PhD Dissertation

International Doctorate School in Information and Communication Technologies

DISI - University of Trento

Risk-based vulnerability management

Exploiting the economic nature of the attacker to build sound and measurable vulnerability mitigation strategies

Luca Allodi

Advisor:
Prof. **Fabio Massacci**
Università degli Studi di Trento

Committee:
Prof. **Julian Williams**
University of Durham

Prof. **Radu Sion**
Stony Brook University

Prof. **Bruno Crispo**
Università degli Studi di Trento

April 2015

Pursuing this PhD work and writing this Thesis wouldn't have been possible without the priceless help and support of many around me including family, friends, and colleagues. Among many, I especially thank my father Adriano for his silent, but unconditional support; my mother Maria Giovanna for her very verbose, but still unconditional, support and encouragement; my brother Alessandro, because profound esteem goes well beyond academic milestones; my Кукуруза, because love motivates everything; and my PhD supervisor, Prof. Fabio Massacci, for his endless and patient scientific support and guidance. Finally, I want to thank this Thesis' committee for their helpful feedback and comments, that all contributed in making this work a better one.

Thank you all.

"The pride of youth is still upon you; late have you become young: but he who would become a child must surmount even his youth." [..] And there was spoken to me for the last time: "O Zarathustra, your fruits are ripe, but you are not ripe for your fruits!"

Friedrich Nietzsche.
Thus Spoke Zarathustra, Pt. 2 (22) The Stillest Hour.

Abstract

Vulnerability bulletins and feeds report hundreds of vulnerabilities a month that a system administrator or a Chief Information Officer working for an organisation has to take care of. Because of the load of work, vulnerability prioritisation is a must in any complex-enough organisation. Currently, the industry employs the Common Vulnerability Scoring System (CVSS in short) as a metric to prioritise vulnerability risk. However, the CVSS base score is a technical measure of severity, not of risk. By using a severity measure to estimate risk, current practices assume that every vulnerability is characterised by the same exploitation likelihood, and that vulnerability risk can be assessed through a technical analysis of the vulnerability.

In this Thesis we argue that this is not the case, and that the economic forces that drive the attacker are a key factor in understanding vulnerability risk. In particular, we argue that attacker's rationality and the economic infrastructure supporting cybercrime's activities play a major role in determining which vulnerabilities will the attackers massively exploit, and therefore which vulnerabilities will represent a (substantially higher than the rest) risk. Our ultimate goal is to show that 'risk-based' vulnerability management policies, as opposed to currently employed 'criticality-based' ones, are possible and can outperform current practices in terms of patching efficiency without losing in effectiveness (i.e. reduction of risk in the wild).

To this aim we perform an extensive data-collection work on vulnerabilities, proof-of-concept exploits, exploits traded in the cybercrime markets,

and exploits detected in the wild. We further collaborated with Symantec to collect actual records of attacks in the wild delivered against about 1M machines worldwide. A good part of our data-collection efforts has been also dedicated in infiltrating and analysing the cybercrime markets.

We used this data collection to evaluate two 'running hypotheses' underlying our main thesis: vulnerability risk is influenced by the attacker's rationality, and the underground markets are credible sources of risk that provide technically proficient attack tools, are mature and sound from an economic perspective. We then put this in practice and evaluate the effectiveness of criticality-based and risk-based vulnerability management policies (based on the aforementioned findings) in mitigating real attacks in the wild. We compare the policies in terms of the 'risk reduction' they entail, i.e. the gap between 'risk' addressed by the policy and residual risk. Our results show that risk-based policies entail a significantly higher risk reduction than criticality-based ones, and thwart the majority of risk in the wild by addressing only a small fraction of the patching work prescribed by current practices.

Keywords
Vulnerability Management, Attacker model, Attacker Economics

Contents

List of Tables

List of Figures

List of Publications

[1] Luca Allodi and Fabio Massacci. The work-averse attacker model. In *In the Proceedings of the European Conference on Information Systems (ECIS)*, 2015.

[2] Luca Allodi, Marco Corradin, and Fabio Massacci. Then and now: On the maturity of the cybercrime markets. *IEEE Transactions on Emerging Topics in Computing*, 2015.

[3] Luca Allodi. The heavy tails of vulnerability exploitation. In *Proceedings of the 2015 Engineering Secure Software and Systems Conference (ESSoS'15)*, 2015.

[4] Luca Allodi and Fabio Massacci. Tutorial: Effective security management: using case control studies to measure vulnerability risk. In *25th IEEE International Symposium on Software Reliability Engineering (ISSRE)*, 2014.

[5] Luca Allodi and Fabio Massacci. Comparing vulnerability severity and exploits using case-control studies. *ACM Transactions on Information and System Security*, 17(1):1:1–1:20, August 2014.

[6] Luca Allodi, Vadim Kotov, and Fabio Massacci. Malwarelab: Experimentation with cybercrime attack tools. In *Proceedings of the 2013 6th Workshop on Cybersecurity Security and Test*, 2013.

[7] Luca Allodi and Fabio Massacci. How cvss is dossing your patching policy (and wasting your money). BlackHat USA 2013 arXiv:1301.1275 [cs.CR], 2013.

[8] Luca Allodi, Shim Woohyun, and Fabio Massacci. Quantitative assessment of risk reduction with cybercrime black market monitoring. In *In Proceedings of the 2013 IEEE S&P International Workshop on Cyber Crime.*, 2013.

[9] Luca Allodi. Attacker economics for internet-scale vulnerability risk assessment. In *Presented as part of the 6th USENIX Workshop on Large-Scale Exploits and Emergent Threats*. USENIX, 2013.

[10] Luca Allodi and Fabio Massacci. Poster: Analysis of exploits in the wild. In *IEEE Symposium on Security & Privacy*, 2013.

[11] Woohyun Shim, Luca Allodi, and Fabio Massacci. Crime pays if you are only an average hacker. In *Proceeding of the 2012 IEEE ASE Cyber Security Conference*, 2012.

[12] Luca Allodi and Fabio Massacci. A preliminary analysis of vulnerability scores for attacks in wild. In *Proceedings of the 2012 ACM CCS Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*, 2012.

[13] Luca Allodi. The dark side of vulnerability exploitation: a research proposal. In *Proceedings of the 2012 Engineering Secure Software and Systems Conference Doctoral Symposium*, 2012.

Chapter 1

Introduction

The management of IT security is becoming a more and more prevalent challenge as system complexity increases. The evolving nature of IT systems further complicates the scenario: on the one side the increasing complexity of software often translates in more software flaws and vulnerabilities to fix [89], and on the other system threats continuously evolve, changing the risk outlook as new vulnerabilities and attack vectors emerge [58, 19]. For this reason, to measure the risk associated with a software vulnerability becomes a central point in any strategy for system security management. This is also reflected in the recent development, both in academia and industry, of software risk measures [83, 68, 124] and vulnerability management strategies [79, 40, 95] that are now adopted as a *standard-de-facto* worldwide [123].

However, the nature of the *risk* associated with these vulnerabilities remains largely unexplored. Risk is typically defined as the product of the impact or severity of an event, and its likelihood. While technical measures of vulnerability impact and exposure have been defined in the past [79, 68], a precise notion of likelihood of exploit remains to be found [30, 124]. On the other hand, this is crucial to a meaningful definition of vulnerability risk: attacks against two measurably similar vulnerabilities from a technical perspective (e.g. both allowing remote code execution via freed memory reuse)

are not necessarily similarly distributed in the wild. A meaningful risk estimation should indeed assign a higher risk score to the most frequently attacked vulnerability. Yet, this is not reflected in current practices and research [108, 68, 124, 79, 83]: current approaches focus mainly on a technical assessment of the exposure of the system to the vulnerability, and likelihood measures are often derived from the technical assessment itself [83, 30]. On the other hand, hackers' and cybercriminals' attitudes toward cyber attacks are known to go well beyond the mere technical matters: the attacker may be motivated by political or social reasons [115], as well as economic ones [58]. Attackers with different motivations and technological or infrastructural capabilities can be expected to generate attacks with different risk profiles both in terms of technical sophistication and distribution in the wild. This opens a set of interesting questions on the decision process of the attacker: how does the attacker choose which vulnerabilities to (massively) exploit? According to what process does the engineering of a new exploit translate into the final risk suffered by the user? It is not clear how current attacker models, often used to prove the security of a communication or cryptographic protocol [44], can be used to define the notion of vulnerability risk: attackers are usually thought of as very powerful (e.g. can access all systems and have complete information about the target) [2], but whether this is representative of the current status of cyber attacks remains an open issue [58, 73, 25].

In contrast, in this Thesis we develop the notion of the 'economic attacker' that is utility-oriented and work-averse (i.e. perceives work effort as a disutility), and that relies on a technological infrastructure for cyberattacks that he can access from the cybercrime markets [58]. We argue that the economic nature and capabilities of an attacker are an important driver for technological and operational risk. In particular, in this Thesis we show that vulnerability risk is largely influenced by the attacker's rationality in deciding which vulnerabilities to exploit, and by the economic environment

the attacker operates in. By accounting for these factors, we define a novel attacker model that, when factored in the risk assessment, allows us to identify vulnerability patching strategies that are significantly more efficient than current best practices.

The remainder of this Chapter unfolds as follows: in Section 1.1 we give a more detailed introduction on current practices for vulnerability management and we outline the inefficiency problem that they entail. In Section 1.2 we define our research problem, and in Section 1.3 we outline the main contributions of this Thesis work. Finally, Section 1.4 presents an outline of this manuscript's organisation.

1.1 Risk Management and the Inefficiency Problem

When it comes to risk mitigation best practices, stating a rule that defines what represents 'unacceptable risk' is probably the most immediate approach. A 'rule' usually sets a critical threshold over some technical dimension [78]. The chosen technical dimension(s) correspond to a *point estimate* of some expected property of the component. The underlying assumption here is that the considered point estimate has a certain descriptive power relative to the distribution. By setting a rule that covers a wide fraction of the probability distribution of 'bad events', one hopes to achieve almost full coverage against possible hazards. However, in computer security this 'point estimate' is difficult to obtain given the wide diversity of systems and technologies involved in the assessment, and the disparate nature and resources available to developers, system administrators, attackers, and system stakeholders [78].

A clear example of this problem emerges from an overview of how vulnerability management currently works: organizations that want a security clearance to operate in certain fields (e.g. in the financial sector) or that simply need guidance to prevent and mitigate security incidents are obliged to

comply to security standards (e.g. PCI-DSS for credit card security [127]) or protocols and best practices (e.g. the NIST SCAP Protocol [95]) to manage the security of their IT systems. These standards and best practices prescribe a 'criticality-based' vulnerability management i.e. based on a measure of how *technically severe* the vulnerability is. We define criticality-based policies in the following way:

Definition. *Criticality-based policies for vulnerability management define a critical level of the technical measure of a vulnerability above which patching is required.*

Being a technical measure, the defined 'rule' is to be applied equally regardless of the organisation's security needs and resources. While this 'technical assessment' has the advantage of being easily manageable by the issuing institution of the certification (as it does not change among organisations), the organisation may suffer from substantial inefficiencies in implementing the rule as prescribed: Is the rule actually fit to the threat types the organisation faces? How can the organisation measure how effective and apt the 'rule' is for them? Can the organisation do any better while remaining within the limits for compliance? Unfortunately, a technical measure is not suitable to answer any of these questions because it can not reflect, by definition, other elements that are proper to the organisation and its specific threat model and operative environment. In other words, organisations are left operating over their vulnerabilities without a way to estimate the risk they are subject to and to evaluate which mitigation strategy works better in their context.

This is particularly undesirable as vulnerability management can be very expensive and risky from a business continuity perspective: in today's highly connected and diverse operative environments, it is difficult to foresee what effects a change upstream may have down the network. For this reason extensive testing is often needed before deploying a patch over a system

(e.g. providing a service) or set of systems (e.g. interfacing with the service). With hundreds of vulnerabilities to manage per year [122, 114], this operation can become very expensive and fraught with organisational problems: which vulnerability(-ies) should the organisation start from? What is the actual return in terms of *additional security* gained from the investment? Is it worth the time and the money it requires? This effect is clearly visible in the recent 2015 Verizon report on PCI Compliance, where the vulnerability management and testing requirements (i.e. requirements 5,6 and 11) are among the least met by companies [123]. It is therefore clear that *vulnerability prioritisation* becomes central to any vulnerability management process. This is in turn representative of a more general issue, that is 'to measure' how better off the organisation is if a certain mitigation action is taken sooner than another. Yet, without a characterisation of 'vulnerability risk' it is currently not clear how to obtain this measure.

Every vulnerability management product available on the market (providing also tools supporting compliance to a number of standards) is essentially based on a 'red-yellow-green' assessment of vulnerability severity: a simple computation of the number of vulnerabilities present on the system and their technical severity. This approach is also employed by the scientific literature [108, 83, 30, 103].

The main problem with this 'criticality-based' approach is that it implicitly assumes that a vulnerability's technical severity level can be considered a proxy for vulnerability risk. Whilst it is certainly true that a critical vulnerability will sooner or later need to be fixed, it is not necessarily true that less critical vulnerabilities will pose a lower immediate risk. Even within the same 'criticality level' different vulnerabilities may pose different risk (e.g. because of some known and publicly available proof-of-concept exploit). In this logic, to immediately fix 'higher criticality' vulnerabilities may cause 'high risk' vulnerabilities to remain untouched longer than necessary, while

the workload remains bloated with unnecessary work over severe but low-risk vulnerabilities. This can be clearly very inefficient and, possibly more importantly, will *not* necessarily benefit the overall security profile of the organisation - if not worsen it as more resources are put in fixing low-risk vulnerabilities rather than in other mitigation actions.

1.2 Research Problem

The inefficiency issue outlined above opens a series of challenges to the community on how to *measure* how better off an organisation's overall security is after a mitigating action has been taken. The following excerpt is taken from a recent report by the Ponemon institute [92]:

> *The majority of security professionals [..] aren't sure how to distil this information [on security risk] into metrics that are understandable, relevant and actionable to senior business leadership. [..] Finding meaningful ways to successfully bridge this communication gap is critical to broader adoption of risk-based security programs. .*

Indeed, one can use metrics such as *attack surfaces* [68] to estimate the overall exposure to potential security threats, but can not obtain an estimate in terms of *diminished risk* to communicate to the business' decision maker or to employ to engineer a better security plan. Being able to *measure* vulnerability risk can also be beneficial when communicating with auditors for compliance, that have to verify the soundness of the implementation of security requirements for the standard certification. Currently, to justify an unmet requirement the organisation has to produce lengthy (and expensive) documentation justifying the decision in relation to the organisation's infrastructure and existing countermeasures [127]. With a sound measure for

risk the lengthy and expensive documentation could be ideally synthesised as follows: '*I haven't yet fully pursued this requirement because its fulfilment entails for me only a 1% reduction in risk, which is negligible when compared to the 90% reduction of this other mitigation action.*'

In order to make such statements possible, one has to shift from a purely technical decision model (i.e. current criticality-based policies) to a *risk-driven* one whereby impact and exploitation likelihood are both accounted for. We define *risk-based vulnerability management policies* as follows:

Definition. *Risk-based policies for vulnerability management define a measure for vulnerability risk based on vulnerability severity and likelihood of exploitation.*

Our research goal is therefore to show that risk-based vulnerability management policies are possible, and that the economic nature of the attacker and his/her rationality are determinant factors in designing more effective vulnerability management practices.

1.3 Thesis Contribution

The principal contribution of this thesis is that we demonstrate that vulnerability risk hugely varies among vulnerabilities, and that the rational and economic nature of the attacker and of the environment he/she operates in are of major importance in creating this gap. To demonstrate that this is the case, in this Thesis we:

1. Present a unique set of datasets comprising vulnerabilities, exploits, exploits traded in the black markets, attacks in the wild, and data on black market operations. The collection of these datasets required a full year of ethnographic research (to identify and infiltrate the cybercrime markets) and planning to meet the requirements needed to have access to

real attack data provided by Symantec. This data is used orthogonally to validate each claim and conclusion made in this Thesis.

2. Show that the attacker is rational in choosing which exploits to engineer and massively deploys in the wild, and that this generates a skewed distribution of risk for the final user.

3. The economic activities of the attacker operating in the underground markets characterise a foremost source of risk for the final user. We demonstrate that these markets are economically and technologically sound and conclude that they are not a temporary phenomenon.

4. The attacker's rational nature and economic environment can be exploited to design better vulnerability management strategies based on the notion of vulnerability risk. These strategies offer great advantages in terms of patching efficiency over current best practices.

1.4 Thesis Outline

This Thesis unfolds as follows. In the next Chapter we outline the objectives of this Thesis and provide a detailed discussion of the methods employed for hypothesis testing. Chapter 3 frames the problem this Thesis addresses by discussing related works on vulnerabilities, exploits and attackers, and by identifying open problems currently not addressed in the literature. The discussion then moves to introducing our datasets, with a focus on the data collection methodology (Chapter 4). A high-level overview of our data is given in Chapter 5. The core of this dissertation unfolds in Chapter 6, where we discuss and test attacker rationality and economics as an enabler for risk-based vulnerability management. Chapter 7 tests the effectiveness of risk-based policies and evaluates their advantages over criticality-based ones.

Finally, Chapter 8 and Chapter 9 conclude this dissertation by discussing limitations and future research venues and conclusions respectively.

Chapter 2

Reserach Objectives and Methods

2.1 Are Risk-based Policies Possible?

The current practice on vulnerability management is based on the conservative notion that, if a vulnerability is there, sooner or later an attacker will exploit it. This is an inheritance from more traditional aspects of security, such as cryptography, where the existence of one flaw in the protocol is enough to invalidate it [44]. For example, Bruce Schneier famously stated in 2005 that *"Security is only as strong as the weakest link"* [2]. Similarly, Williams and Chuvakin, domain experts for PCI-DSS compliance (the standard for credit card management security), state *"Don't spend a huge amount of time and effort prioritizing [vulnerability] risks, since in the end they all need to be fixed"* [127]. Somewhat ironically, Chuvakin himself will later acknowledge the importance of the risk prioritisation problem [7]. Still, the general consensus is that if a vulnerability is there and is technically critical, it must be fixed with high priority.

The implicit assumption here is that all vulnerabilities of the same criticality entail the same risk level, i.e. that attacks are uniformly distributed over similar vulnerabilities. In this scenario, a criticality-based policy stating a criticality level for mandatory patching is a good solution and one that can be hardly improved: because all vulnerabilities are equally likely to be

ultimately exploited, removing only one vulnerability would leave the system at the same level of risk, irrespective of which vulnerability is fixed.

Yet, this may not be the case in practice. In recent years, the figure of the attacker moved from the 'curious hacker' or 'script-kiddie' to the 'organised cyber criminal' that can rely on a pre-existent organisational and technological infrastructure to deliver attacks. The main consequence of this evolution is that attacks are nowadays 'commoditized' [58] through underground markets where the technology producers *sell* the exploitation technology to a multitude of buyers that are *users of the technology*. Therefore, the attacker tends now to be a *rational economic actor* operating in a market.

The main intuition in this direction is that the rational attacker's level of interest in attacking a vulnerability should be a function of the expected 'return-on-investment' from the exploitation. We think of the vulnerability exploitation process as a two-phase process whereby the exploit first needs to be engineered, and then either deployed in the wild or sold to other attackers operating in the cybercrime markets [4, 58]. We make two key observations on this regard:

1. **Engineering phase**: Vulnerabilities get fixed in 'chunks' by the vendor with the release of a new software version [38]. Each software version often addresses tens of vulnerabilities. The attacker has therefore, for each software version, tens of vulnerabilities to potentially exploit. Yet, because a software version is vulnerable to *all* these vulnerabilities, the rational attacker will only need to exploit (a sufficiently powerful) one: exploiting two or more vulnerabilities will not increase the number of successful attacks that can be launched, because all users of that software version are equally vulnerable until the next upgrade, when *no user* will be vulnerable to *any* of those vulnerabilities. It makes therefore no economic sense for the attacker to exploit more than one vulnerability per software version. For this same reason, the attacker that aims at

a *mass exploitation* of final users will need to engineer a new exploit only when a sufficiently high number of users will have switched to a new software version. We therefore hypothesise that few vulnerabilities are *high return* vulnerabilities, and that therefore only few vulnerabilities will be massively exploited by the attacker, generating a skewed distribution in risk for the final user.

2. **Commercialisation phase**: Vulnerability exploits are reportedly traded in the underground black markets [4, 58]. Because these markets enable *mass exploitation* of final users [58], we argue that these exploits are engineered following the rationale described above. As in any market, there is a *1:n* distribution rate of technology, i.e. *one* vendor sells a technological solution to n users of the technology. In a criminal market, this translates in *one vulnerability exploit* being used to massively generate attacks by the n buyers of that exploit. We therefore hypothesise that the cybercrime underground markets can represent a significant *multiplier factor* in the final risk for the user.

A 'risk-based' approach to vulnerability management seems therefore more sensible than the classic criticality-based approach whereby all similar vulnerabilities represent equal risk. Importantly, in this scenario a criticality-based approach to vulnerability mitigation may be largely suboptimal as it may require to address a number of vulnerabilities that could otherwise be safely ignored or postponed in the patching schedule. This defines the main thesis of this dissertation:

Thesis. *Risk-based vulnerability management policies are possible and can significantly improve the efficiency of current vulnerability mitigation practices.*

The discussion above outlines two key enabler factors to risk-based vul-

nerability management policies: attacker rationality, and functioning (and stable) cybercrime markets. Both these conditions need to be verified before proceeding with testing our Thesis: were the attackers not rational, or the markets not sound, any measured effect of risk-based policies may be a temporary (or casual) one. We formulate three running hypotheses, that are presented in the remainder of this Chapter alongside the relative testing methodology. Table 2.1 provides a birds-eye view of this setting.

2.2 The attacker is rational and work-averse

Our first hypothesis aims at establishing that the attacker acts rationally. Rationality is a widely-accepted underlying assumption in the broad fields of economics and information security economics [118, 32], whereby economic actors are driven by a utility maximization function, i.e. each actor tries to maximise his/her own gain from the execution of certain actions.

In the case of the cyber attacker, his/her goal is to maximise the return from the execution of an attack. Because finding and exploiting vulnerabilities is a time-consuming and therefore costly process [81], the rational attacker will choose to exploit a vulnerability only if this represents a high enough gain in terms of increased attack capability with respect to his/her current capabilities. In other words, the attacker will develop a new exploit only if the expected returns from the exploitation of the new vulnerability are lower than the cost of developing and deploying the new attack.

In particular we observe that the exploitation of multiple vulnerabilities does not necessarily imply a more ample pool of potential victims for the attacker. To contain testing, deployment and costumer support costs, software vendors patch vulnerabilities in bulks [38] by releasing a new *software version*. Therefore, to attack a certain software version j the attacker can choose among n vulnerabilities $v_{j,i} \in N_j$, with N_j the set of vulnerabilities

affecting version j and its cardinality n often much greater than 1.

The by-product of this process is that every user that is vulnerable to a certain vulnerability $v_{j,i}$ is also vulnerable to the remaining $n-1$ vulnerabilities for that software version. In this scenario, the attacker that aims at attacking that set of vulnerable users can do so by exploiting one vulnerability only of the available n[1]. As a consequence, for each software version the rational attacker will tend to exploit at most one vulnerability, and leave $n-1$ vulnerabilities unexploited. Extending this to the overall picture, we formulate the following hypothesis on attacker's rationality:

Hypothesis 1 *The attacker ignores most vulnerabilities and massively deploys exploits for a subset only.*

If Hyp 1 holds attackers' rationality implies that attacks are not uniformly distributed among vulnerabilities. A criticality-based approach to vulnerability management may therefore be not optimal.

Hypothesis testing

To test Hypothesis 1, we identify two constraints that the attacker has to respect to be 'work-averse'. Our first observation is that the work-averse attacker needs to exploit only one vulnerability per software version, as exploiting more would not result in an increased volume of final infections. This is because the user of a certain software version will be equally vulnerable to all vulnerabilities affecting that version. If the overall picture of attacks in the wild does not respect this constraint, than we can not conclude that the attacker acts rationally as a work-averse actor. We therefore hypothesise the following:

[1]Clearly, not all these n vulnerabilities are necessarily technically comparable. Some vulnerabilities (e.g. Cross-Site-Scripting vulnerabilities) are less powerful than others (e.g. Buffer Overflows). Similarly, the exploitation of different vulnerabilities may carry different costs for the attacker (for example, some countermeasures deployed at the system level make memory exploitation harder).

Hypothesis 1a *The attacker will massively use only one exploit per software version.*

Then, because patching rates on the side of the user are often slow [69], we expect the work-averse attacker to wait a considerable amount of time before massively deploying a new exploit, as an old one should provide a satisfactory level of infections. If not, then again the attacker would arguably be doing *more* work than what optimally prescribed by his/her rationality.

Hypothesis 1b *The fraction of attacks driven by a particular vulnerability will decrease slowly in time.*

Update rates and software types. From Hyp. 1a and 1b we argue that the average user behaviour in updating a system determines the rate at which the efficacy of an exploit declines. However, not all software is updated at the same pace both on the vendor side (that is slower in developing the patches ([105]) and the users' side (that may be more likely to apply available patches for a software type than for another ([69])). Lately, some software (e.g. internet browsers) started adopting a 'quick development cycle' ([86]) that quickly patches vulnerabilities and sends automatic updates to the users. The attacker behaviour may change with respect to the software type. For example, users may seldom update their Java plugin, whereas they run the latest version of the Internet Explorer browser.

Corollary to Hyp. 1b The attacker waits a longer period of time to introduce an exploit for software types under a slow update cycle than for others.

This corollary will serve as a robustness check to the Hypotheses above, as their acceptance would be incoherent with the rejection of this Corollary.

2.3 The underground is a sustainable market economy

Our second hypothesis investigates the economic sustainability of cybercrime markets. The typical agency problems any market has to address [48] are, in the cybercrime markets case, particularly prominent: the criminal, and largely anonymous and virtual nature of these markets make contract completeness and enforcement hard to achieve. Market operation can be difficult in these conditions. Identifying bad agents and disincentivize unfair behaviour (e.g. in terms of moral hazard) become in this setting central mechanisms of a functioning market [56]. These mechanisms have however been shown to be at best poorly addressed in the cybercrime IRC-based markets [63], where information asymmetry problems effectively push all 'good agents' out of market. On the contrary, we argue that current forum-based cybercrime markets [130, 82] can enforce mechanisms that are effective in mitigating or solving these issues. We formulate the following Hypothesis:

Hypothesis 2 *The underground markets are sound from an economic perspective.*

If Hyp. 2 holds, we conclude that the markets are *not* a transient source of risk for the final user and are therefore *key and permanent enablers* of the 'risk-based' approach to vulnerability mitigation we propose.

As anticipated, a most prominent issue in a market for criminals is the *agency problem*, whereby a principal commissions a work to an agent via an *enforceable* contract. The setting of a criminal virtual community is particularly interesting in this respect as market participants can only stipulate incomplete contracts, as contract enforcement can not be guaranteed by a controlling authority. Moreover, a buyer interested in a good has access to only a limited amount of information to decide whether a particular attack technology fits his needs, or simply if this technology works. Market participants operate therefore in a *bounded rationality* setting where uncertainties

on the trustworthiness of the seller and the quality of the traded good need be addressed in order for the market to be sustainable.

We therefore formulate two propositions following Hypothesis 2, addressing respectively the existence of market mechanisms to mitigate trade uncertainties, and the overall quality of the traded goods. Finally, to test Hypothesis 2 we develop a two-stage model of the underground markets were we formally show that the mechanisms tested under proposition 1, and the product quality shown under proposition 2 allow for a *sustainable cybercrime market environment* that encourages fair trading and discourages scammers from participating.

2.3.1 Proposition 1: The underground markets are mature

Recent literature reports how attackers are now *en-masse* operating in underground markets. This may represent a *multiplicative factor* for vulnerability risk as the same exploit may be distributed to multiple attackers. This opens the question whether these markets are really functioning, or are just a transient phenomenon. If this is not the case, then this multiplicative effect would be a permanent factor favouring risk-based policies over criticality-based policies.

Market design is a problem of great interest in economics, as a successful market necessarily involves an equilibrium of forces that on one side encourages 'traders', and on the other discourages "cheaters". In particular, a market where everybody cheats is not a sustainable market and is doomed to fail because nobody would eventually initiate a trade (or, equivalently, all sellers will eventually exit). Cybercrime markets represent therefore a fascinating case study: they are run by criminals (who are not trustworthy by definition), are typically run on-line, and are to a degree anonymous. How can anonymous criminals trust other anonymous criminals in delivering the promised service or good after the payment has been issued? And even if

the buyer gets 'something', how can she be sure that what she thinks she is buying is effectively what she will end up with? If a trade goes sour, a buyer cannot call the police to apprehend the scammer.

Florencio et al. [63] showed that IRC cybercrime markets (Markets run through Internet Relay Chats) may be no different from the notorious *market for lemons* captured by Akerlof [13], where effectively the *asymmetry of information* between the seller and the buyer is such that "bad sellers" are incentivized in participating in the market to the point that it makes no sense for the "good sellers" to remain active. In Akerlof's case, a "bad seller" is a seller that trades 'lemons' (a defective car that is advertised as a good one). If the customer can not assess the quality of the car before buying it (e.g. because she knows little about cars), then she will buy the cheapest she can find on the market. Since 'lemons' are cheaper than good cars, 'good sellers' are ultimately forced out of the market. In Florencio et al.'s case, a 'lemon' was a credit card number with (allegedly) a certain amount of money ready to be used by the buyer. As shown in Akerlof's work, discerning 'good sellers' from 'bad sellers' is therefore a critical point of a market design. Florencio et al. clearly demonstrated that it is virtually impossible to do so in the IRC cybercrime markets. On the other hand, recent reports show that cybercrime tools and infrastructure seem to work [112, 58]. Following these observations, we formulate the following Proposition:

Proposition 1 *The underground markets evolved from a scam-for-scammer model to a mature state whereby fair trade is possible and incentivised by the enforced trading mechanisms.*

If Prop. 1 holds, we conclude that the underground markets can be a sustainable operating environment for the rational attacker. The 'multiplier effect' in attack volume, enabled by marketed vulnerabilities (as opposed to little known ones), will make technically identical vulnerabilities different in

terms of final risk to the user depending on whether they are traded in the markets or not.

Testing Proposition 1 Forum markets In order to understand whether cybercrime markets evolved to a mature state, we compare two forum underground markets: one that failed and one that is still active. We label these markets Carders.de and HackMarket.ru. Carders.de (which failed) specialized mostly in credit cards, while HackMarket.ru (still active) specializes mostly in cyber-crime tools, albeit some transactions are also about monetary goods (e.g. credentials for Skype accounts). We give a more precise description of both markets in Chapter 4.

Both Carders.de and HackMarket.ru are forum-based markets. They have administrators, moderators, users' registration procedures, reputation mechanisms and so on. The major difference with Alibaba, eBay, or Craiglist is that they mostly advertise 'illegal' goods.

At first, notice that even legitimated forum markets are rife with scams. After 20 years since eBay's foundation, many frauds reported by FBI's 2013 Internet Crime Reports [50] rely on legitimate forum markets to perform scams: good old lemons are advertised and sold via eBay [50, pag. 8]; bogus real estates are sold via Craiglist; failed delivery or payment of goods are common places; etc.

To create 'safe trading places' where only experienced and trustworthy users participate, forum-based markets have created a number of mechanisms aimed at distinguishing 'good' and 'bad' users. A system to effectively manage reputation is a key issue in the trust of an on-line market place. For example, eBay filed its own reputation based mechanisms for patenting in 2000 [97] and at the beginning of 2015 has almost 200 patents listed on Google's patent with the keyword 'user reputation".

The forum mechanisms in legal on-line markets have provided a 'satisfy-

ing', in the sense of Simon [109], protection to legitimate users to make those markets thrive. For example, Melnik and Alm showed that reputation does matter in sales [80]; Resnick and Zeckhauser showed that buyers and sellers actively and deliberatively provide positive or negative ratings, with positive ratings being the majority [98].

