromised and must at the very least be reimaged, incurring costs to IT and downtime for users. Worse still, sophisticated attacks are crafted to immediately take advantage of an exploit, so with this type of detection, by the time the alert has been raised or blocking initiated (such as terminating a connection), the attacker may already have achieved his/her goal, such as stealing a file or moving deeper into the enterprise infrastructure. From the moment an attacker first compromises a single machine, the cost of remediation increases exponentially with time, because the attacker will rapidly penetrate deeper into the enterprise, causing more damage, requiring substantial additional remediation, and exposing more users and data.

Compromise-first detection is problematic. Delays in signature distribution together with detector inaccuracy aid the attacker, and the cost of remediation is high – all systems that might have been penetrated must be reimaged.

Ultimately, EPP vendors face an impossible challenge trading off False Positives versus False Negatives: They lose either way, and so do their customers.

CHAPTER 2

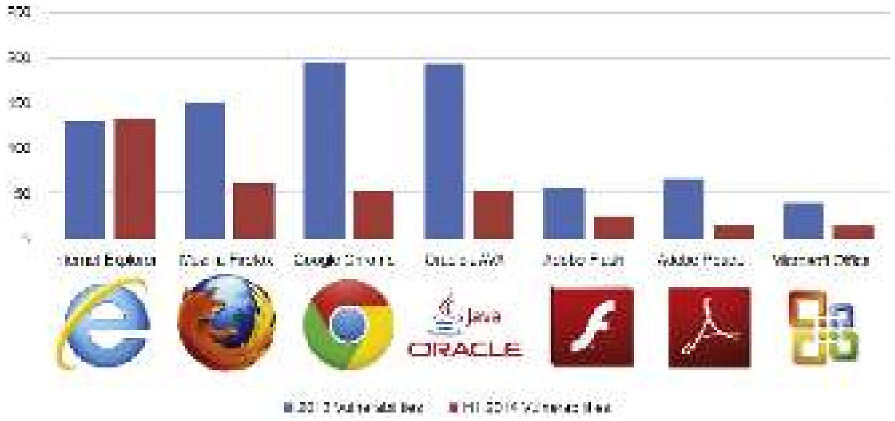
2014 Endpoint Exploitation Trends

Before analyzing potential solutions, security teams tasked with protecting critical enterprise assets must track the shifting attack landscape to understand key attack methods and targets. The Author, in conjunction with Bromium Labs, a team of security analysts with extensive experience in building innovative technologies to counter and defend against advanced attacks, studied key trends in the 2014 cyber attack landscape. These latest trends are summarized below and should be factored into security planning in the coming months:

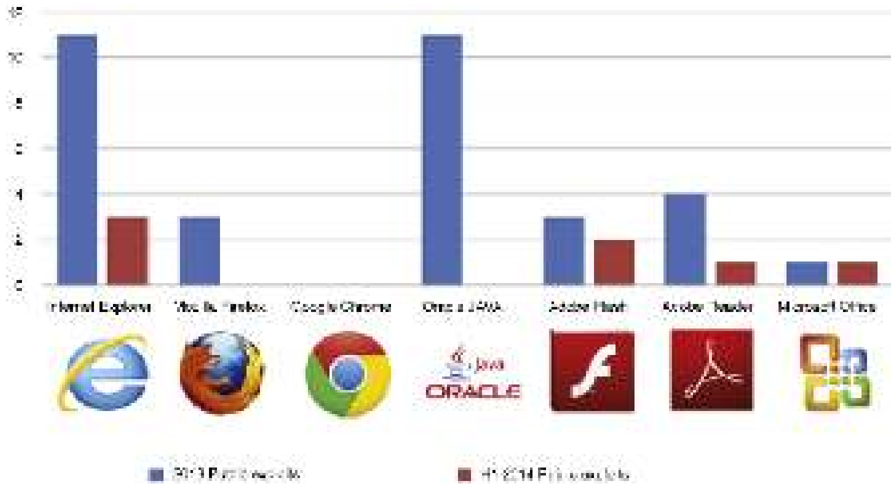
1. Microsoft® Internet Explorer set a record high for reported vulnerabilities in the first half of 2014.
2. Microsoft Internet Explorer also leads in publicly reported exploits.
3. Web browser release cycles are becoming more frequent – as are initial security patches.
4. Adobe Flash is the primary browser plugin being targeted by 2014 zero-day attacks.
5. New "Action Script Spray" techniques targeting Flash have been uncovered that exploit zero-day vulnerabilities.

