BeeWise:
A Futures Market for Fostering Security by Design

Opening

All right, thank you for coming today.

how many futures software security can have? if we believe that managing risks is about changing the future, also software security may have more than one possible future.

and this talk is about a prediction market that might help us in making, among all possible futures for software security, the future where: - information symmetry is finally established between buyers and sellers, - software manufacturers have incentives to build security in, and - software buyers have a financial instrument to hedge against information security risks.

the story of software security futures is the story of Beewise, an experimental project about security metrics, quantification of risks and estimations for the probability of loss events.

Context

Today, security stakeholders face a challenging task in assessing the risks they are exposed to, as they have incomplete information about the number and the severity of vulnerabilities affecting their systems. As a matter of fact, current economic, regulatory and legal incentives are misaligned, distorted or ineffectual.

Software defects are the broken windows of our information society. They broadcast a

message of disorder, inviting more elements of disorder, or even cyber crime.

The reality is that:

”Software products have among the highest levels of defects of any products sold today, and there is very little accountability on the part of producers and software products.”

In order to understand the story of beewise, we need to understand the story of: - bugs & carrots; - M&Ms; - bees & crowd; and - back to the future

Bug

the story of bugs and carrots. it’s the story of why we do things. and why software makers, buyers and cyber attackers behave in the way they do.

for the sake of speech time, it will suffice to note how today…

Due to several economic reasons, namely network externalities and information asymmetries, neither vendors have incentives to build sound security technology into their products nor are users willing to spend extra money on security technology (Ross Anderson, Why Information Security is Hard)

so, from the supply side, the market of lemons suggests that vendors under-supply security to the market. from the demand side, the tragedy of commons tells us that users demands less security than appropriate.

what we observe here is what economists call a market failure.

a market failure is about the inability to self-correct. - software-manufacturers will not forgo markets share. - software buyers will not forgo features. - cyber attackers will not forgo attacking tens of millions of vulnerable systems.

how to invert this market? how do we change?

M&M

in order to change, we need new incentives.

when people engage in activities that imposes high social costs, it usually means that private cost is to low... ... to change behavior then, require raising the private cost of a particular activity.

economists would call this internalize the externalities

we have two way to that, and they are not mutually exclusive.

- establishing laws and regulations in the information security area (fixing liability models, security will not be perceived as an externality anymore) - establishing new markets with feedback mechanisms (hence balancing the information between buyers and sellers, mitigating the problems at their source)

here i want to focus on the second strategy, and more specifically on market mechanisms that can contribute towards establishing information symmetry.

the key realization is that the observation of economic agents' decision yields useful indicators for their expectations and can eventually be used to construct operable metrics.

the kind of metrics arising from market mechanisms are forward-looking, because market prices are based on expectation about the future rather than on historical data.

now, if we look at the countermeasures proposed in the literature some of them stimulate new markets and therefore are not only good tools to align incentives, but also to obtain a new class of security metrics.

vulnerabilities as a security-related information can be traded. a number of vulnerability markets have been proposed so far. they range from:

- bug challenges, where the software manufacturer offers a monetary reward for reported bugs - bug auctions, where vulnerabilities are offered on online auctions and security firm allotting rewards for vulnerability reports - vulnerability brokers, or vulnerability sharing circles build around indipendent organizations who offer money for vulnerability reports - cyber insurance, where coverage for security breaches is offered by carriers to subscribing organization. the premium is assumed to be adjusted to individual risks, which demands on the IT system in use and the security mechanisms in place - exploit derivatives, where the the mechanism of binary options is transfered from finance to computer security

the ideal vulnerability market fulfill three, plus one, functions:

- information function - the ability to use market prices as forward-looking indicators of security properties (ie, countering the lemons effect) - incentive function - allow monetary compensation for security research and development (high priority to security issues) - risk-balancing function - the market provides instruments to hedge against large information security risks (ie, mitigate the impact of occasional security breaches) and - efficiency - orthogonal to the other functions and characterized in terms of: low transaction costs, liquidity, transparency, accountability.

with bug challenges and auctions there is no possibility to do risk-balancing at all; more-

over the information obtained about the market price from a bug challenge is only a lower bound.

brokers not always leads to the disclosure of the vulnerability to the public.

exploit derivatives and cyber insurance looks both promising. the former provides a timely indicator, while insurances can be less efficient due to high transaction costs, bad portfolio balancing or high correlation risks.

let's giver a closer look to exploit derivatives

- they transfer the mechanism of binary options from finance to information security. - they don't require to trade sensitive vulnerability information. and - the derivative is build around contracts that pays out a defined sum in case of security events.

consider a pair of contract (c and c'), where c pays a fixed amount of money, say 100 eur, if there exist a remote root exploit against some specified server software x on platform y at data d in the future.

the inverse contract c' pays out the same face value if there exist no remote root exploit submitted to a market authority before date d.

it is evident that the value of the bundle (c, c') is 100 eur at any time and that selling and buying the bundle is risk-free.