From a legal perspective, reputation mechanisms only provide partial coverage. Law scholars have discussed the issue at length (see e.g. [33, 14] for some of the earliest papers). However, if the reputation mechanism fails, and a 'lemon' is sold via eBay, a customer can always resort to the FBI Internet Crime Center which will pass the complain to the local prosecutor [50, pag. 18]. Similar protections are available to customers in other countries. Such last resort is not available to victims of trades gone sour in criminal forums.

Therefore, illegal markets must either make the reputation mechanism more robust or compensate for the failure of the mechanism with prosecution procedures. Absence or failure of these additional enforcement mechanisms would intuitively re-create the same conditions that Florêncio et al. [63] identified for the IRC markets: information asymmetry would favour 'ripping' behaviour and eventually bring the market to fail.

We formulate a number of hypotheses from the description of Carders.de's and HackMarket.ru's regulatory mechanisms (reputation being just one of them). The goal is to compare the two markets on the same regulatory ground and see if newer and still active markets solved the regulatory problems present in the failed ones.

Effectiveness of reputation mechanism If the reputation mechanism works, known scammers should have the lowest reputation among all user.

Proposition 1a *Banned users have on average lower reputation than normal users.*

If Proposition 1a is true, it is evidence that the regulatory mechanism for reputation is effectively enforced, and provides to forum users an instrument to evaluate traders' historical trustworthiness. If the data does not support this, "reputation" in the forum is not a good *ex-ante* indicator of a users' trustworthiness.

Fora may present a hierarchy of roles or status groups that each user can 'escalate' to. In a functioning system the status should be reflected in the reputation rating.

Proposition 1b *Users with a higher status should on average have a higher reputation than lower status users.*

If Hypotheses 1a and 1b do not hold, it may as well be because moderators left a part of the market to its own and concentrated all regulatory efforts on the higher market tiers. For example, in the Carders.de market, there are three Tiers of traders and the first Tier may just represent noise in the data.

To check this possibility, we can restrict Hyp1a to hold only for users that are higher in the hierarchy.

Proposition 1c *Banned users who happened to have a higher status have a lower reputation than other users with the same status.*

If even Hyp. 1c does not hold, we conclude that the reputation mechanisms even after controlling for market alleged 'status' provide no meaningful way for the forum users to distinguish between "bad traders" and "good traders".

Enforcement of rules Reputation may fail to provide effective information, but the hard-wired categories of the forum users (the ones under the direct control of the administrators) may provide a better indicator of quality. Normally, access to the higher market tiers should be subject to some rules. The market is reliable if such rules are consistently enforced.

To see whether this regulation is enforced we can test the following Proposition:

Proposition 1d *The ex-ante rules for assigning a user to a category are enforced.*

Once transactions fail, Carders.de and HackMarket.ru users cannot turn to legitimate law enforcement agencies for a redress. Therefore, the forum must have some alternative rules to manage trades gone sour.

Proposition 1e *There are ex-post rules for enforcing trades contemplating compensation or banning violators.*

Market existence An obvious, but important question to ask is whether the market actually exists. In other words, whether actual transactions take place (*took* place for Carders.de). Indeed, the role of the forum boards is to provide a platform for sellers and buyers to advertise their merchandise. The actual finalization of the trade usually happens through the exchange of *private messages* between the trading parties [52, 63].

Proposition 1f *Users finalize their contracts in the private messages market.*

If Hyp 1f holds, than the exchange of private messages would be a good proxy for us to measure the successfulness of 'normal' users and 'rippers' in closing trades. To check whether 'normal users' are significantly more successful than 'rippers' we test the following Proposition:

Proposition 1g *Normal users receive more trade offers than known rippers do.*

For Carders.de, where we have access to the whole forum, a suitable proxy is counting the number of times a forum user initiates a trade with another

forum user i.e. the number of *unsolicited incoming private messages* a user receives. The proportion of private messages that are trade-initiation can be calculated to answer the previous Proposition. For HackMarket.ru such analysis must be qualitative as downloading the whole forum would reveal our presence.

We would expect the results for Hyp. 1g to be coherent with the results obtained so far for the forum. In other words, if the reputation mechanism works, the tier system is properly enforced, and the exchange of private messages is used to conclude the trading process, then we would expect normal users to conclude more trades than rippers do. This is because the consistent enforcement of the forum rules would give market participants an instrument to discern rippers from normal users. Otherwise, if the evidence gathered so far suggests a systematic failure in the market regulation, then we would expect rippers to be indistinguishable from normal users because the user cannot do better than randomly picking a seller from the whole population.

2.3.2 Proposition 2: The technology traded in the underground is effective

Besides a mature economic setting to operate upon, a successful market needs goods to be exchanged. Traded goods can be of any nature, but for the economy to be sustainable the goods have to deliver the advertised functionality (or buyers will simply stop buying products). From the perspective of a cybercriminal operating in the black markets, the good must deliver the attack as promised by the vendor.

Recent industry reports [122, 113] and scientific studies [58] reported on the attack capacity of the infrastructure provided by the underground markets; some studies estimate the fraction of attacks that can be traced back to attack tools traded in the underground [93], but no study empirically and ex-

plicitly evaluates their effectiveness. We aim at filling this gap by formulating and testing the following Proposition:

Proposition 2 *The tools bought and used by the attackers are well engineered products that are effective when deployed in the wild.*

If we find evidence supporting Hyp. 2 we conclude that the cybercrime markets distribute effective attack technology to multiple attackers that ultimately deploy those attacks.

Testing Proposition 2 We will directly test for Proposition 2 by testing the effectiveness and resiliency of tools traded in the cybercrime markets against evolving system configurations. These test are run in a laboratory built for this purposes at the University of Trento, the *MalwareLab*.

2.4 Risk-based Policies are Possible

Hypotheses 1 and 2 postulate the feasibility of risk-based policies. In particular, Hyp. 1 postulates that 'high return' vulnerabilities will carry higher risk for the final user than most vulnerabilities. Hyp. 2 postulates that the underground markets act (and will keep on acting) as 'risk amplifiers'. We therefore formulate the following concluding hypothesis:

Hypothesis 3 *It is possible to construct risk-based policies that, levereging the economic nature of the attacker, can greatly improve over criticality-based policies.*

In particular, due to the multiplicative effect we predict from Hyp. 2, we expect risk-based policies that account for the presence of a vulnerability in the black markets to be the most effective ones. We therefore formulate the following corollary to Hyp. 3:

Corollary to Hyp. 3 *Risk-based policies accounting for cybercrime markets are the most effective in reducing risk for the final user.*

Hypothesis testing

We evaluate the effectiveness of risk-based policies as opposed to that of criticality-based policies by developing a case control study accounting for vulnerabilities, exploits in the wild, and cybercrime activities and actors (as established with Hypotheses 1-2). In particular, we evaluate policy effectiveness by measuring the *risk reduction* it entails: risk reduction is a relative measure of the leftover risk after a certain patching decision is taken. To accept Hypothesis 3, we further provide an application example whereby we compare workloads and benefits in terms of foiled attacks in the wild of risk-based and criticality based policies.

Table 2.1 summarises this Section's discussion.

Table 2.1: Summary of running hypotheses and hypothesis testing in this Thesis.

Running Hypothesis	Hypotheses Testing
Hyp. 1. The attacker ignores most vulnerabilities and massively deploys exploits for a subset only.	**Hyp. 1a**. The attacker will massively use only one exploit per software version. **Hyp. 1b**. The fraction of attacks driven by a particular vulnerability will decrease slowly in time. **Corollary to Hyp. 1b**. The attacker waits a longer period of time to introduce an exploit for software types under a slow update cycle than for others.
Hyp. 2. The underground markets are sound from an economic perspective.	**Prop. 1**. The underground markets evolved from a scam-for-scammer model to a mature state whereby fair trade is possible and incentivised by the enforced trading mechanisms. • **Prop. 1a**. Banned users have on average lower reputation than normal users. • **Prop. 1b**. Users with a higher status should on average have a higher reputation than lower status users. • **Prop. 1c**. Banned users who happened to have a higher status have a lower reputation than other users with the same status. • **Prop. 1d**. The ex-ante rules for assigning a user to a category are enforced. • **Prop. 1e**. There are ex-post rules for enforcing trades contemplating compensation or banning violators. • **Prop. 1f**. Users finalize their contracts in the private messages market. • **Prop. 1g**. Normal users receive more trade offers than known rippers do. **Prop. 2**. The tools bought and used by the attackers are well engineered products that are effective when deployed in the wild, as tested in the *MalwareLab* against evolving software configurations. **Hyp. 2**. Develop a two-stage model of the underground markets to show that the underlying economic mechanism is sound.
Hyp. 3. It is possible to construct risk-based policies that, levereging the economic nature of the attacker, can greatly improve over criticality-based policies.	**Corollary to Hyp. 3** *Risk-based policies accounting for cybercrime markets are the most effective in reducing risk for the final user.* Develop a case control study to evaluate the overall risk-reduction of risk based and criticality based vulnerability management policies. A validating example outlines the benefits of risk-based policies over criticality based ones in terms of patching workloads and effectiveness in foiling real attacks in the wild.

2.5 Research methodology and scope of work

This thesis' contribution is grounded on empirical research. Empirical research methodologies are usually divided in two main categories: qualitative and quantitative research methodologies [128].

- Qualitative research aims at studying the phenomenon of interest in its natural setting, usually in order to understand *why* something happens rather than trying to assess *how* or *how frequently* does it happen.
- Quantitative research aims at *measuring* some quantity of interest [74]. The goal is usually to compare these measures among groups that the researcher can control (as in an experiment) or observe and control *a posteriori* (as in a case control study) in order to evaluate a certain hypothesis of interest.

In this Thesis we employ both approaches. In particular, we employ a *case study* to (qualitatively) study the cybercrime markets, and a *case control study* to (quantitatively) study vulnerability risk.

- Case studies are concerned with understanding one particular setting of interest over well-specified dimensions [74]. Case studies are often used for *exploratory* and *descriptive* purposes [99], whereby the researcher aims at both deriving a 'big picture' perspective over the phenomenon of interest, and at deriving the fundamental 'building blocks' necessary to describe it. If the case is general enough, or it fits exactly the boundaries of the research (i.e. is representative of the analysed problem), a case study can also be employed for *explanatory* purposes [102]. From an exploratory and descriptive analysis is also possible to derive *models* for the analysis that use the qualitative results of the study to build and validate a model of the phenomenon of interest. Our case study

is focused on one particularly active underground market that features, among its participants, the main cybercrime players and products often cited in the media [8, 87] and the literature [58, 75].

- A case control study is typically run over *field data*, i.e. data collected through some pre-existent collection mechanism, or through interviews [41]. A case control study looks at existing data to derive, through the implementation of proper controls, conclusions on the *correlation* between an observation and an 'explanatory variable' (i.e. a certain hypothesis on *why* an effect can be measured in the data). Case control studies have notably been employed to initially link smoking and carcinoma of the lung [45], and use of seat belts and likelihood of death in a car accident [49]. As exemplified by these two examples, a case control study is typically run when an experiment can not be run for practical or ethical reasons: one can not randomly assign patients to a twenty-year smoking period, and measure whether they get cancer down the line. Similarly, we can not ask participants to stay vulnerable and then measure who gets their bank accounts emptied. We therefore rely on field data collected by Symantec and use a case control study to derive our conclusions.

Data gathering. The initial part of this work has been dedicated entirely to gather data on vulnerabilities, exploits, and black markets. In particular, we collected data from public datasets such as the National Vulnerability Database (NVD) for the 'universe of vulnerabilities' and the Exploit Database (Exploit-db) for proof-of-concept exploits (i.e. exploits that demonstrate the exploitability of a vulnerability). We further collected three additional datasets that are not fully or directly available in the public sphere. EKITS is a dataset reporting vulnerabilities traded in the black markets. It is built over Contagio's Exploit Pack table [11], that we however substantially

expanded by integrating it with data on more than 90 Exploit kits and 100 unique vulnerabilities for a total of about 900 records. SYM and WINE are a collection of exploited vulnerabilities (SYM) and records of attacks against vulnerabilities (WINE) reported by Symantec through their Worldwide Intelligence Network Environment Data Sharing Programme [46]. WINE is available to use for researchers pursuing projects selected by Symantec.

As per the cybercrime markets, we collected data on two case studies: one for a failed underground market, whose database eventually leaked through underground channels, and a second for an active market, that we infiltrated. These two case studies allow us to perform two analyses:

1. By comparing the two markets over a set of hypotheses on the effectiveness of their regulatory mechanisms, we can highlight the differences between an old and failed market and a new and active one. We perform this analysis through a mixture of quantitative and qualitative analysis of the two markets.

2. By thoroughly analysing the active market we first describe its trade operations, and how issues such as information asymmetry [13] (typical of any principal-agent problem were contracts are incomplete [48]) are addressed. Based on this analysis, we build a model of the underground market activities and show that the mechanism we observe is economically sound.

We provide a more thorough outline of these datasets and their collection methodology in Chapter 4.

Case-control studies. Vulnerability data is fraught with reporting and control problems: time-of-disclosure and time-of-patch is filled with "noise of unknown size" [105] and data on software versions and vendors is biased by limitations inherent to the disclosure process [38]. Unfortunately these

limitations are often ignored in literature [38], and generate hard to interpret conclusions (notable examples are [20, 108]). We propose the use of case control studies as a statistically sound way to measure different 'features' of vulnerability data. Although case-control studies are certainly not novel [45, 49], their use in information security is entirely novel. In our case, case-control studies represent an easily reproducible way to evaluate the effectiveness of vulnerability management policies by estimating the Risk Reduction they entail. Because case-control studies run on hindsight data to estimate correlations valid in foresight, their application can be extended to any operative environment that collects historical data on received attacks. This will be discussed in detail in Chapter 7.

Scope of work. Our data collection and research methodology requires some further consideration on the scope of this Thesis' work. In particular, field data adds realism to the analysis but limits the 'generality' of one's conclusions as it is often hard to extend results to other settings. Most attacks are delivered in an untargeted manner through web attacks [58, 93, 28], spam [64] and social engineering [26]. In this Thesis we focus on the 'general attacker' that 'massively deploys attacks in the wild against the population of users'. We make no claim on target attacks or the so-called APTs (Advanced Persistent Threats) that aim at a particular system of a particular organisation. For this reason we distinguish between *dedicated* and *average* attackers. A general model for the former type of attacker may be hard to design mainly because the attacker's motivation and target can be hard to predicted a-priori [62], and there is little data available to investigate this threat [28]. Consequently, evaluating the risk represented by a dedicated attacker is a rather pointless task as this is strongly case-dependent.

Case control studies represent a strong aid toward the internal and external validity of one's conclusions. Yet, they are not quite as powerful as

a (controlled) experiment setting is [128]. In particular, because in a case control study not all aspects of the 'experiment' are under the control of the researcher (e.g. data is collected elsewhere through an only partially known process), it is hard to build 'causal links' between an hypothesis and an observation. Rather, a case control study is limited to highlight the correlation (as opposed to causation) between the two.

Chapter 3

Measuring Vulnerabilities, Exploits, and Attackers

3.1 Software Vulnerabilities and Measures

One of the first large-scale studies on the life-cycle of a vulnerability has been conducted in 2004 by Arora et al. [20], where they evaluated how different vulnerability disclosure policies impact the velocity of patch and exploit arrival. They find that patching response time largely depends on vendor size, and that public disclosure of the vulnerability increases both the rapidity of the patching action and the arrival of the first exploit. This approach has recently been expanded by Shahzad et al. [108], who used data from public vulnerability sources [9, 10] to estimate vendor's performances by evaluating the average severity of the disclosed vulnerabilities and the average time between patch release and vulnerability disclosure. Unfortunately, the complexities of the vulnerability disclosure process [81, 105, 38] make these comparison hardly significant and representative of the real performance of the vendor. For example, certain vendors may have more 'hackers' or 'security researchers' interested in finding critical vulnerabilities in their software than the 'average vendor' . Moreover, real patching and disclosure times are 'obscured' by the disclosure process itself and therefore, to say it in the

words of the authors of NIST's NVD dataset, "the computation of patch times and exploit times would contain errors of unknown size." [9, 105]. For this same reason, the identification of so-called zero-day vulnerabilities (i.e. vulnerabilities exploited in the wild *before* being disclosed to the vendor) can be tricky. [108], by comparing exploit dates on OSVDB with disclosure dates on NVD, find that about 88% of vulnerabilities have a zero-day exploits. A figure in sharp contrast with this estimation is given by [28] who, by analysing records of attacks in the wild provided by Symantec, find that only a handful of vulnerabilities have an exploit in the wild before the date of disclosure. The vulnerability discovery process has been extensively studied in literature. 'Vulnerability Discovery Models' (VDMs) aim at modelling the overall number of vulnerabilities that will affect a certain software at a given point in time. Alhazmi et al. propose an exhaustive analysis of the main VDMs proposed in literature [15]. While numerous case studies are provided, including Operating Systems [16] and server software [129], the applicability of VDM remains uncertain [86].

3.1.1 The Common Vulnerability Scoring System

On top of the difficulties represented by estimating vulnerability and exploit disclosure and patch availability, remains the more general problem of 'measuring' the criticality of a vulnerability. The Common Vulnerability Scoring System (CVSS) [79], at its second version at the time of writing, is the standard-de-facto vulnerability metric used in the industry[1]. A CVSS score is assigned to each disclosed vulnerability, identified by a CVE-ID (Common Vulnerabilities and Exposures IDentifier). The CVSS score has been designed to give a readily available and standardised measure of the potential impact of a vulnerability over a system. Unfortunately its usage often deviates from its definition, and is often employed as a *risk* metric instead.Although CVSS

[1]The release of CVSS v3 is scheduled to happen in June 2015

resembles the form of a risk metric ($Score = likelihood \times impact$), the characterisation of the 'likelihood' variable is not clear in the CVSS case [30]. This is also reflected in the words of one of the authors of the CVSS score: "CVSS does not, and never has, made the claim that base score is significantly correlated with exploit probability" [100].

The CVSS framework considers three separate metrics: the base metric, the temporal metric, and the environmental metric. The first characterises the technical details of the vulnerability. The second captures characteristics that may vary with time, such as the existence of a patch, a known exploit, or of a workaround for the vulnerability. The third considers additional environmental factors to tailor the final estimation to the particular environment subject of the analysis. However the Temporal and Environmental metrics are not normally assigned to a vulnerability at the time of disclosure. Rather, the assessment along these metrics has to be carried within the vulnerable organization. Moreover, standards and common practices explicitly indicate the *base score* to be used for the assessment of the vulnerability [40, 95]. For this reason, we will limit this discussion to the latter.

The CVSS base score is divided in two submetrics: Exploitability and Impact. The former characterises the 'easiness' of exploitation of the vulnerability by measuring the complexity of its exploitation, how 'remote' from the system the attacker can be to deliver the exploit, and whether the attacker has to be authenticated on the system. From its composition it is easy to see why 'Exploitability' is often regarded as 'likelihood of exploitation': the easiness of exploit is interpreted as a proxy for likelihood of exploit. Although this claim has already been questioned [30], it is often implied in literature [108, 83, 38]. The CVSS 'Impact' metric provides an estimation of the impact of the vulnerability exploitation on the vulnerable system in terms of potential loss in Confidentiality, Integrity and Availability of the data. Table 3.1 reports a summary description of the CVSS base score submetrics.

Table 3.1: Summary table of CVSS base score metrics and submetrics.

Impact		Exploitability	
SubMetric	Description	SubMetric	Description
Conf.	Loss in data confidentiality	Access Vector	Where can the attacker attack from (e.g. remotely)
Integ.	Loss in data integrity	Access Complexity	Whether the successful exploitation depends on factors outside the attacker's control.
Avail.	Loss in service availability	Authentication	Whether the attacker needs to be logged in the system.

3.1.2 Vulnerability and patch management

Recent studies showed that several months pass between the release of a vulnerability patch and its application on the software [84]. In the literature, users' failure to take basic security measures has often been attributed to the incomplete model users have of cybersecurity threats [125, 37]. In an enterprise setting, patch management becomes critical as the application of software patches may break untested functionalities or dependencies, as well as causing downtimes that can affect system productivity [107, 54]. The trade-offs associated with patch management have often been pointed out in the literature [107, 36, 34]. Among these, Serra et al. [107] recently suggested a Pareto-optimal approach to vulnerability patching in enterprises, that merges attack graphs and vulnerability measures to maximise vulnerability coverage and system functionality. Similarly, Okhravi et al. [88] study the optimal amount of pre-patching testing that should be carried out in order to guarantee the best response time to the vulnerability disclosure and the best possible system uptime. Differently, Chen et al. [36] address the vulnerability patching trade-off from a slightly different perspective, and suggest that rather than diminishing the attack surface of the system or network [68]

by applying software patches, it may be possible to obtain a similar result through *software diversity*. The rationale is that, because vulnerability exploits and malware are platform-specific, system diversity will substantially increase the cost of traversing the attack graph for the attacker. In fact, an exploit shellcode or a piece of malware engineered to work on a specific platform (e.g. Windows 7, service pack 1), will not necessarily work on similar but not identical platforms (e.g. Win 7, SP 2) even if the vulnerability is still there. Differentiating the platform type entirely adds an additional layer of complexity as an attack against a Windows platform must be completely re-engineered to work on a Linux or MacOS or **nix* platform [110, 106].

The economics of vulnerability patching have also been considered in the literature, both in terms of patching efficiency [31] and from a game-theoretic perspective [35, 34]. Additionally, Gordon and Loeb [55] showed that the most economically viable patching solution may be one that leaves the most valuable assets vulnerable. This depends on the distribution of patching costs over assets. These economic aspects are especially interesting as the decision to patch a vulnerability has several consequences that are not limited to emerging technical issues or difficulties; on the contrary, in the literature has been shown that the decision to patch or not patch a vulnerability has externality effects lowering the general level of security due to the decentralisation of the patching decision [27], can affect stock prices [117], and can cause damage to non-vulnerable users (either because they have already applied the patch, or because they are not vulnerable - i.e. they don't have the vulnerable appliance installed on the system). These externalities have also been considered in the developing the new version of the Common Vulnerability Scoring System (v3) that, with the inclusion of the *Scope* metric, effectively measures whether the effect of a vulnerability resides on a different 'system' than the vulnerability itself [116].

3.2 Security Actors and Threats

To the aim of this thesis, we identify three main players relevant in the security management scenario: the software developer, the defender, and the attacker.

The software developer is the player who develops the software and typically has to maintain it by deploying software patches. The software interested by the security process can be either an open source software or a closed source software. The main difference between the two models is that open source code can be audited (and is written) by the user and developer community, while in closed source software this is not possible. In both cases vulnerability patching is an expensive process [71], and vendor performance may vary. A number of studies tried to identify 'good' and 'bad' software vendors that respond quicker to the vulnerability disclosure [108, 20]. The organisational and reputation costs attached to the patching and disclosure of a vulnerability are often high[2], and different disclosure and mitigation policies may emerge for different software vendors. For example, to contain costs of both type CISCO Systems discloses only 'high severity vulnerabilities' in their security advisories, while remaining vulnerabilities are disclosed through less prominent channels [5]. Accounting for this, no significant difference in patching behaviour between open and closed source software vendors is found [105], as otherwise often implied [108, 21].

The defender is the actor that has to deploy the patches to defend against the attacker and maintain the service continuity of the system or network. Patch deployment is a critical moment in system maintenance that sees on the one hand a better overall system security, and on the other the risk of 'ser-

[2]While exact figures on the cost of patching are hard to find and may vary significantly between software developers, a representative of a major European player estimated, in a private conversation with the author, that only acknowledging that a bug exists in the code costs for them about 100 US $, let alone verifying whether the bug represents a security threat, fixing it and testing and deploying the patch.

vice disruption' as the deployed patch may 'break' some functionality critical to the normal operation of the system [36]. For this reason, the criticality of the patch deployment process increases with the number of vulnerabilities to fix. Deploying all available patches immediately is usually not feasible in practice [36, 107] for technical and organisational reasons[3], and available data shows that indeed patching waiting times on users machines can vary widely [84]. Vulnerabilities to patch are therefore ordered in a queue, usually following a measure of vulnerability severity [95, 40]. A number of international standards and best practices exist to aid the system administrator in this process. Two notable examples are the NIST SCAP protocol [95], the software security management guidelines proposed by the NIST, and PCI-DSS [40], arguably the most applied international standard used for securing credit card transactions. On top of this exist a plethora of industry tools by Symantec, Rapid7, Qualys etc. that aim at helping the system administrator to prioritize patching work. All these approaches (standards, best practices, or commercial solutions) have a common denominator: the use of the Common Vulnerability Scoring System (CVSS in short) as a *metric for vulnerability risk*.

The hacker is the actor that finds and exploits the vulnerability. The term 'hacker' originally identified 'curious' and technologically-oriented actors whose main goal was to understand the inner functionalities of a piece of technology, a software, or a process [115, 120]. More recently, this 'reverse engineering' capability has been put in use by cyber-criminals to *exploit* software design and implementation flaws to modify the normal operational functionalities of the 'hacked' object (being that a telephone, a software, or a human answering a phone call or reading an email) to their advantage. The figure of the hacker remains however split in two main categories: *white hats*

[3]These include testing all the patches and their dependencies to assure that system and service functionality will not be affected, distributing the patch to the organisation's vulnerable systems, and addressing potential issues that may arise after the update

and *black hats*. The former are hackers that find vulnerabilities in software, write 'proof-of-concept' exploits, and ultimately disclose the vulnerability either directly to the vendor or to some third-party organisation such as iDefense or the Zero-Day Initiative. The white-hat has traditionally been a 'free-lancer', i.e. a security researcher that looks independently at software vulnerabilities and tries to sell them to the interested party [81]. The white-hat hacker is however often faced with the inherent difficulties of the vulnerability disclosure process, which may make the effort itself of disclosing the vulnerability not worth it. As noted by Miller [81], the vendor is often unhappy with the disclosure, and sometimes the hacker can face legal action. Recently the professional figure of the white-hat hacker changed to that of a 'corporate white-hat', i.e. a white hat that is now contracted by a corporation to find vulnerabilities in software (not necessarily of its own production). One notable example of this is Google's Project Zero [6], a project run by Google where hired white-hat hackers look for vulnerabilities in software, including Google's competitors' such as Microsoft and Apple. Similarly, the figure of the black-hat hacker has also gained momentum: black-hat hackers moved from the solitary, self-employed figure of the cybercriminal to more organised underground activities where the hacking is aided by a multitude of technical and infrastructural resources [58]. The figure of the black-hat has also been explored from a social standpoint [65, 120, 115].

3.3 Markets for Vulnerabilities

The importance of a clear understanding of the economic incentives and mechanisms standing behind the information security process have been outlined several times in literature [53, 17, 121, 90, 55]. New markets for information security have recently been proposed: for example, auction-based markets for vulnerability disclosure [90] have been suggested in the past, and

bug bounty programs [51] are nowadays becoming more and more popular. These initiatives partially address the problems attached to vulnerability mining and disclosure that Miller outlined in 2007 at WEIS [81]. As also outlined by Van Eeten et al. [121], market for malware and vulnerabilities offer sometimes perverse incentives that can undermine the security property they are supposed to enforce. For example, Asghari et al. [24] showed that the market incentives behind the release of cryptographic certificates (e.g. used to encrypt and sign the content retrieved from a web server) make more convenient to adopt bad security practices when releasing a certificate, or to hide entirely the compromise of a Certification Authority (as it already happened several times in the past [104]). Similarly, software vendors may have market incentives that go in the opposite direction of vulnerability disclosure [3, 117] and patching [55]; this, in turn, may discourage the security researcher from disclosing the vulnerability to the vendor in the first place, and may encourage instead the selling of the vulnerability to criminals. The debate on the best vulnerability disclosure strategy has been a prolonged one [20, 91, 23, 39], and is still not completely sedated [12]. Vulnerability disclosure may affect the reputation of the vendor, and indeed in the literature have been reported significant effects of vulnerability disclosure on the market value of the firm [117]. On top of this, the hacker who wants to sell the information about the vulnerability to the vendor has to 'prove' that the vulnerability exists without revealing too much information (otherwise he/she effectively gives the vulnerability away). Moreover, the issue of fair vulnerability pricing remains: how to evaluate the market price of a vulnerability? Bug bounty programs are now run by many major players in the IT industry, including Google, Microsoft and Facebook. A bug bounty program effectively encourages the disclosure of the vulnerability to the vendor by fixing vulnerability prices *ahead* of the disclosure, for different types of vulnerabilities. The security researcher can therefore assess beforehand the

value of his/her finding [1], and knowns that the disclosure will not result in legal action against him/her.

A perhaps more controversial portion of the market for vulnerabilities is dedicated to vulnerability and exploit trading between private researchers or agencies (the sellers), and governments (the buyers). Existing reports outline prices in the order of the hundred thousands dollars [4], much higher than the tens-of-thousands figures proposed by Google. These numbers have however been disputed by agencies selling malware and cyber-attacks to governments, such as France's Vupen and Italy's HackingTeam. The pricing of hacking tools and the value of the cybercrime markets have been often at the centre of discussion, and figures vary again widely. McAffee and Presided Obama report the cost of cybercrime markets to be around one trillion dollars (about 6% of the United States GDP in 2014[4]), while other figures are much more modest [19].

One of the issues that generates such wild estimations is that cybercrime markets are yet not very well understood. The trading dynamics of these markets, their operability and technological/economic (in)efficiencies are not fully comprehended. Cybercrime markets have recently been shown to be fraught with information asymmetry problems that make the trading in the markets effectively unsustainable [63]. Yet, empirical evidence from numerous studies shows that the attack tools traded in these markets do work [112, 18, 58, 26, 122], and the losses caused by cybercrime are real [57]. How can these observations be reconciled with the understanding that cybercrime markets *cannot* work? The explanation is that current markets are run under a different structure than IRC markets: rather than anonymous, free-to-join, unregulated communities of criminals, modern cybercrime markets are run as virtual forums [75, 18, 130, 82]. Forums provide an easy way for the community administrators to control the flow of users into the community and

[4] `http://www.tradingeconomics.com/united-states/gdp`

to enforce, through moderation, a number of rules that can be aimed - in a coherent market design structure - at mitigating the issues of information asymmetry [130]. The existence of operative cybercrime markets has indeed been reported in the literature [58, 82], and numerous studies analyzed the technical details behind the infection processes [93, 75] and the creation of botnets [111, 59]. A similar line of research also gave insights on the mechanics of spam [72] and diffusion of attacks [42]. Still, a precise understanding of the inner economic workings of these markets is not present in literature.

3.4 Attacker model and risk

Part of the problem that (not) understanding the economics of the attacker entails is that estimating the threat represented by the attacker is is a difficult exercise. The attacker model generally (explicitly or implicitly) accepted when planning security action is that of the all-powerful all-knowing attacker, an inheritance from cryptography [44]. In fairness, other attacker models exist, such as the 'Honest but curious' attacker that rather than acting outright maliciously, exploits the opportunity he/she might have of exfiltrating information from some channel. This model could be for example applied to the recent Snowden case, where an insider effectively used his access rights correctly until the 'last operation' was executed. The overall picture however does not change: the attacker can and will exploit *any* vulnerability on the system [2]. Somewhat in contrast, [93] showed that about two thirds of web attacks are generated automatically as opposed to being engineered for that specific attack. Moreover, the most popular cyber-criminal tools used to generate these attacks [112] feature in the order of 10-15 exploits [75]. It appears therefore that the majority of attacks may be skewed toward certain vulnerabilities only, and that assuming that the attacker can and will pursue all and every vulnerability in the system is, in most cases, unrealistic. Indeed, in

the literature evidence exists that attackers prefer certain vulnerabilities over others [85], and that most vulnerabilities remain simply unexploited [114]. The disparity between the current perception of the attacker and the trends shown in the data challenges the (conservative) intuition that 'one vulnerability is too many'. Yet, this philosophy is at the root of any standard or best practice for vulnerability and risk mitigation [40, 95], that requires action to be taken over effectively almost any vulnerability. This perception leads to 'naive' risk metrics whereby the risk is calculated as the sum of the vulnerabilities CVSS scores multiplied by the number of vulnerabilities with that criticality level [83]. More elaborated metrics of exposure to attacks exist [67, 124]; still, the substance remains the same: count the number of vulnerabilities in the system and use some criticality score such as CVSS to estimate the impact and the likelihood of an attack to happen. None of these methodologies account for the strong skew in attacker preferences consistently present in historical attack data [85, 114], and substantially rely on the 'allmighty attacker' model[5].