2.1 ZERO-DAY TRENDS

In the first half of 2014, the growth in zero-day exploitation continued unabated from 2013. Unsurprisingly, all of the zero-day attacks targeted end-user applications such as browsers and applications such as Microsoft® Office. Typically these attacks were launched using classic spear-phishing tactics. Although Microsoft Internet Explorer was the most patched product on the market, it was also the most exploited, surpassing Oracle Java and Adobe Flash. Bromium Labs believes that Microsoft Internet Explorer will likely continue to be the target of choice going forward.



Source: NVD.

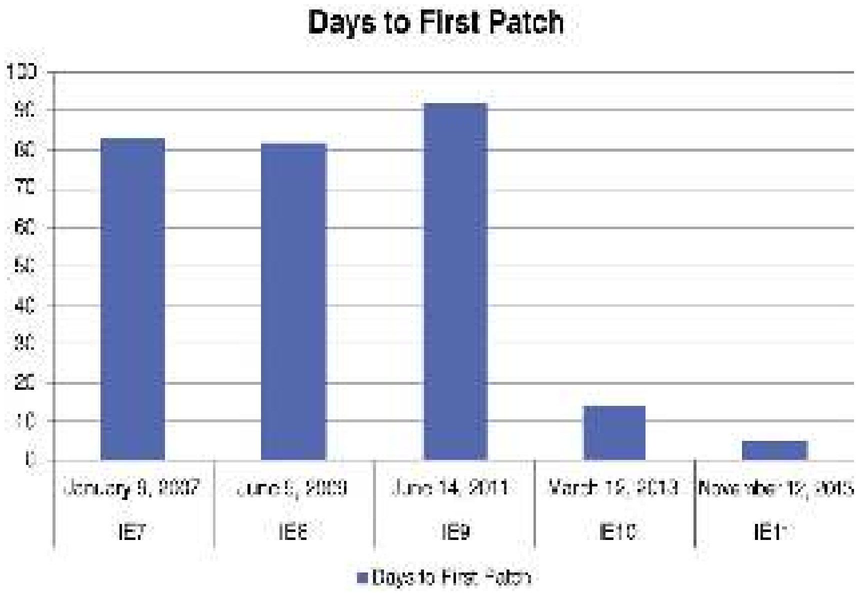


Source: Exploit-db.com.

In comparison, Java had no reported zero-day exploitation in the first half of 2014.

Released in late 2013, Microsoft Internet Explorer 11 has seen a quick succession of security patches, compared to its predecessors. Bromium Labs analyzed the timelines for each Internet Explorer patch release and documented when the first critical patch became Generally Available (GA).

Internet Explorer release to patch timeline.



2.2 NOTABLE ZERO-DAY EXPLOITATION TECHNIQUES

Microsoft Internet Explorer

- Almost all Microsoft Internet Explorer memory corruption exploits now use de facto ROP (Return Oriented Programming) techniques for bypassing the default operating system security mechanisms (address space layout randomization (ASLR), data execution prevention (DEP)).
- Both the Microsoft Internet Explorer zero-day exploits leveraged "Action Script Spray" technique to bypass ASLR.

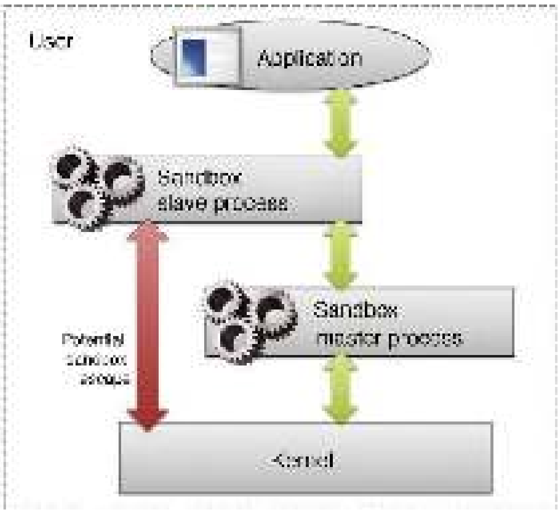
Adobe Flash

- Attackers were quick to leverage new features released in late 2013 to exploit ActionScript Virtual Machine ASVM implementation flaws using "Action Script Spray" techniques.
- Non-ASLR libraries continued to be the weakest link leveraged by malware authors to bypass OS protections.

Adobe Reader Sandbox Escape

- This vulnerability was uncovered late in 2013 and was finally patched in January 2014.

