Software, Vendors and Reputation: An Analysis of the Dilemma in Creating Secure Software

Craig S. Wright

School of Computing and Mathematics
Charles Sturt University
Wagga Wagga, NSW, Australia
craig.wright@information-defense.com

Abstract. Market models for software vulnerabilities have been disparaged in the past citing how these do little to lower the risk of insecure software. This leads to the common call for yet more legislation against vendors and other producers in order to lower the risk of insecure software. We argue that the call for nationalized intervention does not decrease risk, but rather the user of software has an economic choice in selecting features over security. In this paper, we investigate the economic impact of various decisions as a means of determining the optimal distribution of costs and liability when applied to information security and in particular when assigning costs in software engineering. The users of a software product act rationally when weighing software risks and costs. The choice of delivering features and averting risk is not an option demanded by the end user. After all, it is of little value to increase the cost per unit of software if this means that users purchase the alternative product with more features. We argue that the market models proposed are flawed and not the concept of a market itself.

Keywords: Security, Derivatives, Vulnerability Market, Software Development, Game theory, SDLC (Software Development Life Cycle), DMCA (Digital Millennium Copyright Act), IDS (Intrusion Detection System), MTTF (Mean Time To Failure), Ploc (per (source) Lines of Code).

1 Introduction

This paper seek to argue that a well-defined software risk derivative market would improve the information exchange for both the software user and vendor removing the oft touted imperfect information state that is said to belie the software industry. In this way, users could have a rational means of accurately judging software risks and costs and as such the vendor could optimally apply their time between delivering features and averting risk in a manner demanded by the end user. After all, it is of little value to increase the cost per unit of software by more than an equal compensating control.

Arora, Telang and Xu [2, 3] asserted that a market based mechanism for software vulnerabilities will necessarily underperform a CERT-type mechanism. The market that they used was a game theoretic *pricing game* [14]. In the model reported, the

L. Chen and M. Yung (Eds.): INTRUST 2010, LNCS 6802, pp. 346–360, 2011.

players in the market do not report their prices[1]. These players use a model where information is simultaneously distributed to the client of the player and the vendor. The CERT model was touted as being optimal. It relies on waiting until a patch was publically released and only then releasing the patch to the public. This ignores many externalities and assumes the only control is a patch in place of other alternative compensating controls.

Consequently, the examined "market" model is in itself sub-optimal. It both creates incentives to leak information without proper safeguards and creates vulnerability black-markets. As criminal groups and selected security vendors (such as Penetration testers and IDS vendors) have an incentive to gain information secretly[2], they have an incentive to pay more for unknown vulnerabilities in a closed market. This means that a seller to one of these parties has a reputational incentive to earn more through not releasing information as the individual's reputation will be based on their ability to maintain secrecy.

This misaligned incentive creates a sub-optimal market. As a consequence, the market reported (Arora, Telang and Xu 2005; Kannan and Telang 2004) was sub-optimal. This CERT based model was an inefficient market. This does nothing to imply that all markets are less effective. The skewed incentivisation structure recommended in the paper was the source of the inefficiency and not the market itself. This simply highlights the need to allow efficient markets to develop rather than seeking to create these through design.

The other argument posed comes as a consequence of information asymmetry. It is asserted [14] that software vendors have an informational advantage over other parties. The vendor does have access to source code (which is also available for Linux and other open source providers), but it can be proved that this does not provide the levels of information asymmetry that are asserted. Software vendors have a reputational input to their value [8,20].

Telang & Wattal [20] did note that the market value of a software vendor is influenced through reputational costs and those vulnerabilities correlate significantly with a decrease in the companies traded price, a view supported by others [8].

> "*Vulnerability disclosure adversely and significantly affects the stock performance of a software vendor. We show that, on average, a software vendor loses around 0.63% of market value on the day of the vulnerability announcement. This translates to a dollar amount of $0.86 billion loss in market value. We also show that markets do not penalize a vendor any more if the vulnerability is discovered by a third party than by the vendor itself.*"

These results demonstrate that a vendor has an incentive to minimize the vulnerabilities found in their products. If an excessive number of vulnerabilities continue to impact a vendor, their market capitalization suffers as a consequence. This

[1] E.g. iDefense Ltd. And other similar providers have a semi-closed market with limited information exchange.

[2] Criminal groups have an incentive to maximize the time that vulnerabilities remain unknown as this extends the time that they have to exploit these bugs. Penetration Testers etc. have similar incentives as the trade secret of an unknown zero day vulnerability can provide them with a competitive advantage. This also goes to reputational motives for both parties.