[5] The overestimation of the attacker capabilities (and/or willingness to attack) is a common problem in security, that sometimes leads to important (and unfounded) consequences [73].

Chapter 4

Data Collection

4.1 Vulnerabilities and Attacks in the Wild

In this Chapter we provide a comprehensive description of our datasets and the respective collection methodologies. Table 4.2, at the end of this Chapter, provides a summary of our collection efforts.

The universe of vulnerabilities. The National Vulnerability Database (NVD) is NIST's database for disclosed vulnerabilities. It reports a list of disclosed vulnerabilities that have been confirmed by software vendors, identified by the universal identifier 'CVE-ID' (Common Vulnerabilities and Exposures ID). Along with a description of the CVE, the dataset reports the vulnerable software and relevant software versions, and the CVSS base score associated with the vulnerability. Additional details on the technical properties of the vulnerability are also reported in NVD via the CVSS vector that specifies the value for each CVSS metric.

Data collection methodology: This dataset is publicly available at `http://nist.nvd.gov`.

The "white hat" exploits market. White-hat hackers report vulnerabilities to vendor and release proof-of-concept (PoC) exploitation code to demonstrate

the existence of the vulnerability. Datasets that report disclosed PoCs are the Exploit database (EDB) and the OpenSourced Vulnerability DataBase (OSVDB). Both these datasets cooperate with the Metasploit framework to gather data on exploits. However, it is important to note that, if an exploit is featured in EDB or OSVDB, it is not evidence that some company or individual actually reported to have suffered the exploitation in the wild. It only means some proof-of-concept exploitation code is known to exist. Moreover, proof-of-concept exploitation code may be hardly capable of crashing the vulnerable application, rather than allowing the attacker to actually exploit the vulnerability.

Data collection methodology: This dataset is publicly available at `http://www.exploit-db.com`. However, the archival version of the dataset does not directly refer to the CVE-ID of the vulnerability affected by the proof-of-concept exploit. In order to obtain this data, we built a Python script that collects the correct CVE-ID based on the exploit ID reported in the downloaded dataset.

The black markets for exploits. The EKITS dataset is a collection of vulnerabilities whose exploits are traded in the black markets and are bundled exploit kits (widely used attack tools in the underground [58]). Among the exploit kits considered for our study, we have the "most popular" ones as reported by Symantec in 2011 [112]. After a long process of ethnographic research, EKITS comprises almost 900 entries and 103 unique CVEs traded in the black market. Vulnerabilities included in the EKITS dataset affect only client-side and consumer applications running on Windows.

Data collection methodology: EKITS is partially based on Contagio's Exploit Pack Table, from where we got the names of the most popular exploit kits and some CVE entries. We expanded this list in both the list of exploit kits available in the markets, and the list of vulnerabilities bundled in the kits.

To do so, after much 'ethnographic research' we infiltrated the black markets and monitored the tools and vulnerabilities advertised there. A more precise description of the infiltration process is given in Section 4.2. To keep the list of vulnerabilities updated we created a web parser (in Python) that, hidden behind a TOR proxy, would scrape daily the main market page for new entries matching several (Cyrillic) keywords such as "связк*" (kit), "отстук" (the term commonly used to describe exploit success rates), "цен*" (russian for 'price'), and many others. The script's goal was to identify potentially interesting discussion topics in the forum markets. The integration of this data in EKITS was manual. This was a necessary step to perform given the impossibility of reliably parse Cyrillic text that often involves technical slang or abbreviated terms / typing errors.

Exploits in the wild. Obtaining reliable data on exploits in the wild is challenging. Companies are not prone to release data on the cyber-attacks they suffered from, for obvious commercial and reputation reasons. To the best of our knowledge, no reliable or reputable source for attacks against corporations exists yet. On the contrary, more reliable data can be found for non-targeted attacks. Symantec keeps two public datasets of signatures for local and network threats: the AttackSignature[1] and ThreatExplorer[2] datasets. These datasets contain all the entries identified as viruses or network threats by Symantec's commercial products at a given moment. Our SYM database is directly derived from these sources. However, it must be pointed out that this dataset is, by construction, limited to threats that Symantec identifies. These therefore mainly include threats directed against home systems, which are not, in general, victims of targeted attacks.

Data collection methodology: this dataset is publicly available on Syman-

[1] `http://www.symantec.com/security_response/attacksignatures/`
[2] `http://www.symantec.com/security_response/threatexplorer/`

tec's Security Response website. However, the dataset is largely unstructured with respect to the vulnerability information we are interested into, as it only reports a general description of the detected *attack signature*. In order to assess whether we could meaningfully use the dataset to collect exploited vulnerabilities, we sustained an extensive exchange with Symantec representatives to understand the nature of the available data and whether a reported CVE could be considered the CVE affected by a certain attack signature. From our exchange resulted that Symantec's effort in reporting a CVE in their attack signature description has substantially improved after 2009, with the initiation of their data sharing program WINE. Furthermore, we got assured that the reported CVE are always relevant to the affected signature. Because of the unstructured nature of this dataset, we built two independent parsers (the second one has been written by Dr. V.H. Nguyen) and checked that the result was the same. We are therefore confident that our data collection and interpretation is *complete* and *correct* with respect to Symantec's data creation process.

Records of attacks in the wild. Symantec runs a data sharing program, the Worldwide Intelligence Network Environment, or WINE in short[3]. The intrusion-prevention telemetry dataset within WINE provides information about network-based attacks detected by Symantec's products. WINE is indexed by *attack signature IDs*, unique identifiers for an attack detected by the firm's security solutions, which can be linked to the affected CVE, if any, through Symantec's *Security Response*[4] dataset (i.e. SYM). Data for the experiments reported in this thesis is referenced and available for sharing at Symantec Research Labs under the WINE Experiment ID *WINE-2012-008*.

Data collection methodology: in order to have access to the WINE dataset

[3] https://www.symantec.com/about/profile/universityresearch/sharing.jsp
[4] https://www.symantec.com/security_response/

Table 4.1: Summary of our datasets

DB	Content	#Entries
NVD	CVEs vulnerabilities	49.624
EDB	Publicly exploited CVEs	8.189
SYM	CVEs exploited in the wild	1.289
EKITS	CVEs in the black market	103
WINE	Records of attacks in the wild	75.000.000

researchers have to write a research proposal that is subject to Symantec's internal review process. We wrote and got our proposal accepted in May 2012. Access to the WINE platform is possible only *in loco* over at Symantec Research Labs. It was therefore necessary to extensively prepare the experiment *before* moving to the other side of the ocean to perform the data collection. Given the extension of the WINE dataset this preparation phase lasted several months during which frequent calls and e-mail exchange with Symantec where necessary to make sure the experiment design was correct. To complete the design phase we used the database schema of WINE and a VM provided by Symantec.

Table 4.1 summarizes the content of each dataset and the collection methodology. The datasets are available for the scientific community upon request.

4.2 The Underground Markets

From September 2011 to November 2011 we performed an informal analysis with security experts working in the cybercrime domain to identify the most prominent markets in the underground. These resulted to be all run in Russian, with a few exceptions only. The results of that analysis identified one market in particular, HackMarket.ru[5], that was very active and where the ma-

[5]HackMarket.ru is a fictional name we attribute to the market to not hinder future research.

jor players of the cybercrime community allegedly operate. In November 2011 we infiltrated HackMarket.ru and observed that there we could find for trade all the attack tools and malware pieces reported as 'most prominent in the underground' by multiple industry reports [112, 26, 111, 77, 70, 76], as well as the most influential malware authors such as Paunch [8] and others. We kept monitored other markets as well, but those revealed to be not top-of-the-class markets, where few tools where actually advertised and where interested costumers were much less active when compared to HackMarket.ru. We therefore keep HackMarket.ru as our case study of an underground market. The parser written to build the EKITS dataset was originally designed to monitor HackMarket.ru as well as the other markets, but for the aforementioned reasons it is now built around HackMarket.ru exclusively.

4.2.1 Markets description

Carders.de In 2010 an online underground market for credit cards and other illegal goods, Carders.de, have been exposed by a hacking team named "`inj3ct0r`" and leaked, at the time, through underground channels (i.e. a Google search wouldn't help) [82]. We obtained the original dataset through side channels. We have no means to assess whether the dataset was manipulated. Direct comparison with other releases of the dump show no difference. The leaked package contains a Structured Query Language (SQL) dump of the database, a copy of the Owned and Exp0sed Issue no. 1 (documenting the leak) and an added text file containing all private messages on the forum. By examining the added notes Owned and Exp0sed Issue no. 1 we were able to create a replica of the original Carders.de forum. This allows us to explore market operations, evaluate the reputation mechanisms that were implemented at that time, and go through users' posting history and dates. The data consists of forum posts and private message records spanning 12 months from 1 May, 2009 to May 1, 2010 containing a total of 215.328 records.

HackMarket.ru HackMarket.ru is a market for exploits, botnets and malware. It is also one of the main markets that introduced exploit-as-a-service [58] in the cyberthreat scenario, as we find there the main players and products that the industry reports be driving the majority of reported web-attacks [113]. Indirect evidence of this markets' efficacy is the recent burst in cyberattacks driven by means of tools, services and infrastructures traded or rented in these markets [58, 111, 18]. HackMarket.ru appeared in 2009 in the Russian underground. Differently from Carders.de, HackMarket.ru has a flat trading structure, whereby traders all participate in the same marketplace. In contrast to other hacker fora studied in the literature [65], it is not public. HackMarket.ru is run in Russian, and very little interaction happens in English. The trading sections in this market are, like in Carders.de, organised by 'topic of interest'. The virus-related area of the market is by far the most popular one, with tens of thousands of posts at the time of writing. Other goods of interest for the marketeers of HackMarket.ru are 'Internet traffic' (i.e. redirectable user connections for spam or infection purposes), stolen access credentials, access to infected servers, spam, bank accounts, credit cards and other compromised financial services. To access the market the forum administrators perform a background check on the participant, that has to provide additional profiles that provably belong to him/herself on other underground communities. We joined this community in 2011 and remained undercover since. For HackMarket.ru we do not have an SQL dump of the market, but we will provide instead first-hand evidence that the problems we highlighted for Carders.de are not present here. We index our qualitative analysis by referencing internal archived references taken from HackMarket.ru in the format $[ID\ n]$, with ID being an internal code we use to classify the evidence and n being the document number.

4.2.2 Infiltrating HackMarket.ru

To infiltrate HackMarket.ru has proven to be a far from simple task. We infiltrated HackMarket.ru twice, as our first account was banned from the market. The two operations have been characterised by very different problems we had to address. The first time, the real issue has been to identify a significant and interesting market to infiltrate. Choosing HackMarket.ru as a market representative for cybercrime operations was possible only after much ethnological research on several other underground communities. Not speaking Russian and lacking of scientific guidance from the literature (where data analysis involving underground markets is performed over leaked data rather than data collected first-hand) made things worse in this respect. This research effort lasted about three months. Once found and selected HackMarket.ru, the first obstacle was to have access to the communities without exposing the University or myself to future possible hazards. The obvious solution to this has been to access the communities only behind the TOR network. In order to first access the markets we registered an email address with a Russian domain and compiled a 'user description' in correct Russian, with the help of a colleague, Anton Philippov. This was enough to have our first access to the community granted. This however lasted a few months only, after which we were banned from accessing the forum. This ban and the additional segregation of the community that followed was motivated by the arrest of one prominent member of the market community: *Paunch*, the author of the infamous Black Hole exploit kit[6].

To re-enter the community proved to be substantially harder than in our first try. The community closed the entrance to anybody who was not explicitly selected by the forum administrators. We often tried to re-subscribe with several different (fictitious) identities, but systematically failed[7]. The effort

[6] `http://krebsonsecurity.com/2013/12/meet-paunch-the-accused-author-of-the-blackhole-exploit-kit/`

[7] The author, not trusting free proxy services in Moscow, exploited a personal trip to Russia to use a

required to re-access the markets lasted several months, and was partially performed with the help of Stanislav Dashevskyi. We employed a bottom-up approach: the idea was to study the low-end markets as a means to access the high-end market we were (re-) aiming for. We infiltrated several of those and, with the help of Stanislav, built a profile for each community, trying to outline those that are the most tied with HackMarket.ru. Research over these communities was aimed at outlining the social, linguistic, and technical characteristic of a 'typical' market participant in these communities. Leveraging this understanding, we built a user profile on one market community we selected for its apparent closeness to HackMarket.ru. This process lasted about 6 months. We then applied to access HackMarket.ru again and gave as a credential our participation in the other community. This attempt was successful. We are now 'mild participators' in the community, in order to avoid running in the same problem again were other prominent members of the community arrested.

Saint Petersbourg IP address to attempt a new subscription, but with no success.

Table 4.2: Summary of data and collection methodologies.

Dataset	Collection Methodology	Type of analysis	Collection efforts
NVD	XML parsing	Quantitative; Exploratory; Case control study	Download
EDB	Web parsing	Quantitative; Exploratory; Case control study	Download and build web parsers
SYM	Web parsing	Quantitative; Exploratory; Case control study	2 weeks. Discussion with Symantec to understand their collection and reporting process.
EKITS	Manual exploration + Contagio's Exploit pack table	Quantitative; Exploratory; Case control study; Field data	6 months. Infiltration of the black markets; built stealth parsers; ethnographic research.
WINE	Participation in Symantec Inc.'s WINE project	Quantitative; Field data	8 months. Proposal to Symantec; project validation and preparation. Visiting Symantec oversea (2 weeks).
Carders.de	Collection through side channels	Qualitative; quantitative; case study	Find the dataset.
HackMarket.ru	Market infiltration	Qualitative; quantitative (only for participant reputation levels); case study	1.5 years for analysis and data collection. Additional 3 months for first entry, and 6 months for second entry.

Chapter 5

Data Exploration

In this Chapter we provide a first explorative description of our datasets. In particular, in Section 5.1 we outline a map of vulnerabilities to see how do our datasets overlap and how the CVSS score for vulnerability severity is mapped over exploits in the wild and disclosed vulnerabilities. In Section 5.2 we look at our WINE data to explore how vulnerability exploitation (and therefore vulnerability risk) is distributed among vulnerabilities.

5.1 A Map of Vulnerabilities

In the following we provide a first high-level view of the problem with criticality-based vulnerability management practices that ultimately use the CVSS score as an ordering metric for vulnerability mitigation. Figure 5.1 reports a Venn diagram of our datasets. Area size is proportional to the number of vulnerabilities that belong to it; the color is an indication of the CVSS score. Red, orange and cyan areas represent `HIGH`, `MEDIUM` and `LOW` score vulnerabilities respectively. This map gives a first intuition of the problem with using the CVSS base score as a 'risk metric for exploitation': the 'red vulnerabilities' located *outside* of SYM are 'CVSS false positives' (i.e. `HIGH` risk vulnerabilities that are not exploited); the 'cyan vulnerabilities' in SYM are instead 'CVSS false negatives' (i.e. `LOW` and `MEDIUM` risk vulnerabilities that *are* ex-

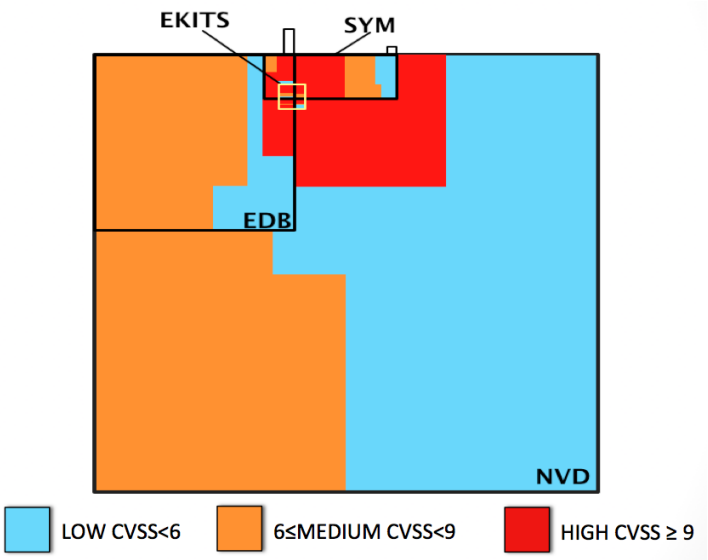


Figure 5.1: Map of vulnerabilities per dataset. Overlapping areas represent common vulnerabilities among the datasets, as identified by their CVE-ID. Area size is proportional to the number of vulnerabilities. In red vulnerabilities with $CVSS \geq 9$. Medium score vulnerabilities ($6 \leq CVSS < 9$) are orange; low score vulnerabilities are cyan and have $CVSS < 6$. CVSS scores are extracted from the NVD database as indexed by the respective CVE-ID. The two small rectangles outside of NVD are vulnerabilities whose CVEs were not present in NVD at the time of sampling. These CVEs are now present in NVD.

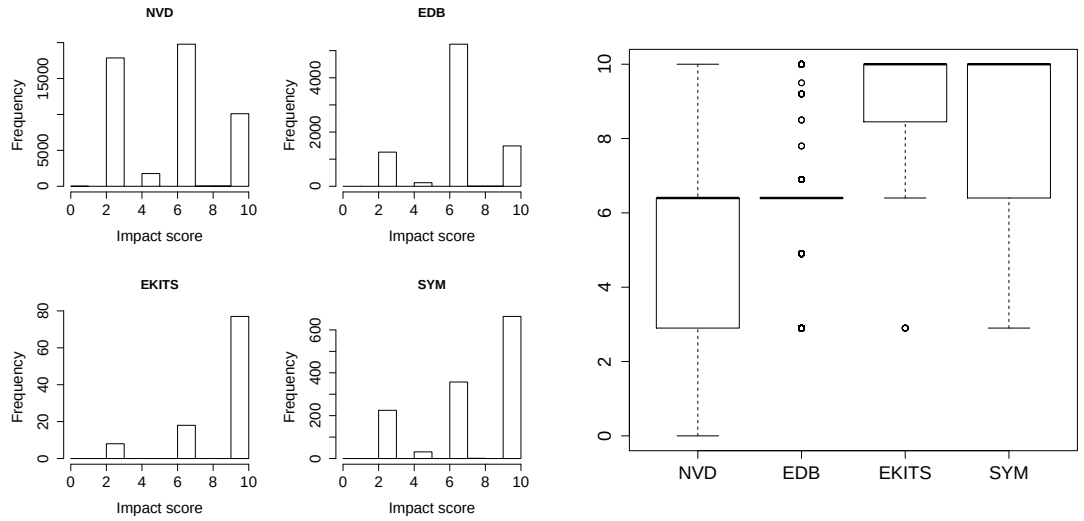
ploited). A relevant portion of CVSS-marked vulnerabilities seem therefore to represent either false positive or false negatives.

5.1.1 CVSS score breakdown

In this Section we perform a breakdown of the CVSS Impact and Exploitability subscores (see Table 3.1) in our datasets.

Breakdown of the Impact subscore

Figure 5.2 depicts a histogram distribution of the Impact subscore. The distribution of the Impact score varies sensibly depending on the dataset. For example, in EDB scores between six and seven characterize the great



The histogram on the left represents the frequency distribution of the CVSS Impact values among the datasets. The boxplot on the right reports the distribution of values around the median (represented by a thick horizontal line). Outliers are represented by dots.

Figure 5.2: Histogram and boxplot of CVSS Impact subscores per dataset.

majority of vulnerabilities, while in SYM and EKITS most vulnerabilities have Impact scores greater than nine. This is an effect of the different nature of each dataset: for example, a low Impact vulnerability may be of too little value to be worth the bounty by a security researcher, and therefore these may be under-represented in EDB [81]; medium-score vulnerabilities may instead represent the best trade-off in terms of market value and effort required to discover or exploit. In the case of SYM and EKITS vulnerabilities, it is unsurprising that these yield a higher Impact than the average vulnerability or proof-of-concept exploit: these datasets feature vulnerabilities actually chosen by attackers to deliver attacks, or to be bundled in tools designed to remotely execute malware. The different distribution of the CVSS Impact subscore among the datasets is apparent in the boxplot reported in Figure 5.2. The distribution of Impact scores for NVD and EDB is clearly different from (and lower than) that of EKITS and SYM.

To explain the gaps in the histogram in Figure 5.2, we decompose the

distribution of Impact subscores for our datasets. In Table 5.1 we first report the incidence of the existing CIA values in NVD. It is immediate to see that only few values are actually relevant. For example there is only one vulnerability whose CIA impact is 'PCP' (i.e. partial impact on confidentiality, complete on integrity and partial on availability). Availability almost always assumes the same value of Integrity, apart from the case where there is no impact on Confidentiality, and looks therefore of limited importance for a descriptive discussion.

For the sake of readability, we exclude Availability from the analysis, and proceed by looking at the two remaining Impact variables in the four datasets. This inspection is reported in Table 5.2. Even with this aggregation on place many possible values of the CIA assessment remain unused. 'PP' vulnerabilities characterize the majority of disclosed vulnerabilities (NVD) and vulnerabilities with a proof-of-concept exploit (EDB). Differently, in SYM and EKITS most vulnerabilities score 'CC'. This shift alone can be considered responsible for the different distribution of scores depicted in Figure 5.2 and underlines the difference in the type of impact for the vulnerabilities captured by the different datasets.[1]

Breakdown of the Exploitability subscore

Figure 5.3 shows the distribution of the Exploitability subscore for each dataset. Almost all vulnerabilities score between eight and ten, and from the boxplot it is evident that the distribution of exploitability subscores is

[1] Metrics to measure the impact of a vulnerability other than the CVSS CIA assessment could be derived from environmental or infrastructural considerations on the vulnerable systems. Possible examples of this are the criticality of the vulnerable system or software in the particular operative context of an organisation, or the impact factor of the system or its components measured over a decay in performance caused by the vulnerability [60]. While several possible metrics to measure vulnerability impact can be devised, we refer here to CVSS's CIA assessment as it is standardised in the industry, and general enough to abstract away from case-specific assessments of vulnerability impact (e.g. using attack surfaces or more case-specific metrics like performance decay indicators).

Table 5.1: Incidence of values of CIA triad within NVD.

Confidentiality	Integrity	Availability	Absolute no.	Incidence
C	C	C	9972	20%
C	C	P	0	-
C	C	N	43	<1%
C	P	C	2	<1%
C	P	P	13	<1%
C	P	N	3	<1%
C	N	C	15	<1%
C	N	P	2	<1%
C	N	N	417	1%
P	C	C	5	<1%
P	C	P	1	<1%
P	C	N	0	-
P	P	C	22	-
P	P	P	17550	35%
P	P	N	1196	2%
P	N	C	9	<1%
P	N	P	110	<1%
P	N	N	5147	10%
N	C	C	64	<1%
N	C	P	1	<1%
N	C	N	43	<1%
N	P	C	17	<1%
N	P	P	465	1%
N	P	N	7714	16%
N	N	C	1769	4%
N	N	P	5003	10%
N	N	N	16	<1%

indistinguishable among the datasets. In other words, Exploitability can not be used as a proxy for likelihood of exploitation in the wild. A similar result (but only for proof-of-concept exploits) has also been reported in [30]).

In Table 5.3 we decompose the Exploitability subscores and find that most

Table 5.2: Combinations of Confidentiality and Integrity values per dataset.

Confidentiality	Integrity	SYM	EKITS	EDB	NVD
C	C	51.61%	74.76%	18.11%	20.20%
C	P	0.00%	0.00%	0.02%	0.04%
C	N	0.31%	0.97%	0.71%	0.87%
P	C	0.00%	0.00%	0.01%	0.01%
P	P	27.80%	16.50%	63.52%	37.83%
P	N	7.83%	0.97%	5.61%	10.62%
N	C	0.23%	0.00%	0.18%	0.22%
N	P	4.39%	2.91%	5.07%	16.52%
N	N	7.83%	3.88%	6.75%	13.69%

vulnerabilities in NVD do not require any authentication (Authentication = (N)one, 95%), and are accessible from remote (Access Vector = (N)etwork, 87%).

Table 5.3: Exploitability Subfactors for each dataset.

	metric	value	SYM	EKITS	EDB	NVD
Exploitability	Acc. Vec.	local	2.98%	0%	4.57%	13.07%
		adj.	0.23%	0%	0.12%	0.35%
		net	96.79%	100%	95.31%	86.58%
	Acc. Com.	high	4.23%	4.85%	3.37%	4.70%
		medium	38.53%	63.11%	25.49%	30.17%
		low	57.24%	32.04%	71.14%	65.13%
	Auth.	multiple	0%	0%	0.02%	0.05%
		single	3.92%	0.97%	3.71%	5.30%
		none	96.08%	99.03%	96.27%	94.65%

For this reason the CVSS Exploitability subscore resembles more a constant than a variable, and can not therefore properly characterise the 'likelihood' of the exploitation.

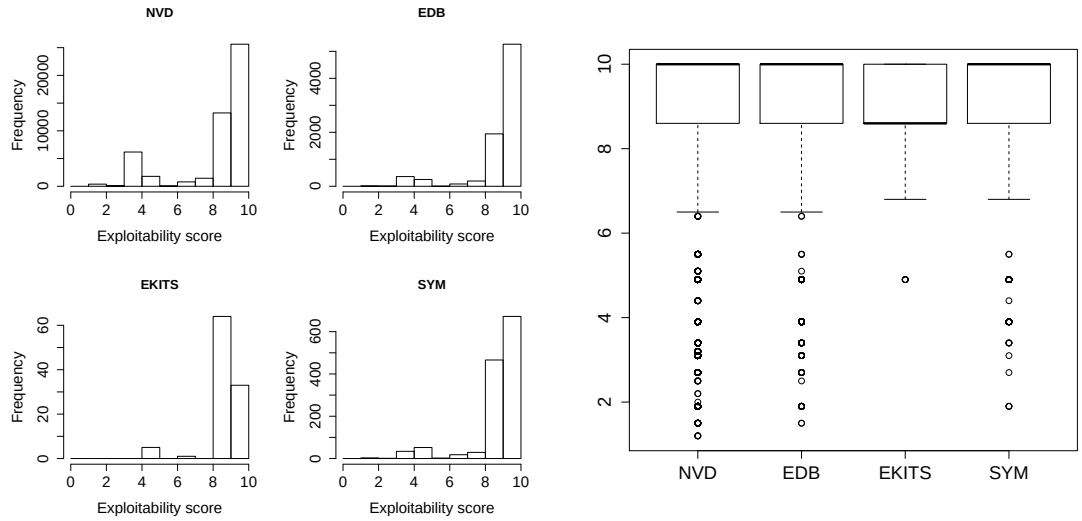


Figure 5.3: Distribution of CVSS Exploitability subscores.

Table 5.4: Categories for vulnerability classification and respective number of vulnerabilities and attacks recorded in WINE.

Category	Sample of Software names	No. Vulns.	Attacks (Millions)
PLUGIN	Acrobat reader, Flash Player	86	24.75
PROD	Microsoft Office, Eudora	146	3.16
WINDOWS	Windows XP, Vista	87	47.3
Internet Explorer	Internet Explorer	55	0.55
Tot		**374**	**75.76**

5.2 The Heavy Tails of Vulnerability Exploitation

From Figure 5.1 it appears that only a small fraction of vulnerabilities is exploited in the wild. This however has two limitations:

1. We are only looking at the boolean variable 'Exploit exists' (Yes or No), without considering that volumes of attack per vulnerability may be strongly skewed.

2. We are not accounting for the selection bias inherent in SYM, whereby only vulnerabilities covered by Symantec's commercial products are reported.

To address the first point we use data reported in WINE on attacks per vulnerability. As to the second point, we take additional precautions in handling the data. We inspected WINE's vulnerabilities and, using software names reported in NVD, we grouped them in eight software categories: Internet Explorer, Plugins, Windows, Productivity, Other Operating Systems, Server, Business Software, Development Software. Because WINE consists largely of data from Symantec's consumer security products, we may have a self-selection problem in which certain software categories are not well represented in our sample. We therefore limit our analysis to the first four categories, for which we consider our sample to be representative of exploits in the wild: Internet Explorer, PLUGIN, WINDOWS and PROD(uctivity). From a discussion with Symantec it emerges that also SERVER vulnerabilities can be considered well represented in SYM and WINE. We do not consider those here for brevity but include them later in the analysis (Section 6.1). Note that distribution of attacks detected by Symantec may also be an artefact of the data generation process for the WINE dataset. In particular, it may reflect Symantec's detection rates rather than real frequency of attacks. In particular, we find that WINE reports attacks against vulnerabilities disclosed over a wide range of years, spanning from 1999 to 2012. Because fewer users might be vulnerable to older vulnerabilities, the detection rate of these may be lower than the detection rate of more recent attacks. Similarly, Symantec may be detecting mainly attacks against certain types of attack vectors (e.g. a malformed file or a piece of javascript) received by different applications. Our analysis mitigates this problem by a) controlling by software type in order to group attacks whose attack vectors are similar; b) considering only vulnerabilities disclosed in a limited time window (2009-2012) as to minimize the variance in detection rates. Our analysis comprises 374 vulnerabilities and 75.7 Million attacks recorded from July 2009 to December 2012. Table 5.4 reports the identified categories and the number of

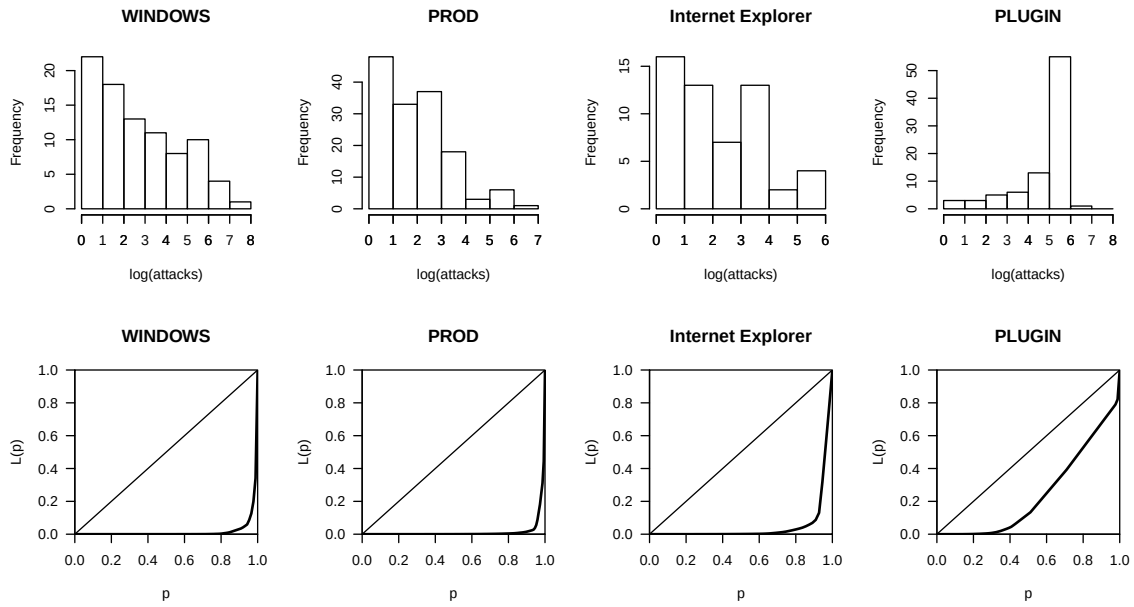


Figure 5.4: Top row: histogram distribution of logarithmic exploitation volumes. Bottom row: Lorentz curves for exploitation volumes in the different categories. p % of the vulnerabilities are responsible for L(p)% of the attacks.

respective vulnerabilities in WINE.