- Two vulnerabilities were used to bypass the Adobe Reader sandbox:
  - o CVE-2013-3346: Use-after-free vulnerability in Adobe Reader
  - o CVE-2013-5065: Kernel-mode zero day vulnerability NDProxy.sys



Adobe Flash Player and Recent Client Exploits

2010–2013 were clearly the years of Java exploits. Since then, a lot has changed: old versions of JRE are blocked by default, Java applets now require explicit activation from users resulting in this attack vector becoming harder to leverage. In response to increased defense deployed by security vendors and software developers, attackers have switched to new plugins. In the past 6 months, Adobe Flash Player was seen to be abused leveraging two attack vectors:

- Exploiting ASVM vulnerabilities
- Abetting exploitation of IE UAF bugs

2.3 EMERGING ZERO-DAY EXPLOITATION TECHNIQUES

Action Script Virtual Machine Attacks

In 2014, there were three severe vulnerabilities that were detected in live attacks. Unlike Java, where in the main, malicious code leveraged JRE's capabilities, Flash exploits require DEP and ASLR bypass for successful execution. The following table provides a summary of 2014 ASVM attacks.

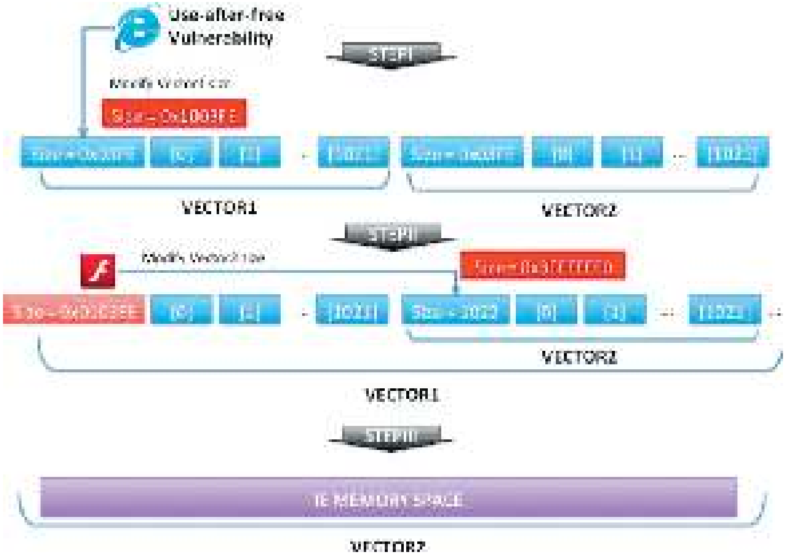
CVE	Vulnerability	Exploitation Technique
2014-0497	N/A	Non-ASLR libraries of Flash Player
2014-0502	Double Free of AS3 Shared Object	Non-ASLR libraries of JRE 1.6 and 1.7 and MS Office 2007 and 2010, ROP chain is built relative to fixed offset
2014-0515	Heap overflow in compiled Shader	Dynamic ROP generation based on Action Script Spray

Unlike the first two exploits, CVE-2014-0515 used a relatively new technique to bypass ASLR allowing dynamic crafting of ROP chain called Action Script Spray. This technique was also seen in two IE exploits released in 2014.

ROP Bypass Using Action Script Spray

Both IE exploits released in 2014 (CVE-2014-1776, CVE-2014-0322) used Flash to build the ROP chain and launch shellcode. This technique leverages the way dense arrays are allocated in the endpoints memory.

If a vulnerability allows an attacker to control the size of a vector they could make it as big as the whole memory space and then search for the necessary API calls and ROP gadgets. The following picture illustrates an Action Script Spray attack.



If the whole process memory is accessible, an attacker can now craft an ROP chain using ASVM capabilities and modify vtable with a pointer to the shellcode and trigger it.

The attack is more complex than a traditional heap spray, which indicates that cybercriminals are ready to invest more time and resources into development of new techniques in response to ever increasing protection measures. In addition to that, the prevalence of IE + Flash is much higher than IE + Java JRE, so this has provided attackers with a larger opportunity.

CHAPTER 3

The Proposed Solution

Computing has changed dramatically over the past decade. Even the most prophetic among us could not have foreseen how cloud computing was going to shift and "even out" the playing field, specifically as it relates to computer storage and networking. Pay as you go cloud infrastructure for application developers and affordable, powerful, touch-enabled mobile devices have transformed client computing forever.