assume now that there is an exchange platform, where the contracts c and c' can be traded individually at prices determined by matching bid and ask orders. the platform settles the deals, and publish the price quotes from order book.

then the ratio of the market price of c and its face value approximately indicates the prob-

ability of software x being compromised before date d.

i argue that exploit derivatives have the promise to attract a large number of interest groups, including: - software users - cyber-insurance companies - investors - software-vendors - security experts and vulnerability researchers.

software users would demand C in order to hedge the risks they are exposed to due to their computer systems in place.

the same applies for cyber-insurance companies underwriting their custmers' cyber-risk.

investors would buy contracts c' to diversify their portfolios.

software vendors could be interested from several point of views. - they could demand contract c that pays if their software remains secure as a means to signal to their customers that they trust their own system. or contract C_comp that pay if their competitors' software get compromised - one could even think of software vendors using exploit derivatives as part of their compensation scheme to give developers incentives to secure programming.

security experts and vulnerability researches could use the market to capitalize efforts in security analyses. if, after a code review, they consider a software a secure (wrt to some family of vulnerabilities), they could buy contract c' at higher rate than the market price. otherwise they buy contract c and afterwards they follow their preferred vulnerability disclosure strategy.

i will not dig into this, but exploit derivatives might remind you about weather derivatives, that gained in the last years more and more popularity in the financial marketplaces.

weather derivatives were first traded by companies accustomed to trading contracts based on the electricity and gas prices in order to hedge their prices risk of their utilities. i don't

believe approaching the CME is today the most viable approach, without some preliminary research corroborating the value exploit deriviatives might create.

the modeling of contracts can be challenging, requiring financial players the knowledge of reliability growth models, and the understanding of the nature of software defects) furthermore, exploit derivatives - as proposed in literature - require a market authority (a kind of trusted third party) to test candidate exploit and publish them to provide verifiability and countering fraud.

how to bootstrap exploit derivatives then?

Bees and Crows

the beewise project tries to answer this question by framing exploit derivatives as an event-futures market, also known as a prediction market.

- the basic idea is to trade virtual stocks on an electronic market whose pay-offs are tied to the outcome of uncertain future events.

- although, the final pay-offs of stocks are unknown during the trading period, rational and risk neutral traders sell stocks if they consider the stockto be overvalued and buy stocks if they consider the stocks to be undervalued.

- as a result, the trading price reflects the traders' aggregated belief about the likelihood of the future event. market price can thus be interpreted as predictions

the theoretical foundation is provided by the hayek hypothesis: asymmetric dispersed information is best aggregated using a price mechanism

how does exploit derivatives contrast with Beewise prediction market?

differences range from the awarding mechanism to the exchange platform. yet, the first key difference is that BeeWise is based on bee dollars, a play money. it is natural to ask then: does money matter?

or more specifically: while the psychological importance of hard currency can't be understimated, and although a number of obstacles exist to the establishment of real-money prediction markets, does the play-money alternative compromise accuracy?

two real-word experiments answered in the negative. they contrasted the odds issued by real-money betting markets with the forecast made by play-money prediction markets, for the 2003-2004 NFL season and the FIFA 2006 world cup

play-money market were found to be about as accurate as betting markets

with real-money markets found to better motivate information discovery (ie, the incentive function) and play-money market to be more efficient in terms of information aggregation

still, a working prediction-market requires: - an heterogeneous user base - sufficient liquidity - low transaction costs

these are the challenges we are trying to address at beewise, where we support two market types:

- binary markets: where contracts pay the maximum settlmenent value to the holder, or nothing, depending on the occurrence of the underlying event by the event closing date

- index-based markets: where contracts pay a variable amount, depending on a pay-off function that takes the index as input

for example, a contract can be created that pays one dollar to the holder depending on whether the linux kernel is found to be vulnerable to a remote code execution by the end of the month

or also, a contract can be created that pays 1 BEE/USD cent for every data breach disclosed by a fortune 1000 in the next year

both the awarding mechanism and the contract naming scheme rely on the NIST Vulnerability DB and on CVSS, Common Vulnerability Scoring System. I will skip over the details. But if you're interested, please be in touch.

Before the concluding remarks, I would like to emphasize how combining information from more than one contract allow for interesting metrics

for instance, the spread – price difference between related contracts – can be directly attributed to distinctions in security (or public scrutiny) due to underlying technical differences

likewise, joint probability of failure can be computed from pairs of contracts to measure the total security of layered defense mechanisms – how much survivability we get with defense-in-depth?

the road ahead is still quite long and if you are interested i will be glad to be in touch a discuss further.

Conclusions

- what will be the future of software security? - will we find the way to balance the information between buyers and sellers in the security market? - which incentives will we manage to provide to vendors to build security in?

there ISN'T only one possible place for software security tomorrow. our society will live only the future it will choose to live into.

i believe the understanding of how humans behave, from an economic and psychological perspective, to be key to holistically address the security challenges and make the software the new foundation our society can rely on.

this is why when we talk about the future of software security we should correct ourselves and say futures - plural! this is why i believe security-event futures might contribute towards establishing information symmetry

thank you!