In Figure 5.4 we report the histogram distribution of the (logarithmic) attack volumes for each vulnerability by the category (top row) and the respective Lorentz curve distribution (bottom row). The histogram distribution clearly shows that for WINDOWS, PROD and Internet Explorer the frequency of vulnerabilities with x attacks is inversely proportional to the logarithm of x. In other words, a (very) small fraction of vulnerabilities is responsible for orders of magnitude more attacks than the remaining vulnerabilities.

A clear way to visualize this is through a Lorentz curve. A Lorentz curve describes the p percentage of the population (of vulnerabilities) that are responsible for the $L(p)$ percent of attacks. The diagonal represents an 'equilibrium state' where each vulnerability is responsible for the same volume of attacks. The further away the two curves are, the higher the 'disparity' in

Table 5.5: $p\%$ of vulnerabilities responsible for $L(p)\%$ of attacks, reported by software category.

Category	Top $p\%$ vulns.	$L(p)\%$ of attacks
WINDOWS	20%	99.6%
	10%	96.5%
	5%	91.3%
PROD	20%	99.5%
	10%	98.3%
	5%	94.4%
Internet Explorer	20%	97.1%
	10%	91.3%
	5%	68.2%
PLUGIN	20%	46.9%
	10%	31%
	5%	24%

the distribution of attacks per vulnerability. As depicted in Figure 5.4, for WINDOWS, PROD and Internet Explorer the two curves are very markedly apart, indicating that the great majority of vulnerabilities are responsible for only a negligible fraction of the risk in the wild. Table 5.5 reports the distribution of attacks recorded in the wild per vulnerability. We report the top 20, 10 and 5 percent of vulnerabilities and the percentage of attacks in the wild they are responsible for. The most extreme results are obtained for WINDOWS and PROD, for which the top 5% vulnerabilities carry more than 90% of the attacks and the top 10% the almost totality. 'Milder' results are obtained for Internet Explorer: the top 10% carries 90% of the attacks, but the top 5% carries 'only' 68%, meaning that among the top 10% vulnerabilities attacks are distributed more equally than in other categories. The less extreme result is obtained for PLUGIN, where the distribution of exploitation attempts seems more equally distributed among vulnerabilities.

With this last exception, we observe that a general rule for vulnerability exploitation is that, within any software category, less than 10% of attacked

vulnerabilities are responsible for more than 90% of the attacks.

This first, exploratory analysis of the distribution of attacks in the wild is *prima-facie* evidence that vulnerability exploitation is not uniformly distributed among vulnerabilities, and consequently that certain vulnerabilities may represent much higher risk for the final user than most others.

Chapter 6

On the Feasibility of Risk-based Vulnerability Management

From the Analyses in Section 5.1 and 5.2 it emerges that on the one hand the attacker is not choosing vulnerabilities to exploit using the CVSS score, and on the other that he/she tends to exploit a small fraction of vulnerabilities only that, as a result, are responsible for the great majority of risk in the wild. From a high-level perspective, these observations seem to support our Thesis.

To further investigate this, in this Chapter we provide evidence supporting the 'enabling hypotheses' outlined in Section 2.1. This Chapter unfolds as follows: Section 6.1 presents the model of the Work-Averse Attacker, whereby the attacker acts rationally when choosing which vulnerabilities to exploit. Importantly, from the model the exploitation trends shown in Figure 5.4 emerge naturally. Our findings strongly support Hypothesis 1.

In Section 6.2 we investigate the maturity and economic and technological sustainability of the cybercrime markets. The discussion starts in Section 6.2.1 where we investigate the maturity of cybercrime markets (Proposition 1). The analysis unfolds by comparing data on two underground markets, Carders.de and HackMarket.ru, with respect to a common set of Hypotheses testing their stability as economic entities. From our analysis we conclude

that Proposition 1 is supported by the data.

To test Proposition 2, in Section 6.2.2 we test in our MalwareLab a set of attacking tools leaked from the black markets. In particular, we test their exploit reliability and resiliency against continuous software updates. Our findings confirm that these tools are efficient and capable of successfully exploiting vulnerabilities over configurations spanning several years.

Finally, in Section 6.2.3 we propose a two-stage model of the underground markets whereby the seller that sells the exploit has strong incentives in behaving fairly in order to maximise his/her profit function. By solving the model we show that the underground markets are economically sound from a trading perspective, and conclude therefore that Hypothesis 2 holds.

Each Section starts with a brief summary of the Hypotheses outlined in Chapter 2.

6.1 The Attacker is Rational and Work-Averse

Running Hypothesis	Hypotheses Testing
Hyp. 1. The attacker ignores most vulnerabilities and massively deploys exploits for a subset only.	**Hyp. 1a**. The attacker will massively use only one exploit per software version. **Hyp. 1b**. The fraction of attacks driven by a particular vulnerability will decrease slowly in time. **Corollary to Hyp. 1b**. The attacker waits a longer period of time to introduce an exploit for software types under a slow update cycle than for others.

The idea that an attacker may not be interested in exploiting 'all' vulnerabilities in the system emerges from a simple observation: in most cases, he/she needs to attack only one ('powerful' enough) vulnerability among the many that affect that particular software.[1] In a broader sense, the expected utility of an exploit for a vulnerability v at time t $E[U_{t,v}]$ comes from the revenue r the attacker can extract from the fraction $n(t, v) \in [0, N]$ of the N systems in the wild the vulnerability allows him to attack at time t. The revenue r an attacker can get from the system out of the exploitation of one vulnerability may depend on two factors:

1. The potential value of the attacked system.

2. The impact I of the vulnerability on the system. For example, a vulnerability granting full administrative access is likely to allow the attacker to extract more revenue from the attacked system.

We therefore model the extracted revenue per attacked system $r(I(v_i))$ as a function of the vulnerability impact. The cost c of the attack comprises the cost of developing/buying the exploit and the cost of delivering the attack by means, for example, of some attacking infrastructure ([18, 58]).

[1] We here refer to a 'worse-averse' agent as an agent that sees work effort as a disutility, i.e. as emerges from the agent's utility function.

The expected utility of an exploit for a vulnerability v at time t is therefore:

$$E[U_{t,v}] = n(t,v) \times r(I(v)) - c(v) \tag{6.1}$$

Note that $lim_{t\to\infty} n(t,v) = 0$ as users update their systems and the exploit for v loses efficacy in the wild. When the efficacy of the old exploit drops too low, the attacker will dedicate his/her resources (abandoning v)[2] to look for a new exploit v'.

Under the assumption that exploit development is costly and an attacker is work averse, s/he will develop the exploit for a new vulnerability v' after some time $t + \delta > t$ if the expected value for v at $t + \delta$ is lower than the expected value for v' at $t + \delta$:

$$E[U_{t+\delta,v'\cup V}] - E[U_{t+\delta,V}] > 0 \tag{6.2}$$

where V is the set of vulnerabilities the attacker already exploits. The boundary condition to choose v' is therefore:

$$n(t+\delta, v' \cup V) \times r(I(v' \cup V)) - c(v') > E[U_{t+\delta,V}] \tag{6.3}$$

By generalising Eq. 6.3 it is possible to obtain the decision condition for the attacker over an arbitrary vulnerability v_j

$$n(t+\delta, v_j \cup V) \times n \times r(I(v_j \cup V)) - c(v_j) > \sum_{i=1}^{j-1} E[U_{t+\delta,v_i}] \tag{6.4}$$

At this point, the attacker will introduce a new exploit for v_j if:

$$c(v_j) < n(t+\delta, v_j \cup V) \times r(I(v_j \cup V)) - \sum_{i=1}^{j-1} E[U_{t+\delta,v_i}] \tag{6.5}$$

[2]The vulnerability finding and exploit writing processes are very time consuming and require the allocation of plenty of resources ([81, 18, 58]). While an attacker can always re-use old technology (i.e. old exploits), maintaining a certain exploit operative requires maintenance costs in terms of both technological resources and time. When not looking for a new exploit, we do not put any constraint on how many exploits the attacker wants to use.

The cost for v_j is therefore bounded by the revenue that can be extracted from all pre-existing exploits the attacker may maintain. It is immediate to see that the more previous exploits have been developed, the higher the potential revenue from v_j must be in order to overcome the cost constraint. With $\sum_{i=1}^{j-1} E[U_{t+\delta,v_i \cup V}]$ growing with the number of available exploits, the upper-bound cost $c(v_j)$ for the new exploit tends to zero. The diminishing return seen in Eq. 6.5 has two main consequences:

1. The attacker is able to afford a diminishing amount of exploits in time.
2. Assuming a direct relationship between exploit quality and cost of the exploit ([81, 22]), the quality of the new exploits would tend to decrease with the amount of exploits available to the attacker.

The cost constraint is positive and greater than zero only when $n(t+\delta, v_j \cup V)$ is greater than $\sum_{i=1}^{j-1} n(t+\delta, v_i \cup V)$ because $\lceil n(t+\delta, v_j \cup V) + \sum_{i=1}^{j-1} n(t+\delta, v_i \cup V) \rceil \leq N$, so there is a cap on the total revenue that can be extracted. In other words, the attacker will build a new reliable exploit only when the overall revenue the attacker can extract from the old exploits drops because of too few vulnerable systems in the wild (i.e. because users at large upgraded their systems).

6.1.1 Data preparation

To build our dataset, we first reconstruct the history of attacks received by every user in WINE. To evaluate the sequence of attacks against a certain software, we then collect all the pairs $< attack1, attack2 >$ of attacks that a user received, and keep track of the time delay (measured in days) between the two attacks. We then group the data by pairs of attacks and delta in time, and count how many users have been affected by that sequence and how many attacks of that type have been observed in the wild. Table 6.1 reports an excerpt from the dataset. Each row represents a succession of

1st attack	2nd attack	Delta days	Affected machines	Volume of attacks
a	a	192	23544	58322
b	c	11	6	6
b	e	580	10	10
d	f	861	389	432
e	b	644	26	43

Table 6.1: Excerpt from our dataset. CVE-IDs are obfuscated as a, b, c, etc. Each <`1st attack, 2nd attack, delta`> tuple is unique in the dataset. The column `Affected machines` reports the number of unique machines receiving the second attack delta days after `1st attack`. The column `Volume of attacks` is constructed similarly but for the number of received attacks.

attacks. The first column and the second column report respectively the (censored) CVE-ID of the attacked vulnerability in the first and in the second attack. The third column reports the number of unique systems in the WINE platform affected at least once by that tuple; the fourth column reports the overall number of attacks detected; the fifth the distance between the two attacks expressed in days. Note that for anonymity reasons we aggregate the attacks against each unique machine in WINE into an ensemble of identical attacks. This does not represent a threat to the generality of our results as we are here interested in measuring attacks at an aggregate level rather than singularly for each user. The columns reporting the software affected by the vulnerability, the latest affected software version and the software's category are here omitted for brevity.

Additional care must be taken when evaluating vulnerability timing data [105]. In particular, because we are evaluating attackers' attitude at developing new vulnerability exploits in time, we need to 1) identify vulnerabilities that are disclosed at the same time and 2) eliminate subsequent attacks targeting vulnerabilities that are very far away in time, as these say little about the attackers' exploit development process. For this reasons, we only consider the tuples <`1st attack, 2nd attack`> respecting the following

constraints:

1. The vulnerability exploited in the first attack was disclosed before or at most 120 days after the vulnerability for the second attack. This robustly large interval has been chosen according to how the vulnerability disclosure process works.

2. The second exploited vulnerability is less than three years older than the first. We choose this time frame as it matches the length of the historic records we have for each WINE user, given the three-year interval covered by our sample.

In the first row of Table 6.1 the two subsequent attacks are against the same vulnerability. The tuple $< a, a >$ affected most machines and was the vector of a high number of attacks in the sample. For example, we find almost 60 thousand attacks against 23.5 thousand users that have received a second attack against a 192 days (6months) after the first. The second and third row report two instances where an attack on b has been followed by an attack against two other vulnerabilities. The third and fourth rows report other two combinations. We present our results for Internet Explorer, PROD, PLUGIN and SERVER vulnerabilities. WINDOWS vulnerabilities are excluded because updating windows versions often results in a new WINE ID for the user, and therefore we are unable to trace a users' attack history throughout his/her Windows updates.

6.1.2 Analysis

We first give an overview of our dataset. Figure 6.1 shows a generalized regression ([61]) of attacked systems (left) and volume of attacks (right) as a function of time.[3] The shaded areas represent the 95% confidence intervals

[3] Regression generated by fitting to a generalized additive model (gam) of the form $g(E(Volume)) = s(Delta)$ where $g()$ is the link function of the expected volume of attacks ($E(Volume)$) and $s(Delta)$ is

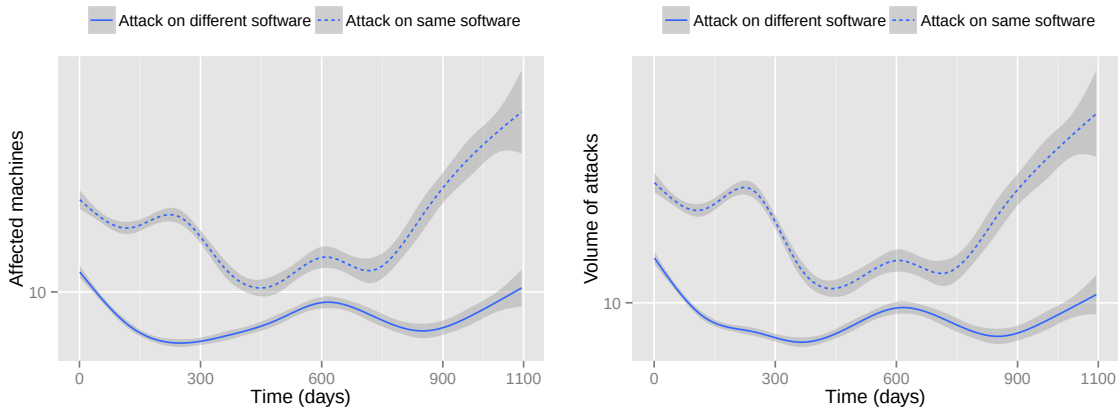


Figure 6.1: Regression of number of attacked machines (left) and volume of attacks (right) as a function of time. Attacks against the same software are represented by the dashed line; attacks against different software are represented by the solid line. Shaded areas represent 95% confidence intervals around the mean.

around the fitted line. Subsequent attacks directed towards the same software are represented by the dashed line. Subsequent attacks against different software are represented by the solid line. We observe that the distribution of attacked machines (left) follows closely the distribution of recorded attacks (right). In this study we will consider only the number of affected machines,

an unspecified smoothing function of the time between attacks ($Delta$). Note that the prediction power of our regression is likely very limited as it does not account for additional covariates of interest, such as geographical location, source of attack, or user type. This is because we are here only interested in a first, exploratory depiction of the relation between volume of attacks and time within our dataset. The goal of this is to pinpoint possible macro-differences in the trends of attacked machines and volume of attacks in time, *not* to predict future attack volumes. Thus, the model used *should not* be interpreted as an estimator of future trends of attacks, as this likely requires a more fine-grained analysis accounting for additional covariates. A more suitable model for this analysis could be of the form $Volume_t = \beta_0 + \beta_1(GEO_t) + \beta_2(USERTYPE_t) + \beta_3(ATTSOURCE_t) + \beta_4(Delta_t) + \mu_t$, where β_i are model parameters to be estimated, $GEO, USERTYPE, ATTSOURCE, Delta$ are the independent variables of the regression, and μ_t is the error term. Note that the model above likely suffers from some degree of heteroscedasticity, as the variance in volumes of attacks ($Var(Volume_t)$) likely depends on the same variables as its expected value ($E(Volume_t)$). This may be problematic in the estimation as the independence assumption on the error distribution, required for classic linear regression and generalised models, is not valid anymore [43]. Were any heteroscedastic effects present, adjustments to the model may be required in order to improve the efficiency of the estimator [126]. We keep this analysis for future work.

as this gives us a more direct measure of how many users are affected by a certain attack. We keep a closer analysis of volume of attacks for future work. We further observe that subsequent attacks against the same software are more frequent than subsequent attacks against different software. This is intuitive as the received attack depends on the software usage habits of the user. For example, a user that uses his/her system to navigate the Internet might be more prone in receiving attacks against Internet Explorer than against Microsoft Office. Because of this we will focus in this study on subsequent attacks against the same software. This will allow us to assess the attacker's attitude toward creating new exploits for the same software platform. We further observe that the fitted curves do not have a clear positive or negative slope as functions of time. This suggests that attacks are only weakly correlated with time, and other factors (such as users' patching attitudes, or just technological chances) may explain the trend.

Hypothesis 1a. To check the veracity of Hyp. 1a we evaluate how many users receive two attacks, after a certain δt, of either of these types:

1. A_1=$A(cve = cve'|\delta t \ \& \ sw = sw')$: Against the same vulnerability and same software version.
2. A_2=$A(cve < cve' \& \ vers \neq vers'|\delta t \ \& \ sw = sw')$: Against a new vulnerability and a different software version.
3. A_3=$A(cve < cve' \& \ vers = vers'|\delta t \ \& \ sw = sw')$: Against a new vulnerability and same software version.

In accordance with Hyp. 1a the attacker should prefer to (a) attack the same vulnerability multiple times, and (b) create a new exploit when he/she wants to attack a new software version. Therefore, according to Hyp. 1a we expect the following ordering in the data to be generally true: $A_3 < A_2 < A_1$. An exception may be represented by SERVER vulnerabilities: SERVER

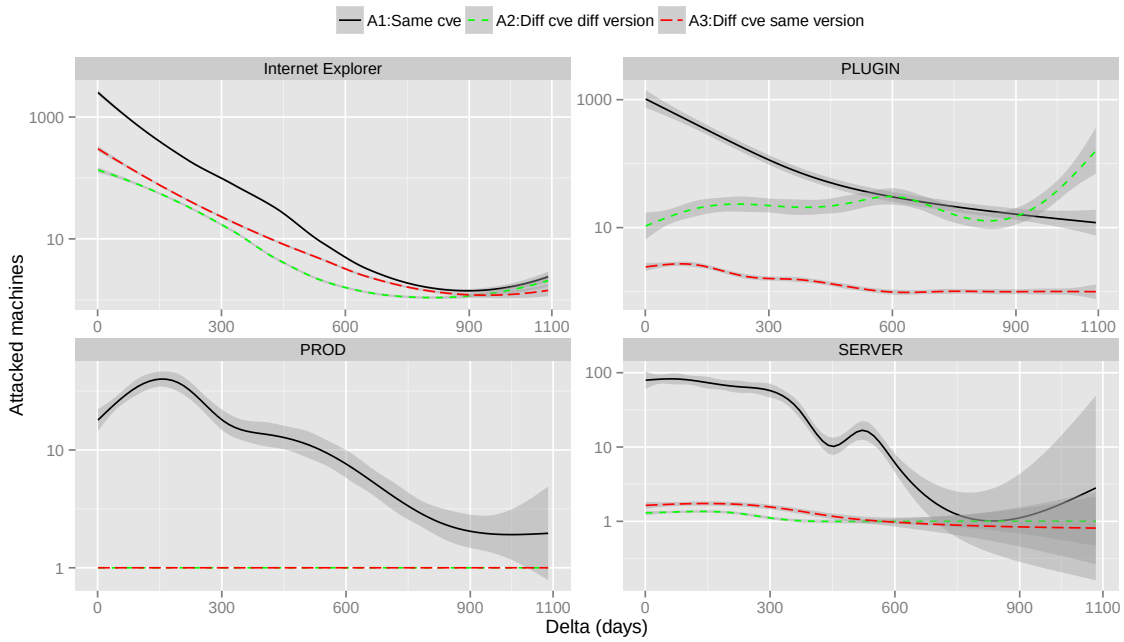


Figure 6.2: Targeted machines as a function of time for the three types of attack. A_1 is represented by a solid black line; A_2 by a long-dashed red line; A_3 by a dashed green line.

environments are typically better maintained than 'consumer' environments, which may affect an attacker's attitude toward developing new exploits. For example, SERVER software is often protected by perimetric defences such as firewalls or IDSs. This may require the attacker to engineer different attacks for the same software version in order to escape the additional mitigating controls in place. For this reason we expect the difference between A_2 and A_3 to be narrower or reversed for the SERVER category.

Figure 6.2 reports a fitted regression of targeted machines as a function of time by software category. As expected, A_1 dominates in all software types. The predicted order is valid for PLUGIN and PROD. For PROD software we find no new attacks against different software versions, therefore $A_2 = A_3 = 0$. This may be an effect of the typically low update rate of this type of software and relatively short timeframe considered in our dataset (3 years), or of a scarce attacker interest in this software type. Results for

SERVER are mixed as discussed above: the difference between A_2 and A_3 is very narrow and A_3 is higher than A_2: attackers forge more exploits per SERVER software version than for other types of software.

Internet Explorer. Internet Explorer is an interesting case in itself. Here, contrary to our prediction, A_3 is higher than A_2. By further investigating the data, we find that the reversed trend is explained by one single outlier tuple: *<CVE-2010-0806,CVE-2009-3672>*. Both these CVEs refer to vulnerabilities affecting Internet Explorer version 7. The two vulnerabilities have been disclosed 98 days apart, 22 days short of our 120 days threshold. More interestingly, these two vulnerabilities are very similar, as they both affect a memory corruption bug in Internet Explorer 7 that allows for an heap-spray attack that may result in arbitrary code execution[4]. Two observations are particularly interesting to make:

1. Heap spray attacks are unreliable attacks that may result in a significant drop in exploitation success. This is reflected in the "Access Complexity=Medium" assessment assigned to both vulnerabilities by the CVSS v2 framework. In our model, this is reflected in a lower $n(v,t)$ value, as the unreliable exploit may affect less machines than those that are vulnerable.

2. The exploitation code found on Exploit-DB[5] is essentially the same for these two vulnerabilities. The code for CVE-2010-0806 is effectively a rearrangement of the code for CVE-2009-3672, with different variable names. In our model, this would indicate that the cost $c(v)$ to build an exploit for the second vulnerability is negligible, as most of the exploitation code can be re-used from the old vulnerability.

[4]CVE-2009-3672: `http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2009-3672`
CVE-2010-0806: `http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2010-0806`

[5]CVE-2009-3672: `http://www.exploit-db.com/exploits/16547/`
CVE-2010-0806: `http://www.exploit-db.com/exploits/11683/`

Category	Test	Significance
Internet Explorer	$A_2 < A_1$	***
Internet Explorer	$A_3 < A_2$	***
PROD	$A_2 < A_1$	***
PROD	$A_3 < A_2$	-
PLUGIN	$A_2 < A_1$	***
PLUGIN	$A_3 < A_2$	***
SERVER	$A_2 < A_1$	***
SERVER	$A_3 < A_2$	

Table 6.2: Results for Hypothesis 1a. Significance (***) is reported for $p < 0.01$.

Having two independent but unreliable exploits that affect the same software version increases the chances of a successful attack, $n(v, t)$. Because the second exploit comes at a very low cost $c(v)$, the attacker chooses to exploit the second vulnerability as well as in this case the combination of the two exploits yields, by setting $c(v_2) = 0$:

$$c(v_1) < [n(t+\delta, v_1 \cup V) + n(t+\delta, v_2 \cup V)] \times R(I(v_1 \cup V)) - \sum_{i \neq \{1,2\}} E[U_{t+\delta, v_i}] \quad (6.6)$$

Eq. 6.6 shows that, at the cost of one exploit, the attacker gets the combined fraction of successful attacks[6] of both vulnerabilities. Moreover, Internet Explorer is used by a significant fraction of Internet users[7], therefore $n(t + \delta, v_1) + n(t + \delta, v_2)$ may be particularly interesting for the attacker.

Although this vulnerability is an exception in the data, the existence of the second exploit for Internet Explorer 7 is coherent with our model and ultimately supports our thesis that an attacker would build an exploit only if the additional cost is balanced by an increased rate of successful attacks over his/her current capability.

Table 6.2 reports the results of the analysis for Hyp. 1a with the exclusion of the Internet Explorer outlier, as discussed above. Significance is given by

[6]Note that because v_1 and v_2 are vulnerabilities of the same type, then $R(I(v_1 \cup V)) = R(I(v_2 \cup V))$.

[7]`http://www.w3counter.com/trends`

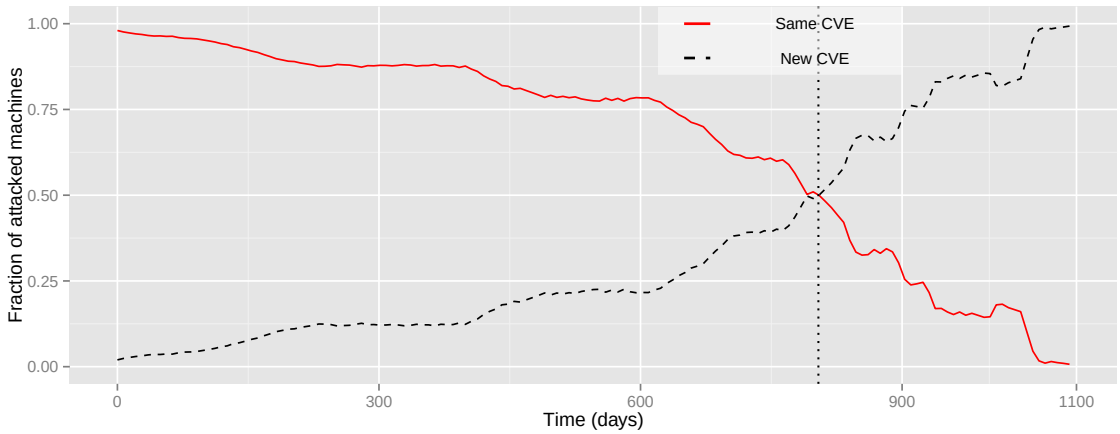


Figure 6.3: Fraction of systems receiving the same attack repeatedly in time (red, solid) compared to those receiving a second attack against a different vulnerability (black, dashed). The vertical line indicates the amount of days after the first attacks where it becomes more likely to receive an attack against a new vulnerability rather than against an old one.

a Wilcoxon paired test. All comparisons but SERVER accept the alternative that $A_2 < A_1$ and $A_3 < A_2$. Overall, we find strong statistical evidence supporting Hyp 1a.

Hypothesis 1b. We now check how the trends of attacks against a software change with time. Hyp. 1b states that the exploitation of the same vulnerability persists in time and decreases slowly at a pace depending on users' update behaviour. This is in contrast with other models in literature where new exploits arrive very quickly after the date of disclosure, and attacks increase following a steep curve ([20]).

Figure 6.3 reports the fraction of systems receiving, once an attack arrived, a subsequent attack against the same vulnerability (red, solid) as opposed to an attack against a different vulnerability (black, dashed). The x-axis reports the elapsed time since the first attack, in days. As hypothesised, the rate at which the same attack arrives decreases slowly with time and is still 20% after almost three years (1000 days). Notably, the event of receiving an attack

against a different vulnerability becomes more likely than its counterpart only 800 days (or 2 years, see dotted vertical line in Figure 6.3) after the first attack happens. This is interesting in itself as it indicates that attackers use the same exploit for a long period of time before substituting it at scale with a new one.

6.1.3 Robustness check

The distribution reported in Figure 6.3 depends on users' patching attitudes. In particular, according to the model presented here, software that is patched more often should see a quicker arrival rate of new exploits in time. We expect that software that is more rarely updated by users receives attacks against new vulnerabilities with a larger delay than software that is updated more often.

To the best of our knowledge there is no available data on the average rate at which users update different software types. However, as previously discussed, we expect SERVER software to be patched regularly ([95]), and to be generally maintained better than consumer software. Therefore, we expect the arrival of new exploits to be quicker for SERVER vulnerabilities than for other software types. This would also be coherent with the results in Table 6.2, as for SERVER $A_3 \geq A_2$ (i.e. the attacker does not wait for a new version to build a new exploit). We expect Internet Explorer to be fairly often updated as Microsoft releases patches every month and automatically pushes it to the users via the Microsoft Update system. PLUGIN software is traditionally seldom updated by the users, as only very recently a few PLUGIN vendors started pushing update notifications. Still, we expect PLUGIN exploits to arrive on average later in time than for other categories. As discussed previously, we have no data on subsequent attacks against PROD software affecting different vulnerabilities.

Figure 6.4 reports the distribution of days for the appearance of a new

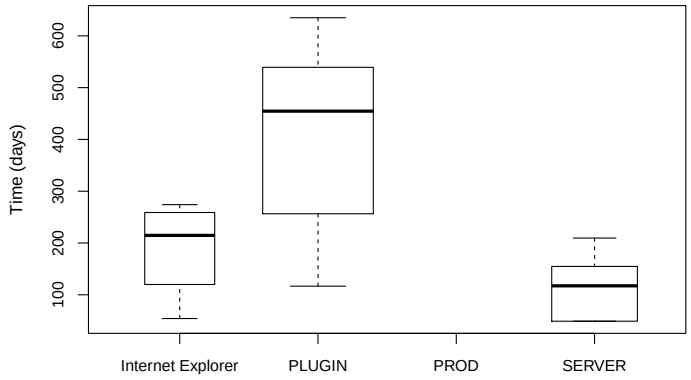


Figure 6.4: Distribution of average days between first exploit attempt and the appearance of an attack attempting to exploit a different vulnerability in the respective category.

attack for each software in the respective category. The delay for the appearance of a new exploit for PLUGIN software is the highest one ($p = 0.02$), with a median arrival delay of 454 days since first exploit. New exploits for Internet Explorer vulnerabilities arrive with a median delay of 214 days. SERVER attacks are the quickest to arrive, with a median delay of 117 days, but the difference with Internet Explorer is statistically significant for the alternative "SERVER exploits arrive faster than for Internet Explorer" at the 10% confidence level ($p = 0.08$).

6.1.4 Discussion

In this Section we discussed the *Model of the Work-Averse Attacker* as a new model to understand cyber threats. Our proposal is attacker-centric and models the attacker as a resource-limited actor that has to choose which vulnerabilities to exploit. We here only address the general case where the attacker aims at the 'mass of systems' in the wild In the 'general threat' case, the cost constraints emerging from the model prevent the attacker from

'exploiting all vulnerabilities' as otherwise currently assumed in academia and industry alike. We supported our claims with evidence from attacks recorded in the wild.

Evidence markedly points in the direction of the predictions our model makes. In particular, we find that:

1. An attacker massively deploys only one exploit per software version. The only exception we find is characterised by:
   - A very low cost to create an additional exploit, where it is sufficient to essentially copy and paste code from the old one, with little modifications, to obtain the new one.
   - An increased chance of delivering a successful attack.
2. The attacker deploys new exploits slowly in time; after three years the same exploits still drive about 20% of the attacks.
3. The speed of arrival of new exploits only weakly correlates with time, but shows a strong dependency on software patching rates.

Our findings suggest that the rationale behind vulnerability exploitation could be leveraged by defenders to deploy more efficient security countermeasures. For example, it is well known that software updates correspond to an increased risk of service disruption (e.g. for incompatibility problems or updated/deprecated libraries). However, if most of the risk for a particular software version comes from a specific vulnerability, than countermeasures other than patching may be more cost-efficient. For example, maintaining network IDS signatures may be in this case a better option than updating the software, because one IDS signature could get rid of the great majority of risk that characterises that system while a software patch may 'overdo it' by fixing more vulnerabilities than necessary.

Of course, the attacker may react to changing defenders' behaviour: in the game-theoretic view of the problem, the defender always moves first and therefore the attacker can adapt his/her strategy to overcome the defenders'. This is an unavoidable problem in security that is common to any threat mitigation strategy.

A more precise and data-grounded understanding of the attacker poses nonetheless a strategic advantage for the defender. For example, software diversification and code differentiation has already been proposed as a possible alternative to vulnerability mitigation ([36, 66]). By diversifying software the defender effectively decreases the fraction $n(t, v)$ of systems the attacker can compromise with one exploit. If the risk over a software version comes from only one vulnerability, than a possible counter-strategy to the attackers' adaptive behaviour is to first patch the high risk vulnerability, and then randomise the additional defences against the remaining vulnerabilities to minimize the attacker's chances of choosing the 'right' exploit to develop (as the attacker's multiple targets will likely choose a different set of vulnerabilities to patch). Diversifying defences may be in fact less onerous than re-compiling code bases (when possible) ([66]) or maintaining extremely diverse operational environments ([36]).