The future of computing continues to be reshaped by powerful forces: cloud-based applications continue to grow in popularity, accessed in the main by personally owned mobile telephones, tablets and computers, via an inherently unsafe internet. And as a result, operating systems, networks, and applications will continue to be susceptible to attack, and although we can expect this challenge to be met head-on by cloud service providers, clearly the sheer scale of the bring your own device (BYOD) phenomenon would indicate that the same cannot be said for client devices. So it stands to reason that computer systems must defend themselves "by design." Significant infrastructural and trust-related changes are needed in this "cloud-mobile" era. Defense must be an intrinsic element of computer system design.

At the heart of this issue is "Trustworthy Computing."[1] Our goal is to propose a new systems architecture solution that not only answers the security needs of future systems, to combat, for example, the zero-day exploits outlined above, but more importantly, a system architecture that deals with our existing "leaky" end-point legacy systems (which continue to be the front line), and offer up the most vulnerable operating systems and applications to attack. Although the concepts we discuss could also be applied to server-class systems, our main focus here is on client devices.

[1] Wikipedia, "Trustworthy Computing" [Online]

3.1 THE PRINCIPLE OF LEAST PRIVILEGE

Humans are inherently social, and our notion of trust is innate. In fact, trust has always been closely associated to survival. We routinely limit the amount of information that we share with others on the basis of what we feel they needs to know. Information, if one were to apply a digital analogy, is shared on a "policy of least privilege."

Although we can understand this instinctively, one of the inherent challenges in cyber security is accommodating the fact that humans also expect their computer systems to have the same ability, to switch between trust domains, and decide what information should be shared, how it should be shared, and what level of access somebody should have to it. We see no issue with using the same mobile device to chat via Twitter, for example, whereas moments later, check our personal bank balances. Phishing attacks continue to grow in popularity, and the consequences of an uninformed user clicking what looks to be a legitimate link in an e-mail, only to see their action invite malware that attacks vulnerability in an operating system, are all too familiar.

The challenge security teams face is both to protect their networks and simultaneously allow their employees to leverage the productivity benefits afforded by, for example, social media and cloud-based applications.

This reality is further complicated by the very business model the "free" Internet has been built around. Online advertising companies and search engines benefit from compromised security. For example, many sites require personal information from users, and make money by selling that information to marketing firms and vendors. A user may be persuaded that a site will respect the user's right to privacy, even when the implicit exchange is free service for the right to sell your data.

That instinctive ability to determine the level of privilege somebody should have in a social relationship is dependent upon "granularity." Unfortunately, today's operating systems (OSes) and applications (e.g., web browsers) are incapable of providing either a similar degree of granularity, or effective embodiment of trust domains, or confinement to apply the concept of least privilege. Critical OS design concepts come from a pre-internet age, where designers did not have to take into account targeted attacks that exploit unpatched weaknesses within the operating system or software, or deliberate monitoring systems that jeopardize individual privacy.

Although all operating systems utilize some kind of software isolation (e.g., sandboxing), access controls, and hardware defense (e.g., user and kernel modes) to segment applications, OS services and data, with the objective of applying least privilege, they cannot manage their inherent, latent vulnerability.

Operating systems offer hackers an enormous attack surface (e.g., the Windows operating system and Android mobile operating systems have approximately 50,000,000 and 10,000,000 lines of code respectively[2]). Mobile device market differentiation boils down to a constantly growing feature list, but it is exactly those features that expose the consumers mobile device to vulnerabilities – approximately 1 significant defect/KLOC that can allow an attacker to increase execution rights and compromise the computer to get into both local and remote resources.[3]

Consumers are also susceptible to the existence of applications that allow websites and search engines to monitor their behavior and betray privacy. Often these applications (e.g., Google Chrome) come from companies whose very aim is to profit from their monitoring of consumers, while apparently offering value (functionality, or claims of security) within their applications. Although privacy is a sophisticated subject that requires an extensive attention on its own, it likewise utilizes a solid implementation of least privilege. Both security and privacy necessitate that our computers are trustworthy.

3.2 DETECTION'S FOLLY

Even if the battle between attackers and security vendors is heavily weighted in the attackers favor, the $70 BN cyber security industry hinges its livelihood on identifying malicious behavior. It is our contributing editor's belief (Bromium Labs), however, that this premise is not only flawed, but mathematically impossible.[4] Simply put, vendors will never be able to reliably detect polymorphic malware in order to block it.

[2] Wikipedia, "Source Lines of Code"
[3] C. Perrin, "The danger of complexity: more code, more bugs," TechRepublic
[4] Wikipedia, "The Halting Problem"