Conclusion 1 *From our analysis we find strong supporting evidence for Hypothesis 1. We therefore conclude that the attacker is rational and will as a result massively deploy exploits for only a subset of vulnerabilities.*

6.2 The Underground is a Sustainable Market Economy

Running Hypothesis	Hypotheses Testing
Hyp. 2. The underground markets are sound from an economic perspective.	**Hyp. 2**. Test Prop. 1 and Prop. 2. Develop a two-stage model of the underground markets to show that the underlying economic mechanism is sound.

In this section we test Hypothesis 2 to demonstrate that the cybercrime economy is sustainable from a market perspective. This section unfolds as follows: first, we analyse the mechanisms that are available to market participants to overcome market difficulties such as contract incompleteness (Proposition 1). This analysis is given in Section 6.2.1. We then proceed with analysing the quality of the technology traded in these markets (Proposition 2), in Section 6.2.2. Finally, we present in Section 6.2.3 a two-stage model of the markets that show, accounting for the discussion given in Sections 6.2.1 and 6.2.2, the sustainability of the market (Hyp. 2).

6.2.1 The Underground Markets are Mature

To investigate Proposition 1, we analyse two different cybercrime markets, Carders.de and HackMarket.ru, by comparing their regulating mechanisms and the effect those have on market effectiveness. The analysis results for HackMarket.ru are in sharp contrast with those of Carders.de and clearly show *prima-facie* evidence that underground cybercrime communities can be mature (and functioning) market .

The Carders.de market

This forum has a strict separation of trade related boards and non-trade related boards. Advertisement of (illegal) goods is permitted in the dedicated trading section. Members in this section are also allowed to request specific

Running Hypothesis	Hypotheses Testing
Hyp. 2. The underground markets are sound from an economic perspective.	**Prop. 1**. The underground markets evolved from a scam-for-scammer model to a mature state whereby fair trade is possible and incentivised by the enforced trading mechanisms. • **Prop. 1a**. Banned users have on average lower reputation than normal users. • **Prop. 1b**. Users with a higher status should on average have a higher reputation than lower status users. • **Prop. 1c**. Banned users who happened to have a higher status have a lower reputation than other users with the same status. • **Prop. 1d**. The ex-ante rules for assigning a user to a category are enforced. • **Prop. 1e**. There are ex-post rules for enforcing trades contemplating compensation or banning violators. • **Prop. 1f**. Users finalize their contracts in the private messages market. • **Prop. 1g**. Normal users receive more trade offers than known rippers do.

goods. The non-trade related boards serve the purpose of providing a discussion forum for the members where they can share thoughts, ask questions, publish tutorials and offer free goods on a specific subject. A third area of the forum, of little interest here, is dedicated to discussion of technical forum-related matters (e.g. maintenance). Carders.de allows both English and German speaking members on their forum. Figure 6.5 shows a schema of the two forum sections for English and German Speakers.

Since we are interested in the market characteristics of the forum, we exclude from the analysis users who have never participated in the trading sections. Further, the German-speaking part of the community is clearly the most developed one: the English section has 8% of all market posts while the remaining 92% are found in the German market. For this reason, we will

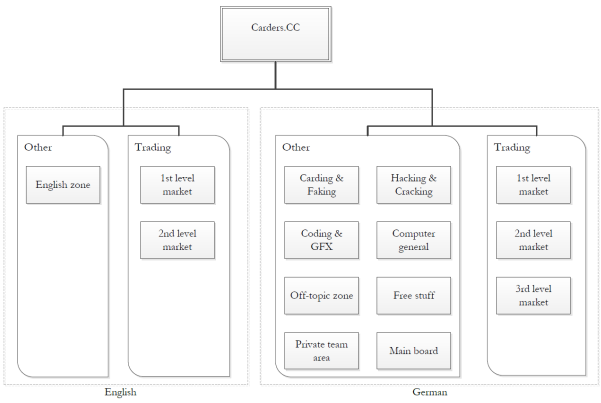


Figure 6.5: Categories of the Carders.de forum. The German market comprises more discussion sections and more market levels than the English market. Similarly, we found most of the activity to happen in the German section of Carders.de.

focus on the German market.

Users that join the community for selling or buying products are active in one of the market tiers within the forum. A user can advertise a product by creating a topic in the designated board in which this specific product falls.

In this newly created thread, other users discuss the product, ask questions and when a user shows interest as a potential buyer they contact the advertiser. According to the forum regulation, product trading should be finalized via private messages between the two parties.

Member roles

An important part of our study is to distinguish between different types of users. A user's status in the forum is also reflected by its membership in one of 12 user roles identified by the forum administrators. Table 6.3 shows these roles with the category to which they belong. The entry rank Newbie labels a newly registered user in the forum. After passing this role a newbie gets the role of normal user. Further up in the hierarchy, the user becomes a 2nd and 3rd tier user and have access to more specialized marketplaces. A verified vendor sells goods that are verified by the administrative team and

Table 6.3: Carders.de User roles

Role	Forum	Admins	Other
Newbie	×		
Normal user	×		
2nd Tier user	×		
3rd Tier user	×		
Verified Vendor	×		
Redaktion		×	
Moderator		×	
Global Moderator		×	
Administrator		×	
Scammers and banned			×

therefore ought to be more trusted by market participators. In contrast to other forum roles, a verified vendor does not require to climb up the rank ladder to achieve this entitlement.

Users with an administrative role manage, maintain and administer the forum. Members of the 'Redaktion' are editors of the forum. They publish news, events, regulation and other administrative information. The moderators maintain the forum and enforce regulation.

Administrative users are also responsible for banning users who have been reported for "ripping" other users in a transaction, or who have violated some internal rules.

Another important distinction to make is among banned users, which may have been excluded from the forum for a variety of reasons. Banned users are usually assigned an (arbitrary) string tag that describes the reason of the ban. By manual inspection we identified five categories of banned users: *Rippers, Double accounts, Spammers, Terms of Service violators* and an additional "Uncategorized" group for users banned without a reported reason. Table 6.4 shows the number of users for each group.

Each user in Carders.de can assign positive or negative *reputation points*

Table 6.4: Carders.de number of users per identified group

User group	no. users
Normal users	2468
Rippers	205
Double accounts	148
Spammers	42
ToS	5
Uncategorized	40
Total	**2908**

to other forum users. Higher reputation points should correspond to a higher "crowd-sourced trustworthiness" for the user. In the data there is no historical record of reputation points per users; we only have the reputation level at the moment of the dump. This prevents us from studying the evolution of a user's reputation level with time. For our stated hypothesis this is not necessary.

Carders.de's Regulation

The administrators of Carders.de published the guiding rules of the community in the regulation section. What follows is an overview of the regulatory structure of the community that will be central to our analysis as it identifies rules to access the trading areas of the forum and provides a principled distinction between "good" and "bad" users.

The forum regulation distinguishes three different *trading areas* (namely *Tiers*) in the forum, the access to which is constrained by increasingly selective sets of rules.

Tier 1 The lowest accessible tier is considered the public market on Carders.de. Newly registered users on the forum (Newbies, above) are not permitted to join the public market in Tier 1. the forum regulation state-

ment reports that users that have obtained the role of "normal user" can access this area. *Access rule: To become a normal user a newbie has to have posted at least 5 messages on the board.*

Tier 2 This market section is intended to be reserved to the 'elite' of the forum. More restrictive rules limit access to higher tiers. *Access rules: 1) Only users with at least 150 posts are allowed in Tier 2. 2) Users must have been registered to the forum for at least 4 months.*

Tier 3 This tier is an invitation-only section of the market. *Access rules: 1) The user has been selected by a team member of the forum to be granted access to Tier 3. 2) Access to Tier 2 is required.* This division clearly aims at creating 'elitist' sub-communities within the forum where the most reliable and active users participate. One would also assume that users of Tier 2 and 3 would be generally considered, in a working market, more trustworthy than users with Tier 1 only access. We however exclude Tier 3 from our analysis because it features only 5 users, including one administrator, and 17 posts. It is a negligible part of the overall market.

Carders.de Analysis

A failure of reputation mechanisms

To test our hypotheses we analyze reputation values for users in the Carders.de market. Figure 6.6 summarizes the distribution between banned and normal users, possibly accounting for the respective tiers. The data is on a logarithmic scale. The distribution of outliers suggests that reputation points make little sense with respect to user categories.

A Mann-Whitney unpaired test (chosen for its robustness to outliers and non-normality assumption) with null hypothesis "*The difference in reputation between banned and normal users is zero*" and alternative hypothesis "*banned users have higher reputation than normal users*" rejects the null ($p = 5.2e -$

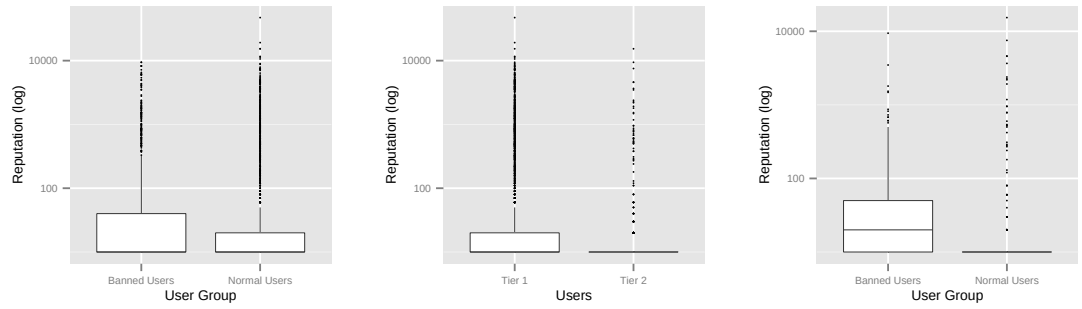


Figure 6.6: From left to right: 1) Reputation levels for normal users and banned users (whole market). 2) Users active in the tier 1 markets and tier 2 market. 3) Reputation of banned and normal users in tier 2. Banned users showed consistently higher reputation than normal users, even when considering only those active in the tier 2 market. The reputation mechanism is ineffective in both market sections.

15). We conclude that banned users have on average higher reputation than normal users. Proposition 1a is therefore rejected.

The Mann-Whitney test rejects the null "*Tier 1 and Tier 2 users have the same reputation distribution*" and accepts the alternative "*Tier 1 users have a higher reputation than Tier 2 users*" ($p = 4.8e - 06$). Hyp. 1b is rejected as well: reputation levels do not reflect membership in a "higher market level" and are effectively misleading.

Finally, we check whether reputation is at least a satisfactory indicator of user trustworthiness in Tier 2. It is not: Tier 2's normal users have on average a *lower* reputation than banned users. Hyp. 1c is rejected ($p = 4.9e - 16$).

All evidence suggests that the reputation mechanism in the forum did not work. We therefore exclude that reputation could have been a significant and useful instrument in the hands of the user to identify trustworthy trading partners. This also means that cheaters, or rippers, had no "fear" of having reputation points decreased by a disgruntled costumer, as reputation itself had no meaning whatsoever in the market. The only evidence is that it was used by bad users to inflate their own ratings.

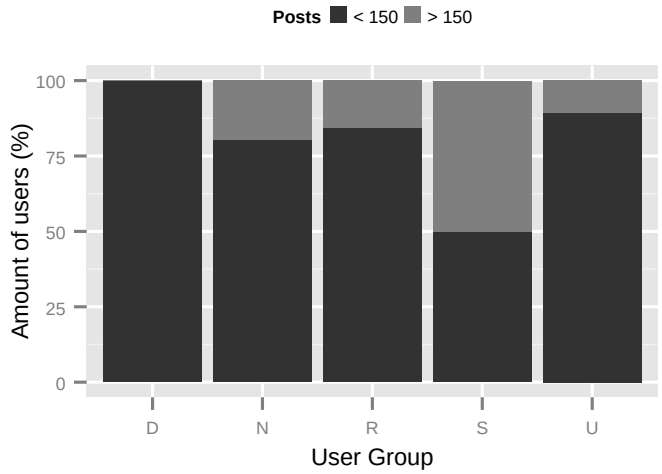


Figure 6.7: Users in tier 2 with more and less than 150 posts at the moment of their first post in tier 2. Most users had access to tier 2 before reaching the declared 150 posts threshold. D=Double accounts; N=Normal Users; R=Rippers; S=Spammers; U=Unidentified banned users.

A failure of regulations

Carders.de had no ex-post system of regulations (Hyp. 1e) and therefore we concentrate on the presence of ex-ante enforcement rules (Hyp. 1d). To test the validity of Proposition 1d we need to check each individual rule.

If rules are enforced in the first tier this would mean that no user with less than 5 posts is able to participate in Tier 1. We find that more than 50% of the users in Tier 1 accessed it before their fifth post in the community. Despite this being a very simple and straightforward rule to automate, there is no evidence of its implementation in the forum.

The first rule for access to Tier 2 states that users should have at least 150 posts before posting their first message in Tier 2. Figure 6.7 reports a breakdown of the posting history for each user category. The totality of users with *double accounts* posts in Tier 2 before reaching the 150 post limit threshold. This may suggest that users already familiar with the forum (e.g. previously

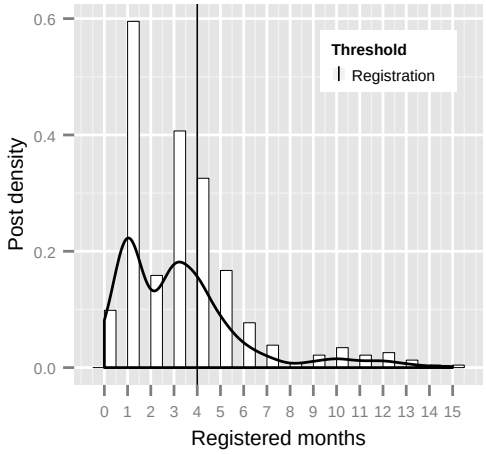


Figure 6.8: Time Distribution of Posts for Users in Tier 2. Most of the posting activity of users in Tier 2 happened well before they reached the required 4 months waiting period.

banned users) were accessing Tier 2 more quickly than others, possibly purposely exploiting the lack of controls. In general, the great majority of users in Tier 2 accessed it before the set limit of 150 posts.

Figure 6.8 shows a density plot of posts in Tier 2 along the months for which a user is registered to the forum. This also supports the previous conclusion that users had access to Tier 2 immediately when registered. Therefore we also reject Hyp. 1d.

Market existence . . . for rippers

Finally, we now measure the effects of these regulatory inefficiencies within the market. We first verify Proposition 1f. Given the unstructured nature of the data at hand, we proceed with a manual inspection of a sample of 50 randomly picked threads in the Private Message (PM) market and classify them as "trade related" or "not trade related". The goal is to understand whether the ratio of Private Message threads aimed at finalizing a trade supports Hyp. 1f or not.

Table 6.5 reports that almost 90% of the manually examined sample

threads are trade related. 54% of the trade-related PM threads also con-

Table 6.5: Classification of 50 Private Message Threads in Carders.de

Type	# Threads	
Trade Initiated	43	86%
Trade Initiated & Concluded	27	54%

Almost all threads in the PM section of Carders.de are about finalizing trades and more than half of them come to a close.

tained contact information between the parties (e.g. ICQ, Post Address and PayPal) and led to a concluding contract between the two. The evidence therefore supports Hyp. 1f: there has actually been a market.

We are now interested in seeing whether users that have been banned for explicitly *ripping* other users are more or less successful than normal users. Given the results we obtained so far, we expect the two types to be indistinguishable: if there is no available tool to distinguish between 'good' and 'bad' users (as the evidence indicates up to here), then choosing with whom to trade can be no better than randomly picking from the population of traders. Figure 6.9 is a boxplot representation of initiated trades for Rippers and Normal users in the forum. The two distributions overlap significantly. A Mann-Whitney test accepts the null hypothesis "There is no difference in the average number of received private messages for rippers and normal users" ($p = 0.98$). As expected in light of the evidence so far, the systematic failure of the forum mechanisms made rippers and normal users effectively indistinguishable to the trade initiator.

The comparison with HackMarket.ru

In this section we provide an introductory overview of the HackMarket.ru market which is still an active and arguably well-functioning cybercrime market.

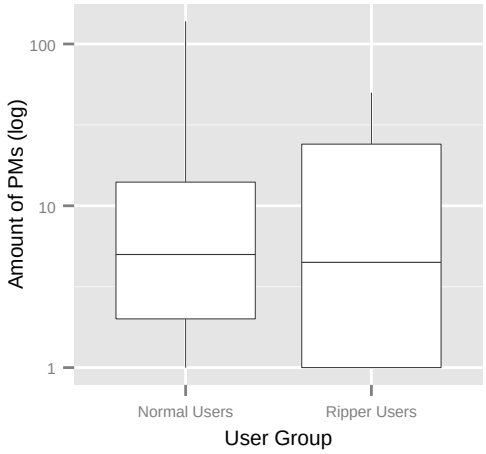


Figure 6.9: Initiated trades for Ripper users and Normal users. There is no difference in the number of trades the users of the two categories are involved in. Consistently with the analysis so far, this indicates that market participants are not able to distinguish good traders from bad traders.

A successful reputation mechanism

The forum regulation outlines seven user groups [DMN 5]. The following list presents these groups in descending order of trustworthiness, i.e. those on top of the list are the most reliable users in the community.

1. Admin.
2. Moderator.
3. Trustee: members of the community that *"own important services, or are moderators or administrators of other forums"* [DMN 5].
4. Specialist: Users elected in this group are considered "advanced" users" with a "high level of literacy".
5. User: Normal users.
6. Rippers: users that have been reported and have been found guilty of "scamming". It is explicitly recommended *"to have no deals (business,*

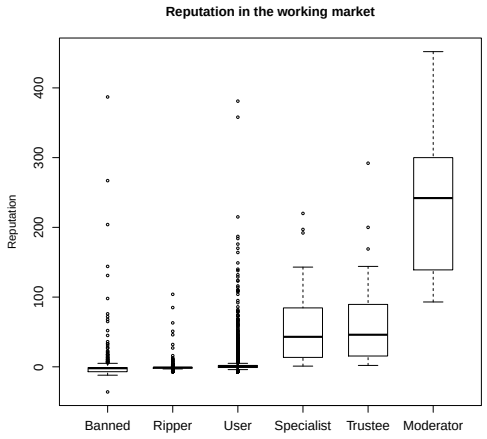


Figure 6.10: Boxplot representation of reputation distribution among categories. Reputation levels are statistically higher for higher categories when compared to reputation at lower categories. Only the categories Trustee and Specialist do not show statistical difference; these two are *elective* categories to which belong users deemed noteworthy by the administrator.

work) with users of this group" [DMN 5].

7. Banned: Users that have been precluded access to the forum.

Reputation points are attributed to users by other users after a positive or negative interaction between the two [DMN 6]. Of course, such system is subject to abuse; for example, a user may want to lower his competitors' reputation level to improve the competitiveness of their own business, or create fake accounts on the market to provide "collective" negative feedback. This adversarial behavior is limited by the mechanism's implementation rules: *"Only users with more than 30 posts can change reputation. Only 5 +/- reputation points per day can be assigned by any user to any other users."* [DMN 6]. This effectively places an upper bound in the number of reputation points one may assign in a given day and decreases one's influence over the overall distribution of reputation points in the market.

Figure 6.10 reports a boxplot representation of the distribution of reputation scores among user categories. Categories are listed in ascending order.

It is here clear that higher rankings are reflected in higher reputation levels of the users. We run a Mann-Whitney unpaired test to check if the difference in reputation levels between categories is significant, and we find that reputation levels significantly increase with higher categories. The only exception is for the Trustee and Specialist categories, for which no difference is found (which is explained by the elective nature of these categories). While this does not mean that higher reputation results in a higher ranking (as a number of endogenous factors other than reputation may be related to the inclusion in a user group - i.e. there is a self-selection problem), it does show that the reputation mechanism is effectively enforced and results in coherent distributions among users. For HackMarket.ru we accept Proposition 1a-1c.

Enforced ex-post regulations

Since there is no market hierarchy, Proposition 1d does not apply to HackMarket.ru. With regard to the ex-post type of regulations (Hyp 1e), users can effectively report other users to the board of administrators when they think they have been scammed. The administrators remark that *"We expose [cheaters] with pleasure."* [ADM 6]. The exposure of a user in the list of cheaters is a fairly refined process, that requires a report to be filed, an investigation to be carried, and that allows the 'alleged scammer' the right to defend himself before the decision by the moderators. The whole phase takes place in a dedicated sub-community of the market, a sort of 'court of justice' where the offended reports the (alleged, at this point) offender.

The reporting is to be filed according to a specific procedure established in the market regulation, that includes the *"name, contacts, a proof of the fact (log, screenshot of correspondence, money transfers,..) and a link to the user's profile."* Following the filing, an actual 'trial' takes place. The defendant has the obligation of replying to the accusation, as not doing so within seven days from the filing results in the accuser automatically winning

the case. The investigation can be carried both by moderators and administrators, while the final decision usually belongs to the administrator. The community is also often active in the discussion, reporting further evidence or personal experience with the accused, or helping in the investigations. An example of regulation during a trial is reported in the following, where the administrator is stating clearly the points of dispute:

Key issues, without which it would be impossible to objectively consider [to put the accused in the] Black [list of scammers]:

1. *Whether the transfer happened at all*
2. *Whether the transfer was cashed*
3. *Exactly who received/took off with the money. [DMN 1]*

A key point is to understand how the punishment mechanism is applied in practice. In particular, we are interested in understanding whether trials unfold with significant discussions, and whether the final decision is ultimately enforced.

To this aim, in Table 6.6 we illustrate three example trials held in the market, two of which ended with a user being 'black listed', and one where the accused is acquitted and no punishment is imposed. We define 'accuser' the user that reports the complaint, and 'defender' the reported user.

Table 6.6: Enforcement of regulation mechanisms in HackMarket.ru.

Case	Challenged amount	#Users involved	Evidence	#Messages	Duration	Outcome	Reason
Defender no show	390$	7	Chat transcripts	11	7 days	Defender banned	Defender never showed up.
Defender loses	2800$	7	Screenshots, transaction logs, chat transcripts.	29	29 days	Defender banned.	Defender did not provide exhaustive evidence that the payment was ultimately committed in favor of the accuser.
Defender wins	1400$	3	Chat transcripts, screenshots.	9	11 days	Defender found not guilty, no action taken.	The defender demonstrated that good was not delivered because the payment happened during a technical malfunction of his Internet connection, and he therefore could not acknowledge it.

Trial regulation is strictly enforced. Evidence brought in support to the case of either the defender or the accuser is always critically analyzed; more controversial trials require longer time to be concluded, and the final decision can be in favor of either participant, depending on how convincing the evidence supporting one's case was.

All three cases were filed by disgruntled clients who paid the sellers but did not receive the goods. All trials above took place within an observation year. In every case, the HackMarket.ru community joins in into the investigation, either providing additional details on the current status of the users involved in the case, or as witnesses with past experience in dealing with the accuser or the defender. As expected, controversial cases take more time than easier ones. In Table 6.6, the first case is quickly closed as simply the defender does not show up in time. This complies with the forum regulation noted above. The second case is the most controversial of the three, with the defender aggressively participating in the discussion and providing more and more (unsatisfactory) evidence of his innocence. The amount of evidence provided, and the intricacy of the discussion require time for the administrator to come to a verdict, which happens after a month. In the third case, the defender was able to show that he never "cashed" the sent payment. The accuser stops replying soon after that and the administrator closes the case.

Evidence is carefully analyzed by the forum administrator as the following excerpt shows:
Judging from the screen from post #num, there is a transfer, and it was received. Double-check that, you can verify online with Western. But I haven't seen proof of receipt. To get the answer for the third question, we need to ask to whom the money was sent through Western. If I am not mistaken, upon request of the sender they can provide full information.
Therefore, we will do as follows. Sender, i.e. #buyer nickname get all details and full information from Western, report here the result before Friday #date.[DMN 1]

In some cases, the administrator tries to arbitrate the question as s/he clearly values both buyer and seller: *It would be great if you two [buyer and seller] contact each other and sort this matter out. We only need to know the details for the recipient, and it will immediately be clear who is at fault,*

even without [proceeding with] the Black [list]. [DMN 1]

On a qualitative note we observed what follows:

1. the defender always reports detailed information on the accused user and on the case of complaint.;

2. many witnesses appear in 'court' giving opinions on the evolution of the case, or providing supporting evidence for either the accuser and the defender;

3. the moderators and the administrators are always present in each report, and actively moderate the discussion;

4. when the defender does not show up within the time limit specified by the administrator [DMN 6], the case always goes to the defender;

5. when the defender shows up, he/she always publishes evidence of his/her case, being those screenshots of chats with the accuser or Webmoney transaction logs;

6. some cases last several months, with all parties actively participating in the discussion and new evidence being examined or asked for iteratively;

7. when the evidence provided by either of the defender or the accuser is not conclusive, the case goes to the opponent or a 'null' is thrown (when neither of the two is convincing, nobody wins);

8. users that end up being found guilty are *always* exposed in the list of cheaters and/or are banned from the forum. The latter is a harsh punishment: in contrast to IRC markets,re-entry into the forum is neither easy in effort nor short in time.

We therefore accept Proposition 1e for HackMarket.ru.

Market existence

We have not direct access to the private conversation of participants in HackMarket.ru, but we collected exhaustive evidence on their private transactions through the conversation logs reported in the trials. In every case reported, the finalization of the contract and the transaction always happen through some type of private communication, usually thought the ICQ chat messaging system or Jabber.ru. We therefore accept Proposition 1f.

Participants initiating a trade also often declare to have performed a background check on the seller by either contacting the administrators or by checking the official blacklist of the forum. One example of this is given in [NTL 12]: "*[The] admin [of the forum] confirmed me that you [the seller] are not a rookie trader*". Evidence for background checks such as this is frequent. We therefore accept Proposition 1g.

Discussion

"Regulation" is the main advantage that a forum-based community has over an IRC-based community: it provides the forum users with a set of rules and mechanisms to assess the information they can collect on a particular trade. The analyzed markets attempted to enforce this by providing a regulatory mechanism for user reputation and access to "elite" market tiers. This may be not sufficient for the user to have complete information on the transaction; yet, it could provide her with some baseline information on her trading partner, ruling out part of the *information asymmetry* problem identified for other markets [63], and precisely by mitigating the *adverse selection* problem [48]. For legitimate markets, reputation proved to be an effective mechanism albeit not a definitive solution.

Table 6.7 reports the summary of Hypothesis testing for the two markets. The organizational and structural differences of HackMarket.ru with respect

to Carders.de is evident. In Carders.de, each of the regulation mechanisms has been faultily implemented and the potential means for a user to assess ex-ante a trade are pointless or even misleading. The systematic failure of the regulatory mechanisms clearly led to a market were users had no incentives in conducting fair transactions and had no means to distinguish "good traders" from "bad traders". We showed that there is in fact no difference in the number of trades initiated with a ripper and trades initiated with a normal user. This effect alone may have brought to the failure of the market, which we show being effectively of the same nature of Florêncio et al.'s IRC market.

In HackMarket.ru the reputation and punishment mechanisms generate meaningful information for the user:

1. Evidence supports the hypothesis that reputation points are meaningfully assigned to users and this arguably results in a useful tool for the user to asses potential trading partners.

2. The punishment mechanism is a well-regulated one and direct evidence suggests that 'trials' are conducted in a fair manner. This boost market activity and incentivizes 'honest' behavior.

3. Users that have been found guilty are, if not banned, publicly exposed and assigned to the 'scammers' group. This allows other users to clearly assess a scammer's trading history and make an informed decision with whom to trade.

It appears that the punishment mechanism is enforced coherently with the stated rules (e.g. the time frame for the defendant to show up is firmly enforced). We find evidence that trials in the market involve an in-depth discussion on the issue raised by the accuser, and witnesses are called to support one's claims. Importantly, evidence supporting the case of both the defender and the accuser (e.g. transaction logs and previous exchanges between the two parties) is always requested and analyzed. This shows that

the forum administrators tend to take well-informed decisions. This is in accordance with the overall reputation levels among categories (Figure 6.10).

The very fact that defendants do show up is a proof that they see a value in preserving their reputation as users and do not just register with a new account. The difficulties of the registration process makes dropping and re-registering a costly and lengthy process.

Conclusion 2 *From our analysis we therefore accept Proposition 1. We conclude that the cybercrime markets evolved from an unsustainable model to one where strong regulation and reputation mechanisms may allow market participants to overcome the asymmetry problems inherent in this setting.*

Table 6.7: Comparison of results for Carders.de and HackMarket.ru.

Proposition	Description	Tested Prop. #	Carders.de	HackMarket.ru
Reputation mechanisms work	Banned users have lower reputation than normal users.	Prop. 1a	Rejected	Verified
	Higher status users have a higher reputation than lower status users	Prop. 1b	Rejected	N.A.
	Banned users with a higher status have a lower reputation than other users with the same status	Prop. 1c	Rejected	N.A.
Regulations are enforced	Preventive (ex-ante) rules are enforced	Prop. 1d	Rejected	N.A.
	Punishment (ex-post) rules are enforced	Prop. 1e	N.A.	Verified
The market works	Users privately finalize their contracts	Prop. 1f	Verified	Verified
	Normal users receive more trade offers than known rippers	Prop. 1g	Rejected	Verified

Hypotheses aimed at assessing the reliability of the reputation mechanism, the enforcement of regulation, and market fairness are all rejected for Carders.de. In contrast, HackMarket.ru appears to be a well-functioning market.

6.2.2 The Technology Traded in the Underground is Effective

Running Hypothesis	Hypotheses Testing
Hyp. 2	**Prop. 2**. The tools bought and used by the attackers are well engineered products that are effective when deployed in the wild, as tested in the *MalwareLab* against evolving software configurations.

To investigate Proposition 2 we test 10 exploit kits leaked from the underground markets to investigate their efficacy in the wild. In particular we test whether they are resilient to the changing operative environment in the wild (i.e. updating software configurations), or if they are effective only for small windows of time.

Exploit kits' main purpose is to silently download and execute malware on the victim machine by taking advantage of browser or plugin vulnerabilities. Errors in applied programming interfaces or memory corruption based vulnerabilities allow an exploit to inject a set of instructions (shellcode) into the target process. Shellcode on its turn downloads an executable malware on the victim's hard drive and executes it. The executable installed on the target system is completely independent from the exploit pack (see [58] for some statistics on the pairings).

Figure 6.11 depicts the generic scenario of drive-by-download attack [58, 75]. A victim visits a compromised web site, from which he/she gets redirected to the exploit kit page. Various ways of redirection are possible: an <iframe> tag, a JavaScript based page redirect etc. The malicious web page then returns an HTML document, containing exploits, which are usually hidden in an obfuscated JavaScript code. If at least one exploit succeeds, then the victim gets infected. An exploitation is successful when the injected shellcode successfully downloads and execute a malicious program on the victim system.

These tools are advertised and traded in the black markets. An example

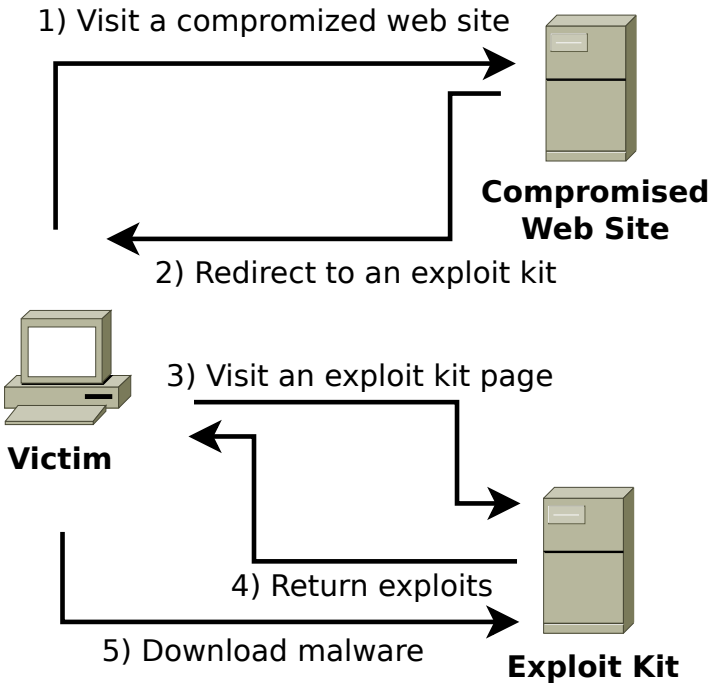


Figure 6.11: Scheme of drive-by-download attack

of such advertisement is given in Figure 6.12. In this ad are reported the vulnerabilities included in the kit and the expected success rate of about 20%. We find similar success rates to be declared in the advertisement of the competition as well.

Design of the experiment

To evaluate exploit kit resiliency, we test exploit kits in a controlled environment, our *MalwareLab*. The core of our design is the generation of "reasonable" *home-system* configurations to test against the infection mechanism and capabilities of exploit kits. We test those configurations as running on Windows XP, Windows Vista and Windows 7. Table 6.8 reports versions and release dates of each operating system and service pack considered (from here on, *system*). After an initial phase of application testing on the selected systems, we fix the life-time of an operating system to be 6 years for com-

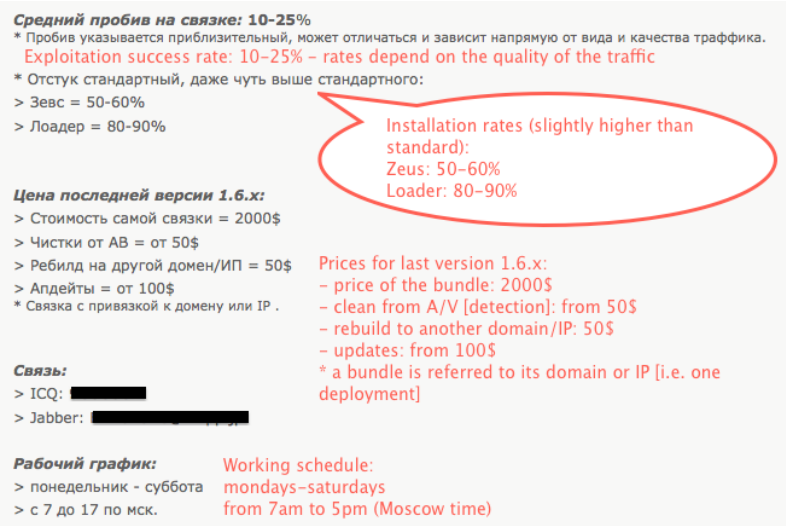


Figure 6.12: Sample advertisment for a popular exploit kit in 2011- mid 2012, "Eleonore".

patibility of software. Y_{sys} indicates the working interval of each operating system.

For our experiment we selected 10 exploit kits (see Table 6.9) out of the 34, leaked from the black markets, we gathered. Some of them proved to be not fully-functional or impossible to be deployed (e.g. because of missing functions). Out of those that were deployable and armed, we selected 10 according to the following criteria: (a) popularity of the exploit kit [112]; (b) year of release; (c) unique functionality (e.g. only one of multiple versions of the same kit family is selected).

Configuration selection

The automated installation of software configurations on each machine followed the definition of a criteria to select software to be installed. As often happens, this is subject to a number of assumptions that define the criteria themselves. For our experiment to be realistic, we need to build configurations that are *reasonable* to exist at a certain point in time. As an example, we consider unlikely to have Firefox 12, released in April 2012, installed on

Table 6.8: Operating systems and respective release date. Configurations are right-censored with respect to the 6 years time window.

Op. system	Service Pack	Y_{sys}
Windows Xp	None	2001 - 2007
	1	2002 - 2008
	2	2004 - 2010
	3	2008 - 2013*
Windows Vista	None	2006 - 2012
	1	2008 - 2013*
	2	2008 - 2013*
Windows 7	None	2009 - 2013*
	1	2011 - 2013*

the same machine with Adobe Flash 9, released 6 years earlier in June 2006. We therefore fix a two-years window that defines which software can coexist. The window is based on the month and year of release of a particular software. Since our oldest exploit kit is from early 2007, we are testing software only released in the interval $(2005, 2013)$. Table 6.10 shows the software versions we consider[8].

The algorithm to generate each configuration iterates through all years Y_{conf} from 2006 to 2013, and chooses at random a version of each software (including "no version", meaning that that software is not installed for that configuration) that satisfy $Y_{swRel} \in [Y_{conf} - 1, Y_{conf}]$. For each Y_{conf} we generate 30 random configurations. Given the construction of Y_{swRel}, we end up with seven windows and therefore 210 configurations per system reported in Table 6.8. However for compatibility reasons each system has a time window of 6 years starting one year before its release date. Because we want to measure the resiliency of exploit kits, we keep the number of configura-

[8]We did not include Google Chrome as it was first released halfway through the timeline considered in our experiment (2008). Introducing Chrome samples in 2008 would have changed the probability of a particular software to be selected. In turn, this would make comparing time windows before and after 2008 statistically biased. We plan to include Chrome in future experiment designs.

Table 6.9: List of tested exploit kits

#	Name	Version	Release Year
1	Crimepack	3.1.3	2010
2	Eleonore	1.4.4mod	2011
3	Bleeding Life	2	2010
4	Elfiesta	1.8	2008*
5	Shaman's Dream	2	2009*
6	Gpack	UNK	2008
7	Seo	UNK	2010
8	Mpack	0.86	2007*
9	Icepack	platinum	2007
10	Adpack	UNK	2007*

For some exploit kits we could not find the respective release advertisement on the black markets, and therefore a precise date of release for the product cannot be assessed. For those (*) we approximate the release date to the earliest mention of that exploit kit in underground discussion forums and security reports. This identifies an upper bound of the release date.

tions per year constant (otherwise results would not be comparable between different runs). This means that some systems are tested, overall, against a lower number of configurations than others. For example, Windows XP Service Pack 1 (2002-2008) will be tested only against configurations in the time windows *{[2006, 2008),[2007-2009))}*[9], which gives us 60 configurations. Windows Vista with no Service Pack (2006-2012) will instead be tested, for the same reason, with 180 configurations. This guarantees that each exploit kit is tested for each system against the same number of configurations per year.

The algorithm iterates through each configuration and runs it against the available exploit kits. Figure 6.13 is a representation of an experiment run for each system. At each iteration i, we select the configuration $conf_i$. If

[9]Note that the last year of the time window is not included. For example, [2006,2008) includes configurations from January 2006 to December 2007a.

Table 6.10: Software versions included in the experiment.

Software	Versions	# of versions
Mozilla Firefox	1.5.0.2 - 17.0.1.0	122
Microsoft Internet Explorer	6-10	5
Adobe Flash	9.0.16.0-11.5.502.135	54
Adobe Reader	8.0.0-10.1.4	17
Java	1.5.0.7-7.10.0.0	49
	Total	247

Overall 9 software versions were excluded from the experiment setup because the corresponding installation package was either not working or we could not find it on the web.

$Y_{conf_i} \in Y_{sys}$, we automatically install the selected software on the virtual machine using the "silent install" interface provided by the vendor or by the *msi* installer. A configuration install is successful when *all* software in that configuration is installed.

When the installation process ends, we take a "snapshot" of the virtual machine. Every run for $conf_i$ will restore and use this snapshot. The advantages of this are twofold: at first we eliminate possible confounding factors stemming from slightly different configurations, because only the exploit kit changes; secondly, this is also faster than re-installing the configuration every time, which would have considerably stretched the (already not short) completion time. When all exploit kits are tested, a new configuration is eligible for selection.

Data collection

In the course of our experiment we keep track of (a) the successfulness of the automated installation of a configuration on a victim machine (VICTIM) at any given time; (b) the successfulness of infection attempts from exploit kits. This data is stored in two separate tables, *Configurations* and *Infections* respectively.

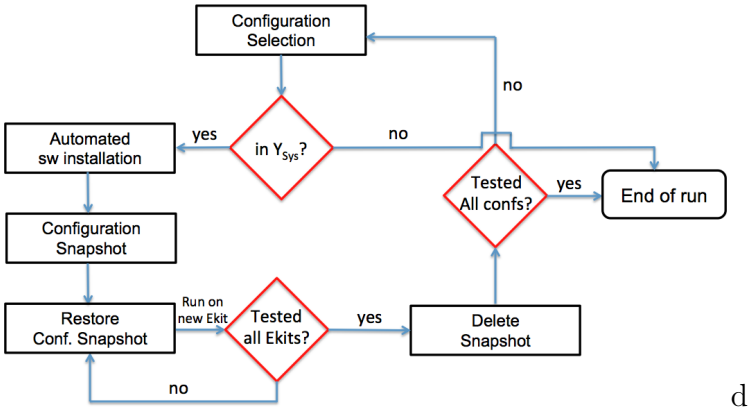


Figure 6.13: Flowchart of an experimet run. This flowchart describes a full experiment run for each system in Table 6.8. Configurations are generated in chronological order, therefore if the first control on Y_{Sys} fails, every other successive configuration would as well and the experiment ends. Snapshots enable us to re-use an identical installation of a configuration multiple times.

1. *Configurations* is needed to control for VICTIM configurations that were *not* successfully installed; this way we can correctly attribute (un)successful exploitation to the right set-ups. This is desirable when looking for infection rates of single configurations or software.

2. *Infections* stores information on each particular configuration run against an exploit kit. We set our infection mechanism to make a call to the Malware Distribution Server (MDS) each time it is executed on the VICTIM machine. A "call back" to the MDS can in fact only happen if the "malware" is successfully executed on VICTIM. The MDS stores the record in *Infections*, alongside *(snapshot_id, toolkit_name, toolkit_version, machine, IP, date, successful)*. Exploit kits have an "administrative panel" reporting infection rates [75]. However, we decide to implement our own mechanism because (a) it allows us to have more control on the data in case of errors or unforeseen circumstances; (b) exploit kits statistics may not be reliable (e.g. developers might be incentivated in exaggerating infection rates).

To minimise detection [58], some exploit kits avoid attacking the same

machine twice (i.e. delivering the attack the same IP). This behaviour is enabled by an internal database controlled by the kit, independent from our *Infections* table. In some cases, e.g. when the experiment run needs to be resumed from a certain configuration, our *Infections* table may report un-successful attacks of an exploit kit, when instead the exploit kit did not deliberately deliver the attack in the first place. We therefore need to control for this possibility by resetting the exploit kit statistics when needed.

Operational realization

In this Section we present the technical implementation of our experiment design in its three key points: (1) virtualised system infrastructure; (2) automated execution; (3) operative data collection;

Virtualised System Infrastructure

When testing for malware, an isolated, virtualised infrastructure is desirable [101]. We set up a five machine network that includes a Malware Distribution Server (MDS) and four machines hosting the Victim Virtual Machines (VICTIMs). Initially, the setup also included an IDS and a network auditing infrastructure to log the traffic; however, to eliminate possible confounding factors caused by the network monitoring and auditing, we decided to eliminate this part of the infrastructure from the design reported here. For practical purposes (i.e. scripting), all machines are run on a linux-based operating systems, upon which the virtualised infrastructure is installed.

The purpose of the MDS is to deliver the attacks. Because of the nature of exploit kits, all we need to attack VICTIMs is an Apache Web-Server listening on HTTP port 80 upon which the kits are deployed. As mentioned, we implemented and armed the exploit kits with our own "malware", Casper.exe (our *Ghost-in-the-browser* [94]) to help us keep track of infected systems. In order to make it compatible with all Windows versions we have linked

it statically with the appropiate libraries (e.g. *Winsock*). Casper reads a special configuration information file that we put on each victim machine and send its content to a PHP script on the MDS by using the *Winsock* API. This script (trojan.php) simply stores the received data along with the VICTIM IP address and timestamp into the *Infections* table in our database.

Automated execution

We use VirtualBox to virtualise victim machines. In order to automate the tests we take advantage of the tool that is shipped with VirtualBox called VBoxManage. It is a command line tool that provides all the necessary functions to start/stop virtual machines, create/delete snapshots and run commands in the guest operating system. The main program, responsible for running the experiment is a Python script that makes a sequence of calls to VBoxMange via subprocess Python module.[10]

At each run, our scripts read configurations.csv, a file containing all the generated configurations for that machine. The scripts iteratively install configurations upon the VICTIM system. The mapping between software version pointers in configurations.csv and the actual software to be installed is hard-coded in the core of the implementation. The automated installation happens via the silent install interface bundled in the installation packages distributed by most software vendors. However, because of a lack of a "standard" interface and the inconsistencies between different versions of the same software, we could not deploy one-solution for all software. We used instead a "trial-and-error" approach and online documentation to enumerate the arguments to pass to the installers and map them with the right software versions. Each configuration is then automatically and iteratively run against every exploit

[10]It should be noted that there is Python API for VirtualBox, that allows to run VirtualBox commands directly from within the Python environment. We tried to use it during our first (failed) experiment, but had to switch to VBoxManage, because Python VirtualBox API functions proved not to be very reliable on our machines.

kit on the MDS.

Despite the experiment being completely automated, we found that some machines were failing at certain points in the run, most often while saving snapshots or uploading files to the VICTIMs. We therefore implemented a "resume functionality" that allows us to "save" the experiment at the latest valid configuration, and in case of failure restore the run from that point.

To reset exploit kits statistics and guarantee the soundness of the statistics collected in the *Configuration* and *Infections* tables, we have implemented a PHP script that clears the records on delivered attacks the kit keeps. This step was rather easy to accomplish: we used the code snippets responsible for statistics reset in each exploit kit, and copy-pasted them into a single script.

We keep track of software installations on the VICTIM machines by means of a second dedicated script. To build it, we manually checked where each program puts its data on the file system at the installation. Because it was impossible to look at every application installation directory we sampled a subset of programs to check whether they always put data in the same place. Then we wrote a batch file that checks for the presence of the corresponding data directories *after* the alleged installation. The results of the batch file inspection are then passed to a Python script on the host machine, sent to the MDS, and stored in the *Configurations* table on our dataset.

To collect the infection data, when the MDS receives a call from a VICTIM machine, the MDS adds a record in the *Infections* table, setting the *successful* record to 0 (the default). When executed, *Casper* connects to the MDS via a PHP page we set up (namely infection.php). This updates the *successful* bit of the corresponding run record in *Infections* to 1.

Experiment results

The automatic installation procedure proved to be rather reliable. Figure 6.14 depicts a 100%-stacked barplot of configuration installs by software. As

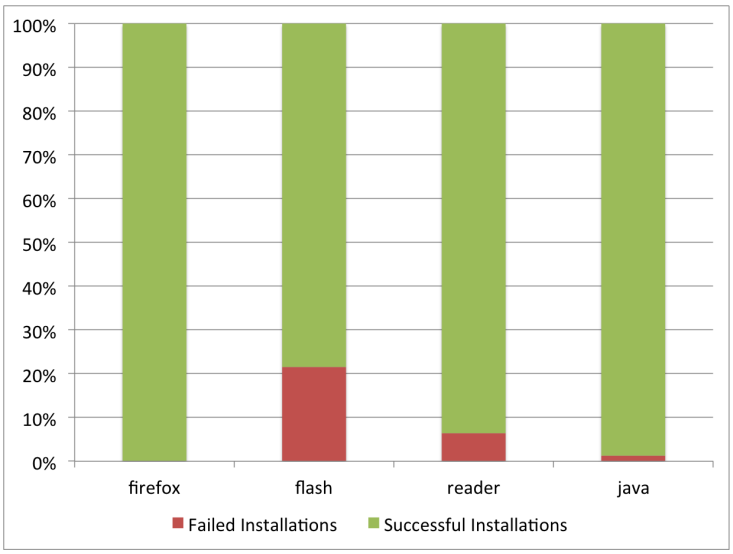


Figure 6.14: Stacked barplot of configuration installs by software. The installation procedure was successful the majority of the time, the only exception being Flash for which we have a 20% detected failure rate.

one can see, Firefox and Java were practically always successfully deployed on the machine. In contrast, 6% of Adobe Acrobat and 21% of Flash installations were reported to be not successfully completed. However, it proved practically unfeasible to manually check failures of our detection mechanism (e.g. the files for that software version on that configuration may be on a different location). We cannot therefore assess the level of false negatives our detection mechanism generates.

Figure 6.15 reports an overview of the infection rates of *all exploit kits* in each time window. Intuitively, because the exploit kits are always the same, the general rate of infection decreases with more up-to-date software. Observationally, from 2005 up to 2009 the success rate of exploit kits seem not to be affected by system evolution. A marked decrease in the performance of our exploit kits starts only after 2010. This observation is confirmed by looking at a break-up of volumes of infections per exploit kit per year, depicted in Figure 6.16. Generally speaking, each exploit kit (apart from Bleeding Life) seem to remain effective mainly within the first three time

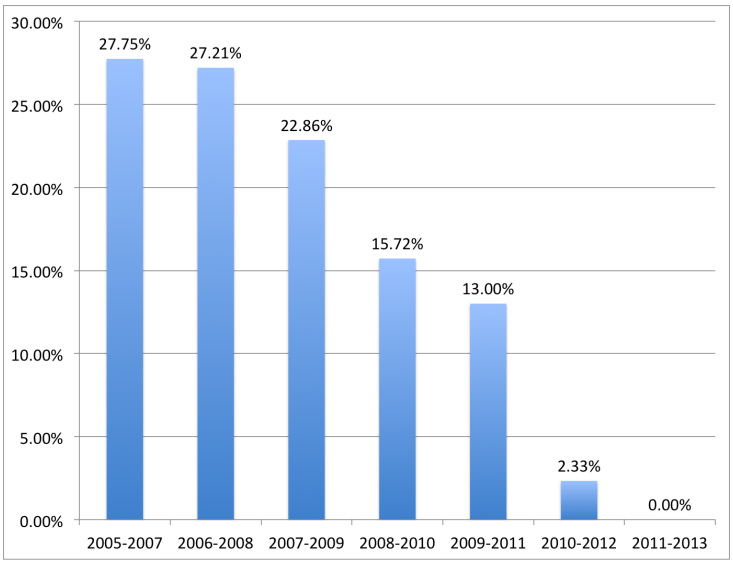


Figure 6.15: Infection rates per time window. Exploit kits obtain a peak of about 30% successful infections and maintain this level for 3 years on average. Afterwards infection rates drop significantly. Only after 8 years overall exploitation rate goes to zero.

windows, from 2005 to 2009. Eleonore, CrimePack and Shaman lead the volume of infections in those years, with Eleonore peaking at more than 100 infections for 2006-2008, which amounts at about 50% of the configurations for that window. Interestingly, a few exploit kits seem identical in terms of performance. Seo, mPack, gPack, ElFiesta, AdPack, IcePack all perform identically throughout the experiment. Most exploit kits's efficacy drops in the fourth time-window, were configurations spanning from 2008 to 2010 are attacked. However, Bleeding Life is here an outliner, as its efficacy in infecting these machines rises and tops in 2009-2011 to more than twice its infection rates for 2005-2009. After 2011, however, its infection capabilities drop to zero. In the last but one time window (2010-2012), the only still effective exploit kits are Crimepack and Shaman. Overall three types of exploit kits seem to emerge:

1. *Lousy exploit kits.* Some exploit kits in the markets seem to be identical in terms of effectiveness in infecting machines. Not only they perform

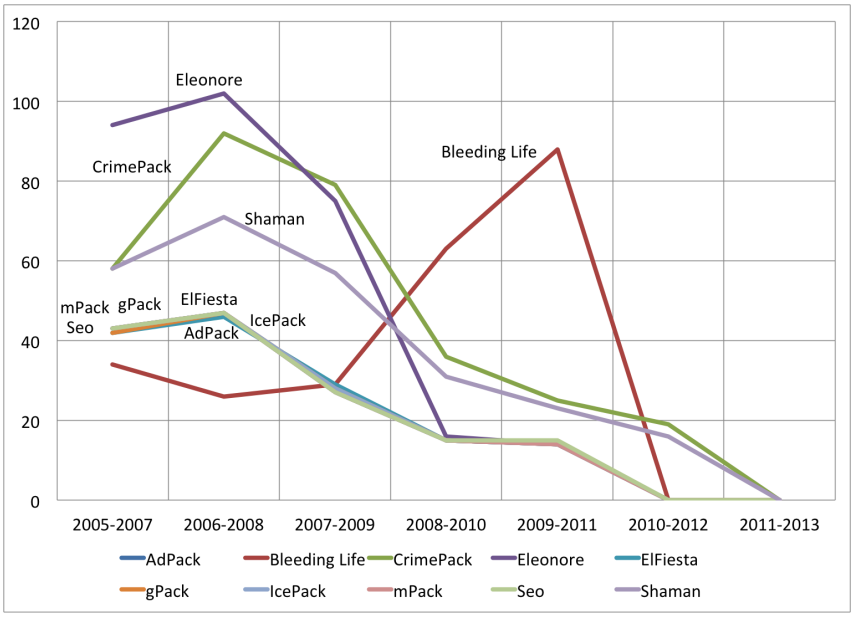


Figure 6.16: Number of configurations that each exploit kit was able to successfully attack in each time window. Number of exploited configurations are reported on the Y-axis, and time windows on the X-axis. We can identify three groups of exploit kits. *Lousy* kits (mpack, Seo, ElFiesta, AdPack, IcePack, gPack) are rip-off of each other and perform precisely the same and are consistently the worst. *Long-term* exploit kits (Crimepack, Shaman) achieve higher exploitation rate and maintain non- zero exploitation rates for up to 7 years. *Time-specific* exploit kits (Eleonore, Bleeding Life) achieve the highest exploitation rates within a particular time frame but their success rate drops quickly afterwards.

equally, but the identical trend throughout our experiment suggests that the exploits they bundle are themselves identical. This may indicate that some exploit kits may be rip-offs of others, or that an exploit kit author may re-brand the same product.

2. *Long-term exploit kits.* From our results, a subset of exploit kits (in our case Crimepack and Shaman) perform particularly well in terms of resiliency. Crimepack and Shaman are the only two exploit kits that remain active from 2005 to 2012, despite not being the most recent exploit kits we deployed (see Table 6.9). For example, in the period 2008-

2012 Shaman performs up to two times better than Eleonore, despite being two years older. In other words, some exploit kits appear to be designed and armed to affect a wider variety of systems in time than the competition.

3. *Time-specific exploit kits.* As opposed to *long-term exploit kits*, some kits seem to be extremely effective in short periods of time only to "die" shortly after. Eleonore and Bleeding Life belong to this category. The former achieves the highest amount of infection per time window in 2006-2008, and drops then to the minimum within the next two years. The latter is the only exploit kit capable of infecting "recent" machines, i.e. those with configurations since 2009 on. Bleeding Life was in particular clearly designed to attack machines around the period of the release of the kit (2010).

Overall, we find that exploit kits are capable of delivering successful attacks over a prolonged period of time. This supports our Proposition that attack tools traded in the black markets are effective and well-engineered pieces of software that represent a non-transient risk factor for the final user.

Conclusion 3 *From our analysis, we accept Proposition 2. We conclude that the goods traded in the underground are well-engineered and differentiated attack tools that are capable of maintaining the infections over a significant period of time.*

6.2.3 The Markets are Sustainable

Running Hypothesis	Hypotheses Testing
Hyp. 2. The underground markets are sound from an economic perspective.	**Hyp. 2**. Develop a two-stage model of the underground markets to show that the underlying economic mechanism is sound.

For this analysis we utilize qualitative case-study data obtained by infiltrating HackMarket.ru to provide evidence regarding the nature of cognition and bounded rationality in information rich communities engaged in transaction relationships. Our specific goal is to illustrate the emergent market design in communities where contracts are incomplete by construction and the only mechanism of enforcement is based on the shadow-of-the-future.

In particular, we identify three central points that are relevant for our analysis[11]:

1. The markets are strongly regulated, have a coherent reputation mechanism and have *trials* in place to evaluate 'ripping cases' reported by market participants. The trials effectively represent a punishing mechanism whereby the ripper is collectively punished by being effectively exposed as such and listed in the 'do not trade with these users' list.

2. Buyers regularly leave positive and negative feedback on a seller's product by posting publicly on the forum their usage impressions. In this way, buyers that 'arrive second' have additional information on the quality of traded good, and sellers that receive negative feedback will effectively be out of market.

3. To encourage the first buyer in engaging in the trade (as he does not have the cognitional advantages of the second buyer), the seller often provides trial periods, demos and videos of the tool in action. This effort

[11]For conciseness, we do not report here the full record of evidence supporting these claims. Part of it is outlined in Section 6.2.1. A future article version of this Section will contain the full set of evidence.

on the side of the seller decreases the level of uncertainty for the first buyer, effectively addressing part of the asymmetry between her (the seller) and him (the buyer).

We build a two-phase cognition model whereby a seller and a buyer stipulate a contract A for the delivery of a technology. In particular, we consider two independent and a-priori indistinguishable buyers, B_1 and B_2, that may be interested in buying the tool. Because there is no guarantee that the advertised product is not a lemon, the buyer that goes second has an inherent advantage over the first because he can leverage from the first buyer's experience to decide whether A is good for him. Therefore, no buyer would be willing to go first. We show how, to overcome this problem, the seller provides a cognitional advantage to the first buyer (e.g. by giving a trial of the product). Because the seller's goal is to extract maximum value from both buyers, by 'discounting' the cost for B_1 (by decreasing the cognitional effort needed to decide that A is good for him) she (the seller) creates the conditions whereby the profit π_{B1} for the first buyer equals that of the second buyer, π_{B2}. This solves the 'trade entry' problem whereby the first buyer would always want to go second. We show that the condition $\pi_{B1} = \pi_{B2}$ leads to an equilibrium whereby the model solves. We will show that the moderating activity described in Section 6.2.1 is central to establish the equilibrium. By showing that the model is analytically tractable, we conclude that the market mechanism is sound from an economic perspective.

A two period cognition model

We follow [119] and consider a contract to provide a technology denoted A, which may or may not be suitable for a particular buyer. The specificity of the market means that production of A is very costly and that its use is extremely specialised. We will also see that the messaging board type approach to sales means that a single price is generally posted for this product.

The very nature of the market, anonymous posts by cyber criminals on a closed forum, means that enforcement of all contracts is incomplete and the only punishment is via a credible removal of future transactions through a multilateral punishment action or via the dissemination of information about contractual arrangements that have not been fulfilled.

In our case we have a two period set-up, with ex-ante identical buyers labelled B_1 and B_2 contracting from a single seller S. Buyers are ex-ante identical and receive payoff v if the technology A is appropriate for their requirements. Following the notation of [119] the probability that A is the correct technology is denoted $1 - \rho$. This outcome is independent for all buyers, therefore each buyer has an independent probability ρ of receiving the incorrect technology. Whilst buyers are ex-ante identical, cognition on behalf of the seller for a specific buyer is not deemed to be transferable. In the event that the technology A is not appropriate then the buyer suffers a penalty $\Delta \leq v$ and as such receives only $v - \Delta$, rather than v. In the main we will use the working assumption that if the technology is not appropriate then $\Delta = v$ and in effect the buyer will receive no surplus from its deployment.

We view each buyer-seller interaction as being a separate experiment. However, the degree of common knowledge gained in phase one is assumed to permeate into phase two. For instance, buyers post feedback about the efficacy of technology, for good or for bad and this feedback allows future buyers to narrow down their choices. We can think of A as being the standard, or pro-forma, technology and A', the technology suitable for the buyer, is some bespoke arrangement. Therefore each buyer may or may not require the standard arrangement of A and suffer a loss of utility if it turns out that A is not appropriate. As such neither seller not buyer know precisely what the buyers requirements are.

Buyers and sellers can engage in costly cognition to determine whether A is the correct technology for them. For the first buyer, B_1, the cost of

discovering with probability b_1, whether A is the appropriate technology for them is denoted $T_{B_1} = T_B(b_1)$. We follow the standard assumptions in the cognition literature and [119] in particular and assume that the cognition costs originate at the origin, are strictly increasing and have a singularity at unity. Hence, $T_B(0) = 0$, $T'_B(0) = 0$, $0 < T_B(z) < \infty, \forall 0 < z < 1$ and $T_B(1) = \infty$, where $z = \{b_1, b_2\}$. For simplicity we assume that the second buyer has an advantage over the first buyer in discovering if A is appropriate by a fixed cost factor $0 \leq \delta \leq 1$, therefore the cost of cognition for B_2 is $T_{B_2} = \delta T_B(b_2)$. Sellers can also engage in costly cognition by studying the buyer and their own technology and can discover if A is suitable for a given buyer independently with probability s that may or may not be revealed to the buyer. Similarly to the buyer we assume that the cost of cognition is denoted $T_S(s)$ and that $T_S(0) = 0$, $T'_S(0) = 0$, $0 < T_S(z) < \infty, \forall 0 < z < 1$ and $T_S(1) = \infty$, however $T_B(z)$ need not equal $T_S(z)$ for a given z. We will assume that T_S is the same for the seller in both periods; whilst this appears to be a limiting assumption our analysis will in general focus on cognition by the seller with buyer B_1. We assume that buyers will be indifferent from being first or second if their respective pay-offs are the same.

For analytical tractability we will show that it is simpler to relax the $T'_B(0) = 0$ assumption and place a constraint on the parameters of the function $T_B(z)$ to ensure that the optima of the function lies within $0 < z^* < 1$ range.

We assume that $T_S(s)$ is independent of b_i, therefore seller cognition does require costly buyer cognition as an input. This non-collaborative condition is justifiable for many types of technology, whereby the cost of cognition for buyer or seller is in the revelation of the 'modality' of the technology, e.g. the revealing of source code or methods of forcing memory overflows by exploitation of certain key vulnerabilities in common software. From the viewpoint of the seller, in-the-main, this is a one sided cost as the buyer now

has the information needed to replicate the sellers technology at near zero cost.

The cognition mechanism

We now assume two update functions: First $\hat{\rho}(b_{i\in\{1,2\}})$, which is the private ex-ante probability that the buyer knows if A is suitable. By construction when $b_i = 1$, $\hat{\rho}(b_{i\in\{1,2\}}) = 0$, that is Buyer i knows with certainty the suitability of A. When $b_i = 0$, $\hat{\rho}(b_{i\in\{1,2\}}) = \rho$, that is Buyer i is subject to the unconditional probability ρ of A being incorrect. Second, when the seller provides costly cognition to assist the buyer in the first sub-step and the buyer then chooses their own costly cognition in the second step, we denote this $\bar{\rho}(s, b_{i\in\{1,2\}})$. Following [119] Bayesian updating we find the following functional forms:

$$\begin{aligned} \hat{\rho}(b_{i\in\{1,2\}}) &= \frac{\rho(1-b_i)}{1-\rho b_i} \\ \bar{\rho}(s, b_{i\in\{1,2\}}) &= \frac{\rho(1-s)(1-b_i)}{1-\rho s-\rho b_i(1-s)} \equiv \frac{\hat{\rho}(s)(1-b_i)}{1-\hat{\rho}(s)}, \\ & \quad where, \quad \hat{\rho}(s) = \frac{\rho(1-s)}{1-\rho s} \end{aligned} \tag{6.7}$$

When A is incorrect, the buyer suffers a variance of utility $v > \Delta > 0$ from the endowment v, therefore the good now provides is $v - \Delta$ rather than Δ. Let us consider a price p provided by the seller, in the event that the seller provides some costly cognition to the buyer $s > 0$, then the buyers expected payoff for any given b_i is:

$$\pi_{B_i} = v - p - \bar{\rho}(s, b_i)\Delta - T_{B_i}, \quad for \quad i \in \{1, 2\}, \tag{6.8}$$

where

$$T_{B_i} = \begin{cases} T_{B_1}(b_1) = T_B(b_1), & for \quad i = 1 \\ T_{B_2}(b_2) = \delta T_B(b_2), & for \quad i = 2, \quad and, \quad 0 \leq \delta \leq 1 \end{cases} \tag{6.9}$$

if the seller engages in no cognitive effort then this is denoted:

$$\pi_{B_i} = v - p - \hat{\rho}(b_i)\Delta - T_{B_i}, \quad for \quad i \in \{1, 2\} \tag{6.10}$$

When $0 < \delta < 1$, the buyer in period 2, B_2, has a cognitive advantage that for any given probability b_2, the cost of acquiring this 'extra' reduction in likelihood that A is not the correct technology is cheaper than for the first buyer. We justify this by presuming that the level of common-knowledge about the technology, on the buyer side, may increase with use. This also forms the basis of our initial hold-up problem, as it is obvious that the second buyer will have a higher pay-off, in expectations for any given p. We shall now demonstrate this effect.

The Price Setting Seller Assumption

We consider a seller S who is a price setter with bargaining power such that he extracts all of the joint surplus of the Buyers. This assumption simplifies the price setting problem such that the sellers optimal price is that which maximises their surplus π_S and has a boundary such that the buyers must at least breaking even $\pi_{B_{i\in\{1,2\}}} \geq 0$.[12] The seller anticipates that for a given buyer $B_{i\in\{1,2\}}$, the highest surplus maybe extracted by the buyer engaging in cognition and reducing the likelihood of the buyer obtaining $v - \Delta$ rather than Δ. This brings us to our first case when the seller suffers no penalty for selling A when it is not suitable for a given buyer in one or both periods.

The Tight Margins assumption

Let $0 \leq \gamma \leq 1$ be the discount rate between period one and period two. Let the seller incur deterministic cost c_1 and c_2 in each period for producing the technology A, the cost $c_1 + c_2$ is assumed to be committed. We assume that profit margins are tight therefore $(1 - \rho)p \leq c_{i\in\{1,2\}}$ and $(1 - \rho)p + \hat{\rho}(b^*)p >$

[12] In the [119] set-up this is setting $\sigma = 1$ and hence $\beta = 0$.

$c_{i\in\{1,2\}}$, where b^* is a degree of cognition on the equilibrium path. This assumption is simply to push our model to a unique solution with both buyers.

The Price Commitment assumption

Our fundamental assumption, in terms of cultural constraints, is that once the price is announced by the seller she has to commit to this price across both time periods. Our evidence from the market is that prices are extremely sticky, in fact we have found no evidence of a single change in price without a substantial change in the good on sale. The major reason for this is that we can think of the product being simultaneously advertised to both B_1 and B_2 and the time interval between purchases is effectively the time taken to licence the technology and deploy the malware. This maybe measured in a few days. Once B_1 has deployed the malware 'in-the-wild' then the next buyer B_2 will now be able to view the modality of the technology by simple observation of its performance for specific tasks on the internet and by the reaction of security firms in attempting to mitigate its impact. The new buyer knows their particular requirements and hence they can update their position on the effectiveness of the malware or the supply of compromised machines more cheaply than B_1.

The need for the seller's cognition effort

We here consider two cases: in the first the seller does not sustain any cognition cost. We will show that in this case the first buyer will always want to go second, and therefore no trade would be initiated. In the second case, the seller engages in costly cognition to alleviate the costs for buyer one, such that the revenue buyer one can extract from the trade equals that of the second buyer.

Our two cases are therefore defined in the following way:

1. Seller engages in no cognitive effort with either buyer.

$$\pi_S = (1-\rho)p + \hat{\rho}(b_1)p - c_1 + \gamma((1-\rho)p + \hat{\rho}(b_2)p - c_2) \tag{6.11}$$

2. Seller engages in cognitive effort with first buyer.

$$\pi_S = (1-\rho)p + \bar{\rho}(s, b_1)p - c_1 - T_S(s) + \gamma((1-\rho)p + \hat{\rho}(b_2)p - c_2) \tag{6.12}$$

Proposition 1a: The buyer equality problem When $0 < \delta < 1$ the the seller cannot set a unique price p in both periods such that the buyer surpluses maybe equalized, i.e. $\pi_{B_1} = \pi_{B_2}$, for the unique optimal choices of cognition b_1 and b_2, denoted $b_1^\dagger$ and $b_2^\dagger$ respectively.

Proposition 1b: The price hold-up problem It follows that even when the seller has complete bargaining power, when $0 < \delta < 1$, when the seller sets when $\pi_{B_1} = 0$, for an optimal choice of b_1, denoted $b_1^\dagger$, the surplus of the second buyer, π_{B_2} will be greater than zero, for the unique optimal choice of b_2, denoted $b_2^\dagger$.

Following [119] we constrain ourselves to the cases where $v - \Delta > 0$. The seller is a price setter able to extract all of the joint surplus, therefore the maximum available price is that which sets $\min(\pi_{B_i}(b_i^*) = 0)$ for $i \in \{1, 2\}$. In our set-up the price does not affect the cognition choice, only the trade-off between $T_{B_i}(b_i)$ and $\hat{\rho}(b_i)$, this is evident as the seller's statistical model of the buyer solves separately $\pi'_{B_i} = 0$. For the case when the seller engages in no cognition $s = 0$ for either buyer, this yields an optimal cognition of $b_i^\dagger$ that satisfies:

$$T'_{B_i}(b_i) = -\frac{(\rho-1)\rho}{(b_i\rho - 1)^2}\Delta, \quad for, \quad i \in \{1, 2\} \tag{6.13}$$

by definition $T_{B_2}(b_2) = \delta T_{B_1}(b_2)$, where $0 < \delta < 1$.

Let $\mathcal{T}_i(b_i) = \hat{\rho}(b_i)\Delta + T_{B_i}(b_i)$, be the cognition trade-off. The sub-problem of each buyer is equivalent to $b_i^\dagger \triangleq \arg\min_{b_i} \mathcal{T}_i(b_i)$. Consider any $b_i^\dagger, i \in \{1, 2\}$

that solves the cognition for the first buyer, we know that by construction $\delta T_B(b_1^\dagger) < T_B(b_1^\dagger)$, hence B_2 can always find a $b_2^\dagger \geq b_1^\dagger$ that provides an identical or greater reduction in uncertainty for lower cost, as such $\mathcal{T}_2(b^\dagger) < \mathcal{T}_1(b^\dagger), \forall 0 < b^\dagger < 1$. Is $b_1^\dagger = b_2^\dagger = 0$ cognition a viable optimal point in equalizing the expected loss of utility to $\rho\Delta$ for both B_1 and B_2? No, as again by construction of the cognition cost function $T'_B(0) = 0$, therefore by if $v > \Delta > 0$ then it is always better to conduct at least a finite amount of non-zero cognition, hence $b_i^\dagger > 0, i \in \{1, 2\}$ and the seller can still find a positive price $p > 0$ such that $\min(\pi_{B_i}) \geq 0$. Therefore, by construction $\pi_{B_1} \neq \pi_{B_2}$ and $\mathcal{T}_1(b^\dagger) < \mathcal{T}_2(b^\dagger)$ for $b^\dagger = b_1^\dagger = b_2^\dagger$, the lower bound of $B_2's$ optimal cognition. Furthermore, for any given price $p^\dagger$ that the seller optimally sets for either B_1 or B_2, the pay-off π_{B_2} will be greater than π_{B_1} as the term $\mathcal{T}_2(b_2^\dagger)$ will always be finite and smaller than $\mathcal{T}_1(b_1^\dagger)$

Hence, from a buyer point of view it is always sub-optimal to be the first buyer even if the seller sets a price on or above $\pi_{B1} = 0$ as a better payoff can always be achieved by going second when $0 < \delta < 1$, similarly if the seller sets a price to extract the surplus of B_2, the surplus of B_1 will be negative . Whilst Proposition 1a and 1b trivially fall out of the model construction, it is worth noting their implication. When prices are very sticky, it is sub-optimal to enter into a contract for a good with a potentially random pay-off as a first buyer. The advent of social learning and hence the ability to conduct cheaper cognition as a buyer in the second phase results in a natural hold-up that would not occur if the buyers had equal inter-temporal cognition costs. Whilst trial period sales are usually a mechanism of reducing the impact of deviation in consumption (measure by Δ here) by the buyer at each step we demonstrate a new mechanism, which is the dissemination of new information and the ability to cheaply process this after the fact.

How much surplus is gained by going second?

We have established that Seller Case 1 results in a hold-up as B_1 will always prefer to be B_2 as B_2 has greater bargaining power than B_1 directly because of the cognition channel. By setting a specific functional form to T_B we can exactly quantify the implicit cognition discount the second buyer receives. This also provides insight on the trade-offs the seller must make to acquire the best price given her explicit bargaining power.

Consider now the case whereby we choose $\bar{p}$ such that $\pi_{B_2} = 0$. The seller's statistical model of the buyer indicates that his cognition trade-off is independent of v and p therefore we can set:

$$\bar{p} = v - \mathcal{T}_2(b_2^\dagger) \equiv v - \hat{\rho}(b_2^\dagger)\Delta - T_B(b_2^\dagger), \quad where \quad b_2^\dagger \triangleq \arg\min_{b_2} \mathcal{T}_2(b_2) \tag{6.14}$$

We know that $\bar{p}$ is the highest price the seller can charge before B_2 drops out and is therefore the upper boundary on the sellers price range. From Proposition 2b we know that at $\bar{p}$, B_1 will now be below break-even as $\mathcal{T}_1(b_1^\dagger) > \mathcal{T}_2(b_2^\dagger)$.

In contrast the seller can set $\underline{p}$ as the lower boundary price under consideration by the seller. By construction of the model this is set to be the price when $\pi_{B_1} = 0$. Because the difference in cognition costs is independent of the price, the seller will need to compensate the first buyer by $\mathcal{T}_1(b_1^\dagger) - \mathcal{T}_2(b_2^\dagger)$ whatever the price of A. The optimal cost of doing this $T_S(s^\ddagger)$ will be at least the same at $\underline{p}$ as any other price.

However, once the seller engages in cognition, the price cannot exceed $\bar{p}$ as this eliminates buyer 2. Is it possible for the seller to engage in cognition and drive the price above $\bar{p}$? Yes it is possible; if the seller has a relatively flat cognition function, when $0 < s < 1$ and the likelihood of A not being suitable is relatively small then the seller can drive the first buyers $\bar{\rho}(s, b_1) \to 0$ inexpensively (from the viewpoint of decreasing the revenue from $\bar{\rho}(s, b_1)p$ and increasing costs associated with $T_S(s)$). Subsequently the buyers choice

of cognition b_1 will also tend to zero, $\lim_{b_1 \to 0} T_B(b_1) = 0$ and hence the break even price the seller can charge and still leave B_1 at break-even tends to v. We eliminate this case by appealing to our 'Tight-Margins' assumption, that is the seller must sell to both buyers, rather than using cheap cognition to eliminate B_2 and extract maximum revenue from B_1.

The seller's optimization problem now appears more complex. In the first case the seller had no direct control over the first and second buyers cognition, however this led to no single price equalising the surplus of both buyers and hence a hold–up. Now that the seller chooses to engage in non-zero cognition to equalise the first buyers costs, she now directly impacts her own surplus by directly influencing the probability that the first buyer will not be in error in choosing A, by increasing s, she pushes the term $\bar{\rho}(s, b_1^\dagger)p$ towards zero and hence she pushes her own profit towards her cost constraint c_1 in period 1.

Let us assume that the seller engages in non-zero cognition and transfers this to the buyer, then B_1 will adjust their choice of cognition b_1 to a new optimum $b_1^\ddagger$ by solving:

$$\frac{\partial T_{B_1}(b_1)}{\partial b_1} = -\frac{\partial \bar{\rho}(b_1)}{\partial b_1}\Delta \equiv -\frac{(\rho - 1)\rho(s-1)}{(b\rho(s-1) - \rho s + 1)^2}\Delta.$$

Recall that the seller does not influence $\hat{\rho}(b_2^\dagger)p$ in this instance. Furthermore, recall that as the seller pushes the term $\hat{\rho}(b_2^\dagger)\Delta$ lower she can now extract more surplus from the buyer at cost $T_S(s)$ to herself. However, we can see by simple inspection that the constraint:

$$\mathcal{T}_1(b_1^\dagger) - \mathcal{T}_2(b_2^\dagger) = \Delta(\hat{\rho}(b_1^\dagger) - \bar{\rho}(s^\ddagger, b_1^\ddagger)) + T_B(b_1^\ddagger) - \delta T_B(b_2^\dagger) \tag{6.15}$$

is required to solve for the required amount of cognition, regardless of price. Whilst depending on the functional form of $T_B(b_1)$, the optimal $s^\ddagger$ might simplify, the strict ordering to ensure the seller creates maximum extractable surplus for the buyer requires $s^\ddagger$ to be solved backwards from $b^\ddagger$. Recall that

the seller has buying power, so the best s may not be the joint minimization of $\tilde{\mathcal{T}}_1(s, b_1) = \bar{\rho}(s, b_1)\Delta + T_B(b_1)$. To ensure that she chooses the smallest viable s she rearranges the constraint in (6.15) to give $b_1^{\ddagger}$ as a function of s. The joint solution with (6.15) is the unique level of cognition needed to provide B_1 with the same cognition cost – error trade-off as B_2, who benefits from the global improvement in technology cognition through the factor δ.

Quantifying the trade-off

We will now investigate a case where the seller is will engage in cognition in order to overcome the hold-up. We will then quantify the boundary at which cognition is now too expensive for the seller to overcome the hold-up and still at least break-even.

It is useful at this juncture to place a functional form on $T_B(\cdot)$ and $T_S(\cdot)$ so that we can illustrate the trade-offs and simplify the discussion for our approach to the re-contracting phase. An obvious choice for $T_B(\cdot)$ is $Hz^2/(1-z^2)$. To ensure analytical tractability, we specify:

$$T_j(z) = \begin{cases} \frac{H_j z}{1-z} & 0 < z \leq 1 \\ \frac{H_j z^2}{1-z^2} & z = 0 \end{cases}, \quad j \in \{B, S\}, \tag{6.16}$$

this enforces an interior solution on the problem, but permits a tractable solution, where $H_j j \in \{B, S\}$ is a scale parameter that we will refer to as the "scale of costs". In general we will focus on the $b^{\dagger} > 0$, therefore the solutions to $T'_B(b_1) = -\hat{\rho}'(b_1)\Delta$ are constrained to cases when

$$H_B < -\Delta(\rho^2 - \rho) \tag{6.17}$$

similarly, when the seller engages cognition with the first buyer and sets $0 > s > 1$, we restrict ourselves to analyzing the cases where

$$H_B < -\Delta \frac{\rho^2 - \rho - \rho^2 s + \rho s}{\rho^2 s^2 - 2\rho s + 1}. \tag{6.18}$$

The constraints are needed to ensure that the $T_B(z)$ function forces a solution within the $0 < b_i < 1$. The more general interpretation of this is that if cognition is relatively expensive for $0 < b_i < 0$ then this ceases to be the major issue for the contracting phase. Furthermore, for (6.18), we can substitute for s the functional form for $s^{\ddagger}$, to compute the upper bound on H_B.

The optimal cognition bundle with and without seller cognition effort

It is helpful in the following discussion to specify the following pair of auxiliary functions that form components of the optimal solutions for $b_{i\in\{1,2\}}$.

$$\mathcal{H}_i = \sqrt{-\delta_i \Delta H_B(\rho-1)^3\rho}, \tag{6.19}$$

$$where, \quad \begin{cases} \delta_i = 1, & i = 1 \\ \delta_i = \delta, & i = 2 \end{cases}$$

$$\mathcal{D}_j = \rho(s_j - 1)(\Delta(1-\rho) + \delta_i H_B \rho(s_j - 1)), \tag{6.20}$$

$$where, \quad \begin{cases} s_j = 0, & j = 0 \\ s_j = s, & j = s \end{cases}$$

we can interpret $\mathcal{H}_i/\mathcal{D}_j$ as the relative probabilistic advantage of choosing a particular level of b relative to the costs (again in probability equivalents) created by the uncertainty in the quality of A. When the seller engages in no cognition, Seller Case 1 and the functional form of $T_B(z)$ is as specified in (6.16) then the optimal choice of b_i for each of the buyers is given by:

$$b_i^{\dagger} = \frac{1}{\mathcal{D}_0}(\Delta(\rho-1)\rho + \delta_i H_B \rho + \mathcal{H}_1), \quad for, \quad i \in \{1,2\} \tag{6.21}$$

where Δ and H_B are subjective to the conditions specified in (6.17). For Seller Case 2 when s is non-zero the optimal choice of b_1 is given by:

$$b_1^{\ddagger} = \frac{1}{\mathcal{D}_s}(\rho(s-1)(\Delta(1-\rho) + H_B(\rho s - 1)) + \mathcal{H}_1) \tag{6.22}$$

it is relatively trivial to show that $b^{\dagger}$ is always greater than $b^{\ddagger}$ when $0 < \delta < 1$ and hence $\mathcal{T}_1(b^{\dagger}) > \mathcal{T}_1(b^{\ddagger})$. The optimal solution for B_2 will be the same as

B_1 in both Seller Case 1 and Seller Case 2, which is the rearrangement of the solution for $b_1^\dagger$ with δH instead of H. To compute the optimal cognition s that the seller should choose to ensure that B_1 and B_2 receive the same pay-offs we replace the optimal solutions for $b_1^\dagger$ and $b_2^\dagger$ and $b_1^\ddagger$ into (6.15) and solve for $s^\ddagger$, this is our next proposition.

Proposition 2: The seller's optimal level of cognition

When $T_B(z)$ is defined as in (6.16), the Sellers required cognition $s^\ddagger$ needed to equalise the expected surplus of B_1 and B_2 is determined by

$$\begin{aligned} s^\ddagger &= \pm\frac{2\sqrt{\Delta(\rho-1)^2\rho^2\left(\Delta(\rho-1)^2-\frac{1}{2}\mathcal{H}_1-H_B(\rho+\delta-1)(\rho-1)\right)}}{H_B(\rho-1)\rho^2} \\ &\quad +\frac{2\Delta(\rho-1)}{H_B\rho}-\frac{\mathcal{H}_1}{2H_B(\rho-1)\rho}-\frac{\delta-1}{\rho} \end{aligned} \tag{6.23}$$

Notice that both roots of (6.23) can provide a solution to $s^\dagger$ in the $0 < s^\ddagger < 1$ domain, the seller would obviously choose the lower s.

We have not yet optimized the sellers pay-off explicitly. This is because the highest available price $p^\dagger$ can be already charged to buyer B_2 and at this stage the cognition is solely dependent on the first buyers relative cognitive disadvantage to the second buyer. Furthermore, the optimal cognition choice for the second buyer is, by construction, not affected by the price. Hence, for the seller, if cognition is the only mechanism of discount then for any higher price the seller violates the 'tight margins' constraint as B_2 will drop out. Furthermore, the hold-up discount needed by B_1 is not connected to the price at all, cognition apart, he is ex-ante identical to B_2, so the only driver for the degree of s needed by B_1 in order to motivate him to transact in period one is the relative cognition costs. As such $s^\ddagger$ is the required level of cognition needed to ensure that buyers do not strictly prefer to go second. However, the level of $s^\ddagger$ may not be consistent with the break-even requirement of the seller. We shall now explore the implication and observe how a social planner

can set a re-contracting penalty that increases the domain of solutions over which pairs of two period buyers and sellers can enter into arrangements that overcome cognition based hold-up problems.

The seller's cost constraint

Following convention we assume that the seller only enters into a contract when $\pi_S \geq 0$. It is trivial to show that by inspection there is a critical upper level on H_S, after which the seller finds the process of cognition too expensive to equalize the pay-offs of both B_1 and B_2 hence overcoming the hold-up. This upper bound denoted $\bar{H}_S$ is given by

$$\begin{aligned}\bar{H}_S &= \frac{s^{\ddagger}-1}{s^{\ddagger}}c_1 - \frac{(s^{\ddagger}-1)}{\Delta(\rho-1)^4 s^{\ddagger}}(\Delta\rho(\rho-1)^2 + \\ &\quad +\mathcal{H}_s(-(\rho-1)(\delta H_B - \rho v + v) - \mathcal{H}_1),\end{aligned}$$

where $\mathcal{H}_s = \sqrt{\Delta H_B(\rho-1)^3\rho(s^{\ddagger}-1)})$, therefore for any given configuration of the structural parameters v, ρ, δ, c_1, H_B and Δ, seller cognition costs above $\bar{H}_S$ result in a systematic cognition hold-up that the seller cannot overcome whilst still at least breaking even. We can interpret seller cognition costs beyond this boundary as a market failure as systematically buyers will prefer to delay going first and sellers cannot provide enough cognition to discount B_1.

The role of the board moderator

A simple case exists where the cost of cognition for the seller is high enough that they cannot provide enough of a cognition discount to B_1 to prevent delay without violating the sellers expectation of at least breaking even. For a criminal market any promise or requirement by a social planner is, of course, incomplete. By their very nature a buyer within a criminal market cannot enforce a re-contracting phase so that the seller provides and adjustment

to compensate for the variance from v to $v - \Delta$. Let us assume that re-contracting costs for the seller to provide an adjustment are given by $\lambda(a)$, where a is a proportion of Δ recouped by the Buyer if A is not suitable. Alternatively, we can think of a as a promised transfer of surplus from the seller to the buyer ex-post.

Let us consider a cognition cost coefficient $H_S^* > \bar{H}_S$. Here the cost of cognition for the seller needed to discount the initial buyers is too high. If B_1 believes that the seller will provide appropriate adjustment or compensation then the seller can reduce their costly cognition by promising an ex-post correction upon discovery of whether A is suitable for the buyer. Given that both the buyer and seller assumes that ex-post all contracts are incomplete there is a commitment issue. If the seller provides a guarantee of offsetting an a- priori agreed fraction after the buyer has taken possession of A then the buyer will not trust that this off-set will be delivered as there is no mechanism to enforce the contract. Similarly, if the seller provides collateral (for instance a trial discount) then they have no guarantee that the buyer will pay for the full value of the good, if A turns out to be suitable.

However, if the seller and buyer can provide guarantees the seller will be able to find a solution to the $H_S^* > \bar{H}_S$. Furthermore, the seller will, in all likelihood, be able to rebalance the expected surplus of the buyer and seller at a cheaper rate, even if H_S^* is not greater than $\bar{H}_S$. However, at this stage it is instructive to address the stage at which the seller must be able to provide this guarantee in order to ensure an initial sale to B_1. The sellers objective is to finance (via cognition or compensation)

$$\mathcal{T}_1(b_1^\dagger) - \mathcal{T}_2(b_2^\dagger) = \frac{1}{(\rho-1)^2}(2(\mathcal{H}_1 - \mathcal{H}_2) + H_B(\delta + \rho - \delta\rho - 1)) \quad (6.24)$$

as cheaply as possible. This is achievable by compensating $a\Delta$ with probability $\bar{\rho}(s, b_1)$ ex-post or by cognition $\Delta(\hat{\rho}(b_1^\dagger) - \bar{\rho}(s, b_1)) + T_B(b_1^\dagger) - T_B(b_1^*)$.

Therefore the seller needs to discount the buyer by

$$\hat{\rho}(b^{\dagger})\Delta + T_B(b^{\dagger}) - \bar{\rho}(b_1^*, s^*)(\Delta - a^*\Delta) - T_B(b_1^*) = \mathcal{T}_1(b_1^{\dagger}) - \mathcal{T}_2(b_2^{\dagger}) \quad (6.25)$$

In this case we have one constraint and two decision variables a and s chosen by the seller and the anticipated b_1 from the sellers statistical model of the first buyer.

Proposition 3: Existence of the buyer's optimal compensation and cognition bundle

When the cognition function $T_B(z)$ is as defined in (6.16) if the seller can provide a full commitment to B_1, then from the sellers statistical model of the buyer the optimal choice of cognition for the seller is given by:

$$\begin{aligned} b_1^*(a^*, s) &= \frac{1}{\mathcal{D}_a}\rho(s-1)((a^*-1)\Delta(\rho-1) + \\ &\quad + H_B(\rho s - 1)) + \frac{1}{\sqrt{\delta}\mathcal{D}_a}\sqrt{(a^*-1)(s-1)}\mathcal{H}_1 \end{aligned} \quad (6.26)$$

where $\mathcal{D}_a = \rho(s-1)((a-1)\Delta(\rho-1) + H_B\rho)$ the seller's optimal choice of compensation, denoted a^*, for a given level of s by:

$$\begin{aligned} a^*(s) &= \frac{1}{4\Delta(\rho-1)^2\rho(s-1)} \times \\ &\quad 4(\delta + \rho s - 1)\mathcal{H}_1 + H_B(\rho-1)(\delta + \rho s - 1)^2 + \\ &\quad -4\Delta\rho(\rho-1)^2(\delta + s - 1) \end{aligned} \quad (6.27)$$

Further substitution into the expression for the buyer profit function, denoted $\pi_S(a^*(s), b^*(a^*, s))$, permits a one dimensional optimization with respect to the optimal cognition s^*. The expression for s^* can be derived analytically. We provide its formulation in the internet Appendix.

Conclusion 4 *By showing that a cognition bundle exists that solves the model, we showed that the market is sustainable. We therefore accept Hypothesis 2.*

Chapter 7

Risk-based Policies for Vulnerability Management

The analyses proposed in the previous Chapter all strongly support the idea that vulnerability exploitation and attacker preference can represent a significant factor to think of more efficient risk-based vulnerability management practices as opposed to current criticality-based practices. We therefore proceed in testing our last running hypothesis:

Running Hypothesis	Hypotheses Testing
Hyp. 3. It is possible to construct risk-based policies that, levereging the economic nature of the attacker, can greatly improve over criticality-based policies.	**Corollary to Hyp. 3** *Risk-based policies accounting for cybercrime markets are the most effective in reducing risk for the final user.* Develop a case control study to evaluate the overall risk-reduction of risk based and criticality based vulnerability management policies. A validating example outlines the benefits of risk-based policies over criticality based ones in terms of patching workloads and effectiveness in foiling real attacks in the wild.

In the following we present our case control methodology to assess the effectiveness of a vulnerability management policy, and show that risk-based policies outperform by far current criticality-based ones in terms of patching efficiency. In particular, we:

1. Introduce the 'case-control study' as a fully-replicable methodology to

soundly analyze vulnerability and exploit data.

2. Check the suitability of the current use of the CVSS score as a risk metric by comparing it against exploits recorded in the wild and by performing a break-down analysis of its characteristics and values.

3. We use the case-control study methodology to show and measure how the current criticality-based CVSS practice can be improved by considering additional risk factors defining a risk-based policy. To do this, we provide a quantitative measure of the reduction in risk of exploitation yield by the resulting policies.

7.1 Risk-based vs Criticality-based Policies

Following the analysis provided in Chapter 6 we conclude that cost of exploit and existence of the exploit in the underground markets are significant factors for likelihood of exploitation. In order to measure these factors, we account for:

1. The existence of a proof-of- concept exploit that lowers the attacker's cost to deploy a working exploit (Section 6.1).

2. The existence of technology traded in the black markets that bundles the exploit (Section 6.2.1-6.2).

This, in combination with a criticality measure, results in the definition of a risk-based policy that accounts for both the likelihood of exploitation *and* the criticality of the vulnerability. Table 7.1 reports the criticality-based and risk-based policies we consider here. Whereas a criticality-based policy relies solely on the CVSS score, a risk- based policy leverages the presence of a risk factor as an indicator of likelihood of exploitation. This gives a risk estimation of the vulnerability that corresponds more closely to the classic

Table 7.1: Criticality-based and risk-based policies.

Policy type	Policy name	Likelihood measure	Criticality measure
Criticality-based	CVSS	-	CVSS score
Risk-based	PoC	Existence of a proof-of-concept exploit	CVSS score
Risk-based	BMar	Presence of the exploit in the black markets	CVSS Score

$Risk = likelihood \times impact$ definition of risk. To measure the risk factors for BMar and PoC we use the information reported in the EKITS and EDB datasets respectively.

7.2 Randomized Case-Control Study

Randomized Block Design Experiments (or Controlled Experiments) are common frameworks used to measure the effectiveness of a treatment over a sample of subjects. These designs aim at measuring a certain variable of interest by isolating factors that may influence the outcome of the experiment, and leave to randomization other factors of not primary importance. However, in some cases practical and ethical concerns may make an experiment impossible to perform.

When an experiment is not applicable, an alternative solution is to perform a retrospective analysis in which the *cases* (people with a known illness) are compared with a random population of *controls* clustered in 'blocks' (randomly selected patients with the same characteristics). These retrospective analyses are called *randomized case-control studies* and are in many respects analogous to their experimental counterpart. A famous application of this methodology is the 1950 study by [45], where the authors showed the correlation between smoking habits and the presence or absence of cancer of the lungs by performing a case-control study with data on hospitalization. We

revisit this methodology to assess whether a vulnerability risk factor (like the CVSS score) can be a good predictor for vulnerability exploitation, and whether it can be improved by additional information.

We start by giving the reader some terminology:

- *Cases.* The cases of a control study are the subjects that present the observed effect. For example, in the medical domain the cases could be the patients whose status has been ascertained to be 'sick'. In a computer security scenario, a 'case' could be a vulnerability that has been exploited in the wild. For us a case is therefore a vulnerability in SYM.

- *Explanatory variable or risk factor.* A risk factor is an effect that can explain the presence (or increase in likelihood) of the illness (or attack). Considered risk factors for cancer may be smoking habits or pollution. As reported in Table 7.1, for vulnerability exploitation we consider the existence of a Proof-of-Concept exploit ($vuln \in \mathsf{EDB}$) and the presence of an exploit in the black markets ($vuln \in \mathsf{EKITS}$).

- *Confounding variables* are other variables that, combined with a risk factor, may be an alternative explanations for the effect, or correlate with its observation. For example, patient age or sex may be confounding factors for some types of cancer. In our case the existence of an exploit in SYM may depend on factors such as type of vulnerability impact, time of disclosure, and affected software.

- *Control group.* A control group is a group of subjects chosen at random from a population with similar characteristics (e.g. age, social status, location) to the cases. In the original design of a case-control study, the control group was composed of healthy people only. However, with that application of the case-control study we can only ascertain whether the

risk factor of interest has a greater incidence for the cases than for the controls. We relax this condition and leave open the (random) chance that cases get included in the control group. This relaxation allows us to perform additional computations on our samples (namely CVSS sensitivity, specificity and risk reduction). This, however, introduces (random) noise in the generated data. To address this issue, we perform the analysis with bootstrapping.

- *Bootstrapping.* A classic statistical significance test allows the researcher to relax certain conditions on the linear relationship between dependent variables (our cases) and independent variables (our hypotheses, or risk factors). However, the precision of these tests often depends on the underlying data distribution, that need be known. In our case the real data generation process (DGP) underlying our observations is however unknown: we do not have a precise model of the engineering of an exploit, its delivery in the wild, and of the probability distribution of detection by Symantec. Bootstrapping is a statistical technique that allows us to overcome this problem by re-sampling cases, with replacement, from our distribution of exploits in the wild. The *Fundamental Theorem of Statistics* guarantees, in fact, that with enough random and independent extractions from a distribution, the 'empirical distribution function' (EDF) that is obtained converges to the real one [43], as does the statistic of interest (e.g. the mean, or a p-value). Therefore, by bootstrapping our sample, we can compute our statistics over an EDF that converges asymptotically with the real distribution of exploits in the wild (that we can not observe). This improves the statistical efficiency of our estimation, and therefore the precision of our conclusions.

Confounding variables Deciding which confounding factors to include in a case-control study is usually left to the intuition and experience of the re-

searcher [45]. Because SYM is the 'critical point' of our study (as it reports our cases), we consulted with Symantec to decide which factors to consider as confounding. While this list can not be considered an exhaustive one, we believe that the identified variables capture the most important aspects of the inclusion of a vulnerability in SYM. In the following we discuss the confounding variables we choose and the enforcement of the respective controlling procedure:

- **Year.** Symantec's commitment in reporting exploited CVEs may change with time. After a detailed conversation with Symantec it emerged that the inclusion of a CVE in an attack signature is an effort on Symantec's side aimed at enhancing the usefulness of their datasets. Specifically, Symantec recently opened a data sharing program called WINE whose aim is to share attack data with security researchers [47]. The data included in the WINE dataset spans from 2009 to the present date. Given the explicit sharing nature of their WINE program, we consider vulnerabilities disclosed after 2009 to be better represented in SYM. We therefore consider only those in our study.

  Enforcement: Unfortunately vulnerability time data in NVD is very noisy due to how the vulnerability disclosure mechanism works [105, 81]. For this reason, an *exact match* for the disclosure date of the sampled vulnerability sv_i and the SYM vulnerability v_i is undesirable. In our case a coarse time data granularity is enough, as we only need to cover the years in which Symantec actively reported attacked CVEs. We therefore enforce this control by first selecting for sampling only vulnerabilities whose disclosure dates span from 2009 on, and then by performing an exact match in the year of disclosure between sv_i and v_i.

- **Impact type.** Our analysis (Section 5.1.1) showed that some CIA types are more common in SYM than elsewhere (e.g. CIA='CCC').

An explanation for this may be that attackers contrasted by Symantec may prefer to attack vulnerabilities that allow them to execute arbitrary code rather than ones that enables them to get only a partial access on the file system. We therefore also control for the CVSS Confidentiality, Integrity and Availability assessments.

Enforcement: The CVSS framework provides a precise assessments of the CIA impact. We therefore perform an exact match between the CIA values of the sampled vulnerability sv_i and that of v_i (in SYM).

In addition, we 'sanitize' the data by *Software.* Symantec is a security market leader and provides a variety of security solutions but its largest market share is in the consumer market. In particular, the data in SYM is referenced to the malware and attack signatures included in commercial products that are often installed on consumer machines. These are typically Microsoft Windows machines running commodity software like Microsoft Office and internet plugins like Adobe Flash or Oracle Java[1] [46]. Because of this selection problem, SYM may represent only a subset of all the software reported in NVD, EDB or EKITS.

Enforcement: Unfortunately no standardized way to report vulnerability software names in NVD exists, and this makes it impossible to directly control this confounding variable. For example, CVE-2009-0559 (in SYM) is reported in NVD as a "Stack-based buffer overflow in Excel", but the main affected software reported is (Microsoft) Office. In contrast, CVE-2010-1248 (in SYM as well) is a "Buffer overflow in Microsoft Office Excel" and is reported as an Excel vulnerability. Thus, performing a perfect string match for the software variable would exclude from the selection relevant vulnerabilities affecting the same software but reporting different software names.

[1]Unix software is also included in SYM. However we do not consider this sample to be representative of Unix exploited vulnerabilities.

The problem with software names extends beyond this. Consider for example a vulnerability in *Webkit*, an HTML engine used in many browsers (e.g. Safari, Chrome, and Opera). Because Webkit is a component of other software, a vulnerability in Apple Safari might also be a Webkit vulnerability in Google Chrome.

For these reasons, to match the 'software' string when selecting sv_i would introduce unknown error in the data. We can therefore only perform a 'best effort' approach by checking that the software affected by sv_i is included in the list of software for $\forall v_i \in \mathsf{SYM}$. In this work *software* is therefore used as a 'sanitation' variable rather than a proper control.

7.2.1 Experiment run

We divide our experiment in two parts: sampling and execution. In the former we generate the samples from NVD, EDB and EKITS. In the latter we compute the relevant statistics on the samples. What follows is a textual description of these processes.

Sampling To create the samples, we first select a vulnerability v_i from SYM and set the controls according to the values of the confounding variables for v_i. Then, for each of NVD, EDB and EKITS we randomly select, with replacement, a sample vulnerability sv_i that satisfies the conditions defined by v_i. We then include sv_i in the list of selected vulnerabilities for that dataset sample. We repeat this procedure for all vulnerabilities in SYM. The sampling has been performed with the statistical tool R-CRAN [96]. Our R script to replicate the analysis is available on our Lab's webpage[2].

Execution Once we collected our samples, we compute the frequency with which each risk factor identifies a vulnerability in SYM. Our output is repre-

[2] `https://securitylab.disi.unitn.it/doku.php?id=software`

Table 7.2: Output format of our experiment.

Risk Factor level	$v \in \mathsf{SYM}$	$v \notin \mathsf{SYM}$
Above Threshold	a	b
Below Threshold	c	d

Table 7.3: Sample thresholds

CVSS ≥ 6
CVSS ≥ 9
CVSS ≥ 9 & v $\in$ EDB
CVSS ≥ 9 & v $\in$ EKITS

sented in Table 7.2. Each risk factor is defined by a CVSS threshold level t in combination with the existence of a proof-of-concept exploit ($v \in \mathsf{EDB}$) or of a black-marketed exploit ($v \in \mathsf{EKITS}$). Examples of thresholds for different risk factors are reported in Table 7.3. We run our experiment for all CVSS thresholds t_i with $i \in [1..10]$. For each risk factor we evaluate the number of vulnerabilities in the sample that fall *above* and *below* the CVSS threshold, and that are included (or not included) in SYM: the obtained table reports the count of vulnerabilities that each risk factor correctly and incorrectly identifies as 'at high risk of exploit' ($\in$ SYM) or 'at low risk of exploit' ($\notin$ SYM).

The computed values depend on the random sampling process. In an extreme case we may therefore end up, just by chance, with a sample containing only vulnerabilities in SYM and below the current threshold (i.e. $[a = 0; b = 0; c = 1277; d = 0]$). Such an effect would be likely due to chance alone. To mitigate this we repeat, for every risk factor, the whole experiment run 400 times and keep the median of the results. We choose this limit because we observed that around 300 repetitions the distribution of results is already markedly Gaussian. Any statistic reported here to be intended as the median of the generated distribution of values.

7.2.2 Parameters of the analysis

Sensitivity and specificity In the medical domain, the sensitivity of a test is the conditional probability of the test giving positive results when the illness is present. The specificity of the test is the conditional probability of the test giving negative result when there is no illness. Sensitivity and specificity are also known as True Positive Rate (TPR) and True Negatives Rate (TNR) respectively. High values for both TNR and TPR identify a good test[3]. In our context, we want to assess to what degree a positive result from our current test (the CVSS score) matches the illness (the vulnerability being actually exploited in the wild and tracked in SYM). The sensitivity and specificity measures are computed as:

$$Sensitivity = P(v\text{'s Risk factor above } t|\ v \in \mathsf{SYM}) = a/(a+c) \quad (7.1)$$

$$Specificity = P(v\text{'s Risk factor below } t|\ v \notin \mathsf{SYM}) = d/(b+d) \quad (7.2)$$

where t is the threshold. Sensitivity and specificity outline the performance of the test in identifying exploits, but say little about its effectiveness in terms of diminished risk.

Risk Reduction To understand the effectiveness of a policy we adopt an approach similar to that used in [49] to estimate the effectiveness of seat belts in preventing fatalities. In Evan's case, the 'effectiveness' was given by the difference in the probability of having a fatal car crash when wearing a seatbelt and when not wearing it ($Pr(Death\ \&\ Seat\ belt\ on) - Pr(Death\ \&\ not\ Seat\ belt\ on)$).

In our case, we measure the ability of a risk factor to predict the actual

[3]Some may prefer the False Positive Rate (FPR) to the TNR. Note that TNR=1-FPR (as in our case $d/(b+d) = 1 - b/(b+d)$). We choose to report the TNR here because 1) it has the same direction of the TPR (higher is better); 2) it facilitates the identification of the threshold with the best trade-off by intersecting TPR.

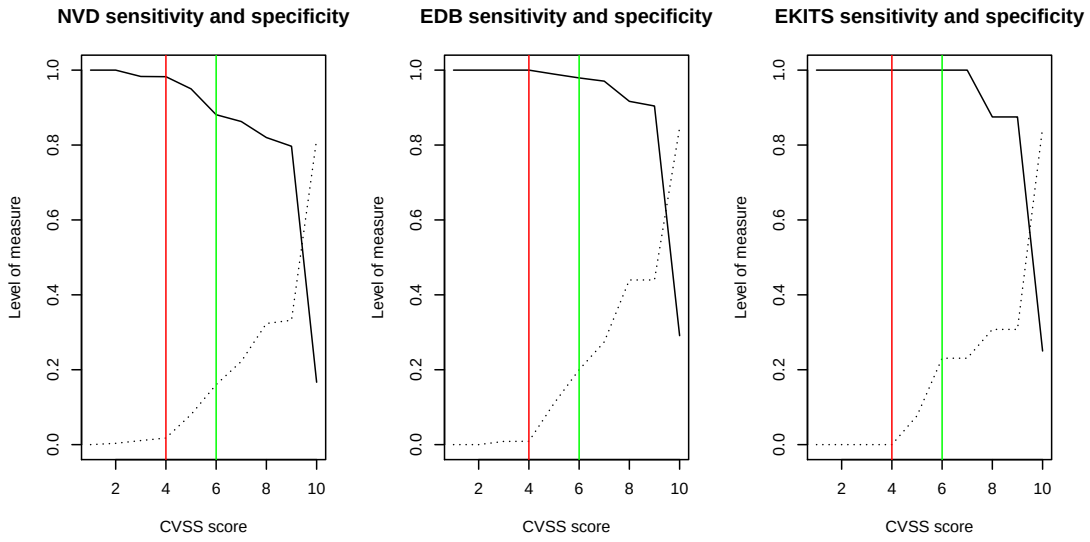


Figure 7.1: Sensitivity (solid line) and specificity (dotted line) levels for different CVSS thresholds. The red line identifies the threshold for PCI DSS compliance ($cvss = 4$). The green line identifies the threshold between `LOW` and `MEDIUM`+`HIGH` vulnerabilities ($cvss = 6$). No CVSS configuration, regardless of the inclusion of additional risk factors, achieves satisfactory levels of Specificity and Sensitivity simultaneously.

exploit in the wild. Formally, the risk reduction is calculated as:

$$RR = P(v \in \mathsf{SYM} | v\text{'s Risk factor above } t) - P(v \in \mathsf{SYM} | v\text{'s Risk factor below}$$

therefore $RR = a/(a+b) - c/(c+d)$. An high risk reduction identifies risk factors that clearly discern between high-risk and low-risk vulnerabilities, and are therefore good *decision variables* to act upon: the most effective strategy is identified by the risk factor with the highest risk reduction.

7.2.3 Results

Sensitivity and specificity Figure 7.1 reports the sensitivity and specificity levels respective to different CVSS thresholds. The solid line and the dotted line report the Sensitivity and the Specificity respectively. The vertical red line marks the CVSS threshold fixed by the PCI DSS standard ($cvss = 4$).

Table 7.4: Risk Reduction and significance levels for our risk factors PoC and BMar. Significance is indicated as follows: A **** indicates the Bonferroni-corrected equivalent of $p < 1E-4$; *** $p < 0.001$; ** $p < 0.01$; * $p < 0.05$; nothing is reported for other values.

Factor	RR	95% RR conf. int.	Significance
PoC	36%	35% ; 38%	****
BMar	46%	44% ; 48%	****

The green vertical line marks the threshold that separates LOW CVSS vulnerabilities from MEDIUM+HIGH CVSS vulnerabilities ($cvss = 6$). Unsurprisingly, low CVSS scores show a very low specificity, as most non-exploited vulnerabilities are above the threshold. With increasing CVSS thresholds, the specificity measure gets better without sensibly affecting sensitivity. The best trade-off obtainable with the sole CVSS score is achieved with a threshold of eight, where specificity grows over 30% and sensitivity sets at around 80%. To further increase the threshold causes the sensitivity measure to collapse. In EKITS, because most vulnerabilities in the black markets are exploited and their CVSS scores are high, the specificity measure can not significantly grow without collapsing sensitivity.

Risk reduction First, we analyse the significance of our risk factors alone, i.e. the significance of PoC and BMar over the patching decision. Table 7.4 reports the RR results for our risk factors alone, without considering the criticality level indicated by the CVSS score. This gives us a measure of the significance of the risk factors in the vulnerability assessment. The entailed RR is high in both cases, with BMar performing better than PoC. We therefore can conclude that the Black Markets represent a significant risk factor for vulnerability exploitation. Similarly, PoC-based policies can achieve satisfactory Risk Reduction levels at a high significance. Yet, BMar and PoC are only measures for 'likelihood' of exploitation and, considered by themselves, are not yet qualified to be employed as risk metrics. To achieve

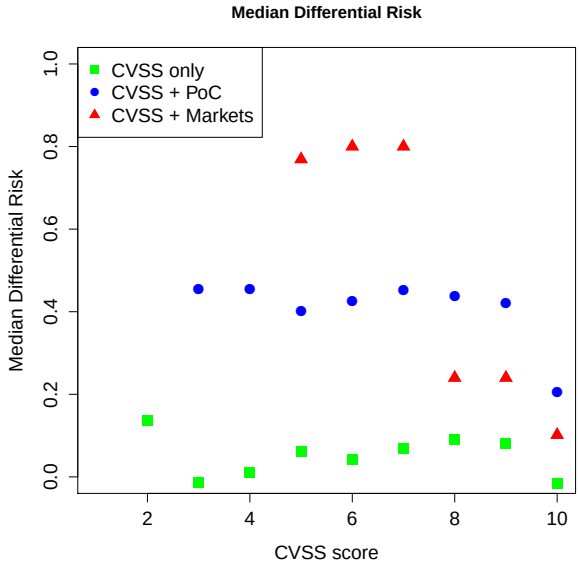


Figure 7.2: Risk reduction (RR) entailed by different risk factors. The Black Markets represent the most important risk factor with an entailed RR of up to 80%. The existence of a proof-of-concept exploit is significant as well and is stable at a 40% level. The CVSS score alone is never significant and its median RR lays in the whereabouts of 4%.

this, we couple our risk factors with the CVSS assessment on the vulnerability criticality. We expect the RR levels to increase significantly.

In Figure 7.2 we report our results on risk reduction (RR) for each risk factor coupled with all CVSS levels. The mere CVSS score (green squares), irrespectively of its threshold level, always defines a poor patching policy with very low risk reduction. The existence of a public proof-of-concept exploit confirms its significance as a risk factor, yielding higher risk reduction levels (40%). As expected, this is also an improvement over the sole use of PoC without considering vulnerability criticality. The presence of an exploit in the black markets is the most effective risk factor to consider; in the case of BMar, the maximum risk reduction (80%) is achieved at CVSS levels within the interval $[5, 7]$. Outside of these boundaries the risk factor becomes insignificant; we can conclude that attackers do not trade vulnerabilities in the black markets below a CVSS score of 5, and trade vulnerabilities above

Table 7.5: Risk Reduction for a sample of thresholds. Risk Reduction of vulnerability exploitation depending on policy and information at hand (CVSS, PoC, Markets). Significance is reported by a Bonferroni-corrected Fisher Exact test (data is sparse) for three comparison (CVSS vs CVSS+PoC vs CVSS+BMar) per experiment [29]. A **** indicates the Bonferroni-corrected equivalent of $p < 1E-4$; *** $p < 0.001$; ** $p < 0.01$; * $p < 0.05$; nothing is reported for other values. Non-significant results indicate risk factors that perform indistinguishably at marking 'high risk' vulnerabilities than random selection.

Policy Type	Policy	RR	95% RR conf. int.	Significance
Criticality-based	CVSS ≥4	1%	-35% ; 19%	
	CVSS ≥6	4%	-5% ; 12%	
	CVSS ≥9	8%	1% ; 15%	
Risk-based	CVSS ≥4 + PoC	45%	42% ; 49%	
	CVSS ≥6 + PoC	42%	38% ; 48%	****
	CVSS ≥9 + PoC	42%	36% ; 49%	****
	CVSS ≥4 + BMar	-	-	
	CVSS ≥6 + Bmar	80%	80% ; 81%	*
	CVSS ≥9 + Bmar	24%	23% ; 29%	

a CVSS of 7 irrespective of their actual CVSS level.

Table 7.5 reports the numerical Risk Reduction for a sample of thresholds. A CVSS score of four, as indicated by PCI-DSS, entails a risk reduction that is never significant, even when integrated with the PoC and BMar risk factors. In the PoC case, we have a median RR of 45%, but no significance because there is effectively no vulnerability with a PoC *and* below a CVSS threshold of 4. The CVSS threshold becomes therefore insignificant with respect to the distribution of exploits. This holds true also on the BMar case. We therefore conclude that setting a CVSS threshold of 4 has no statistical value. On the contrary, a CVSS score of six is more significant, but only when coupled with our risk factors: CVSS≥6 alone entails a Risk Reduction of 4%; the performance is slightly better, but still unsatisfactory, if the threshold is raised to nine. Overall, CVSS' Risk Reduction stays below 10% for most

thresholds. Even by considering the 95% confidence interval, we can conclude that CVSS-only based policies may be unsatisfactory from a risk-reduction point of view. Unsurprisingly, the test with the CVSS score alone results in very high p-values, that in this case testify that CVSS as a risk factor does not mark high risk vulnerabilities any better than random selection would do. We therefore conclude that *criticality-based vulnerability mitigation is ineffective in identifying vulnerabilities to patch with high priority.*

The existence of a proof-of-concept exploit (PoC) improves greatly the performance of the policy: with 'CVSS $\geq$ 6 + PoC' a RR of 42% can be achieved with very high statistical significance. This result is comparable to wearing a seat belt while driving, which entails a 43% reduction in risk [49]. The highest risk reduction (80%) is obtained by considering the existence of an exploit in the black markets. The significance for BMar with a CVSS $\geq$ 9 is below the threshold ($p = 0.19$).

7.3 Effectiveness of Risk-Based Policies

We now evaluate the effectiveness of risk-based policies by measuring the reduction in workload and the number of foiled attacks they entail. We will focus on the advantages in this terms entailed by risk-based policies as opposed to criticality-based policies.

Policy workload. Each policy may require different levels of effort to be implemented. For example, the same vulnerability could be present in hundreds of machine or could reside in a server for which a 1 hour downtime is already too much. This information is *company dependent* and therefore we can not consider it here. We discuss in Chapter 8 how the whole framework can be lifted to include this (and adjust the risk notion accordingly). We consider here a simpler proxy for cost, that is the number of vulnerabilities that should be considered by each policy (workload). The cost-effectiveness

Table 7.6: No. of vulnerabilities to fix by policy.

Policy	Workload
All	14380
$CVSS \geq 4$	13715
$CVSS \geq 6$	8341
$CVSS \geq 9$	3081
PoC	3030
P$oC + CVSS \geq 4$	3004
P$oC + CVSS \geq 6$	2416
P$oC + CVSS \geq 9$	550
BMar	58
B$Mar + CVSS \geq 4$	58
B$Mar + CVSS \geq 6$	54
B$Mar + CVSS \geq 9$	48

of a policy is then reflected in the relation between workload and the volume of attacks in the wild the policy thwarts.

Workloads for our policies over a sample of vulnerabilities are reported in Table 7.6. The full set comprises 14380 vulnerabilites, 3030 of which have a PoC and 58 are in BMar. The workloads decrease with increasing CVSS threshold as more vulnerabilities to 'ignore' fall below the CVSS level.

7.3.1 Potential of Attack (pA)

In order to better visualize and independently validate the effectiveness of the selection of policies based on risk reduction we introduce a new notion to capture the number of attacks that would be thwarted by deploying it.

In chemistry, the pH of a solution is a function of the concentration in hydrogen ions $[H^+]$. To be precise, it is an *empirical measure* of the capacity of the hydrogen ions to be involved in chemical reactions (and therefore determine the degree of acidity of a solution). Because the concentration of these ions is typically low, pH is calculated as the logarithm of the inverse

of $[H^+]$. More formally, $pH = log_{10}\frac{1}{[H^+]}$.

We define pA as an empirical measure of the *potential for attack* of a vulnerability. Specifically, we define pA as the base ten logarithm of the volume of attacks in the wild received by 10^6 machines (i.e. those sampled in the WINE dataset). We define

$$pA = log_{10}(A_v) \tag{7.4}$$

where A_v is the number of attacks observed in the wild for the vulnerability v. The reason we choose a base 10 logarithm is that this allows us to make a direct comparison of the volume of recorded attacks with the number of machines in the wild potentially affected by it. Further, this gives a more immediate intuition of the diverse order of magnitude of attacks (e.g. an attack with $pA = 6$ is ten times more common than one with $pA = 5$).[4] For example, a pA of 6 indicates that the attack has the potential to be distributed to every machine included in the dataset. In WINE pA ranges in [0..7.5]. Its distribution has two modes at $pA = 1$ and $pA = 6$. The median pA is 1.6.

7.3.2 Quantification of patching workloads and pA reduction

Table 7.7 reports the fraction of patching workload entailed by the policy and the reduction in pA. At first glance we see that most of the reported pA columns have the same value across the rows. It should be noticed that the actual values are not equal. pA is a logarithm in base 10 and it is truncated to the first decimal. It offers a bird's eye view on the attacks, eliminating most of the noise. It shows that there is very little difference in the magnitude of foiled attacks between *'high-workload'* policies and *'low-workload'* ones.

[4]Bases other than base ten could have been chosen for pA. For example, e is a base commonly used in econometrics to define likelihood measures [43]. However, this does not allow, without further transformations, for a direct comparison between different volume of attacks and attacked machines, and does not give a direct intuition of the relative distances between attacks with different pA.

Table 7.7: Workloads and reduction in pA for each policy. Risk-based policies allow for an almost complete coverage of the attack potential in the wild with a fraction of the effort entailed by a criticality-based policy.

Policy Type	Policy	Workload	Foiled pA
Criticality-based	All	100%	6.8
	CVSS $\geq$ 4	95%	6.8
	CVSS $\geq$ 6	58%	6.7
	CVSS $\geq$ 9	21%	6.7
Risk-based	PoC	21%	6.5
	PoC+ CVSS $\geq$ 4	21%	6.5
	PoC+ CVSS $\geq$ 6	17%	6.5
	PoC+ CVSS $\geq$ 9	4%	6.5
	BMar	<1%	6.3
	BMar+ CVSS $\geq$ 4	<1%	6.3
	BMar+ CVSS $\geq$ 6	<1%	6.3
	BMar+ CVSS $\geq$ 9	<1%	6.3

For example, the difference in decreased pA for a PoC-only policy and an *All* policy is only 0.3 points, but the former achieves this by addressing 80% vulnerabilities *less* than the latter: the workload is massively reduced with an only negligible loss in attack coverage.

This same observation can be generalized to all risk-based policies as compared to criticality-based ones. The case of BMar is particularly clear as with less than 1% of the original workload almost all the pA is foiled. This result provides additional support to Hypotheses 1-2 whereby the underground markets are to be considered a relevant source of risk for the final user.

From the results presented in this Chapter we conclude that *risk-based* policies for vulnerability management are possible and can lead to substantial improvements in terms of patching efficiency over current criticality-based approaches.

7.4 Discussion

In this Thesis we explored the idea of implementing risk-based policies for vulnerability management. The resulting contribution adds to current scientific literature in several ways.

1. We showed that current criticality-based vulnerability management policies are widely suboptimal in prioritising the vulnerability mitigation process. Their shortcoming is that they lack of a proper characterisation of exploitation likelihood, a fundamental part of any risk assessment.

2. We hypothesised that a significant factor for likelihood of exploitation are exploit cost and availability in the underground markets for cybercrime. These two line of research led to the following conclusions:

   (a) The attacker is rational and has incentives to re-use the same exploit until the overall number of vulnerable users drops significantly. As a consequence, the same exploit is used in subsequent attacks for more than two years before being substituted at large with a new one. Similarly, new attacks arrive quicker as the pace of software updates increases.

   (b) Contrary to present claims in the scientific literature, the underground markets are mature and economically sound. Current underground markets show strong internal regulation that incentivizes fair trading, and indeed the traded technology works well and reliably against software configurations spanning as many as 8 years. We developed a two-stage model of the underground markets and showed that the economic principles over which they are founded are sound.

3. Building on these conclusions, we develop a methodology based on the notion of case-control studies to measure the reduction in risk entailed by

current criticality-based policies and two risk-based policies. On the one hand, we show that criticality-based policies are statistically equivalent to 'randomly picking' vulnerabilities to patch. On the other, we show that the exploitation factors discussed above are indeed significant for vulnerability management and can lead to risk reductions as high as 80% (as opposed to current practices' 4%).

4. We showed how risk-based policies enable vulnerability management practices that get rid of the almost totality of risk in the wild by addressing a few vulnerabilities only. Our methodology is therefore suitable to guide the prioritisation of vulnerability mitigation actions.

Conclusion 5 *The results of our case-control study and the validation example confirm that risk-based policies significantly improve over criticality-based ones. We therefore accept Hypothesis 3. As expected from the analysis provided in Chapter 6, we find that risk-based policies based on cybercrime black markets benefit from the multiplicative factor they enable. We therefore also accept the Corollary to Hypothesis 3.*

Chapter 8

Limitations and Future Work Directions

In this Chapter we discuss some limitations of this work and how it could be extended to account for additional considerations on the value and costs associated to the vulnerable system. Further, we outline what we believe are interesting venues for future research on these same lines, in particular from a policy perspective.

8.1 Limitations and Extensions

The results on risk-based policies presented in this Thesis are not accounting for additional variables such as the value of the vulnerable system, that has a clear impact on the level of acceptable risk. In the current formalization of risk reduction we do not consider the fact that a company has typically many instances of software and therefore many instances of the software's vulnerabilities, and that different companies may effectively face different costs according the their specific environment. Rather, because RR is a measure conditional on the existence of at least one exploit, it provides an "upper bound" of risk reduction.

It is however impossible for us to consider these costs explicitly in our case-

control experiment for two reasons: 1) the estimation is necessarily bounded to a particular case-study, which can only reproduce our results after a *general validation* is available (i.e. the work presented here); 2) even in a case-study scenario, a correct estimation of these costs may be very difficult to calculate. Therefore, in our study we have simply considered cost to linearly increase with the number of vulnerabilities, i.e. every vulnerability has a unit cost.

To tailor our results to a more case-oriented application, a more appropriate cost estimation should account for the occurrences of a vulnerability v in n_v systems:

$$Cost_{mult} = \sum_{v \in Selected} n_v \tag{8.1}$$

In this case we should also revise the notion of *risk reduction with multiple occurrences*, because we should consider also the number of occurrences of each vulnerability v in the n_v systems.

$$RR_{multi} \frac{\sum_{v \in Attack \cap Selected} n_v}{\sum_{v \in Selected} n_v} - \frac{\sum_{v \in Attack \cap \overline{Selected}} n_v}{\sum_{v \in \overline{Selected}} n_v} \tag{8.2}$$

The value of RR_{multi} would therefore be company specific since it depends on the number of instances of the installed software base.

Calculating the risk reduction with this formula is again an approximation. It assumes that all instances of software where the vulnerability is present will be affected by the patching policy and that they will all be equally attacked. The former assumption is correct (if the policy is implemented correctly), but the latter is an approximation. In practice only a subset of the systems with the vulnerability will be attacked (albeit they are all potentially attackable). This approximation is conservative from a security perspective: it overestimates the risk reduction that can be obtained when we decide to patch a vulnerability that is present in many systems but could be attacked only in some of them.

Another variable worth mentioning is "infrastructural impact" , i.e. the cost of having a critical system impacted by an attack. A vulnerability could affect a mission critical system at the core of the corporation or a computer in an obscure subsidiary. Yet, by itself the vulnerability has no "infrastructural impact". It is the compromise of the system on which the vulnerability is present that can lead to a more or less severe cost. "Infrastructural impact" should *not* therefore be considered in the actual calculation of risk reduction, but rather when deciding which is the appropriate risk level for the systems under consideration. In this sense the RR_{mult} can be normalized by a system criticality estimation when limiting its evaluation on a particular set of systems. A company could therefore decide that a risk reduction of 50% is a good trade-off for the desktop of the subsidiary while a moderate 10% reduction is worth the money for the mission critical system.

8.2 Future Research Venues

This work outlined on the one hand the importance of attacker economics in the general threat scenario, and on the other how this could be exploited to design better vulnerability management practices.

The main point behind this body of work is that attackers are rational. Rationality and market activities have a dual effect in our context: first, they enable the attacker in more proficient and focused attack capabilities. Second, and more interestingly for us, it makes the attacker's decisions *predictable* to a degree: if the attacker has to act rationally, economic theory will point in the direction of the attacker's next step. For example, an attacker that has to decide which vulnerability to massively exploit next, will necessarily choose the one providing the highest return on investment.

Similarly, the economic environments that empower the attacker are based, as shown in this Thesis, on well-known economic principles that ultimately

make it work. From this perspective, it is possible to leverage this knowledge to drive future international policies in the direction of 'discouraging' the formation of these markets. The attempt of influencing the convenience of criminal activities through policies is certainly not new in itself, but the cybercrime markets represent a brand new and interesting field that is yet unexplored.

The idea of designing 'rational vulnerability management' practices can and should go beyond the mere 'application of a patch': an interesting direction would be to define policies to *develop* vulnerability patches, i.e. policies to decide, on the vendor's side, which vulnerabilities to address first. This is a different issue from that of installing a new patch on an existent system: the system administrator (may) know his source of threat, while for a developer shipping the software to hundred of thousands customers this is not possible (as the threat is ultimately on the customer and not the developer). In other words, there is a balance in positive and negative 'externalities' that the patching decision can create. This certainly requires future research to be carried in this direction.

Finally, in this work we have considered only the 'general' attacker that aims at masses rather than on specific targets. Extending a 'risk-based' approach to the latter scenario may be, however, a pointless exercise: there is an inherent asymmetry there between the attacker and the defender whereby the attacker knows *more* about the affected system than the defender does. For example, if the attacker exploits a 0-day vulnerability or a non-default configuration that makes an otherwise harmless vulnerability reachable there is nothing that the defender may really do. Unfortunately, software vulnerabilities are not going to disappear from code [89], and therefore this problem is unlikely to be solved in the foreseeable future. A different approach may instead be more sensible to apply. Rather than focussing on risk mitigation, the defender may accept that something he *cannot* avoid, and be instead

prepared to reacting quickly and efficiently to an attack. In the practitioner community this is an often-acclaimed concept, but it is hardly formalised and remained largely untouched in the scientific literature.

A particular type of 'dedicated attacker' is a 'governmental attacker', i.e. an organisation or person working for a governmental agency that deploys a certain attack for monitoring or surveillance purposes. Without discussing here the ethical issues attached to this practice, it is clear that establishing a sensible policy to govern and limit this capability is of central importance for the future society. Establishing a field of research that aims at filling this regulation gap requires a clear understanding of the technical, economic, and governance aspects of the problem.

Chapter 9

Conclusion

The contribution of this Thesis is twofold. On the one hand, it provides a previously unexplored perspective over the economic environment in which attackers operate. In particular, by infiltrating and studying the HackMarket.ru market and testing real attack tools in our MalwareLab we were able to draw a picture of the resources available to the attacker that starts from the economic infrastructure supporting him or her, to the technical quality of the goods in his or her hands. Importantly, we showed that the model underlying cybercrime market operations is sound form an economic perspective, and there is therefore no good reason to believe these markets will stop operating anytime soon without an external intervention (e.g. by means of international policies).

The economic rationality of the attacker is also at the core of the definition of a new attacker model, the 'Work-Averse Attacker' model, whereby the attacker's decision of massively deploying a new exploit depends on the expected utility of the new exploit relative to the already present one.

We argued that these considerations on the economic nature of the attacker are the key enablers for more efficient vulnerability management strategies that account for the *risk* represented by a vulnerability rather than merely its technical severity. We tested this hypothesis by running a case

control study over vulnerabilities and exploits in the wild, and showed that indeed a risk-based approach enables for much more efficient vulnerability management. Further, we showed that this efficiency translates in vulnerability management strategies that address few vulnerabilities only and are still able to address the overwhelming majority of risk in the wild. With this methodology and taking in account our considerations, an organisation may ultimately be able to design more sensible vulnerability management strategies that are easy to communicate and effective to enforce.

Bibliography

[1] Google reward program. [online] http://www.google.com/about/appsecurity/reward-program/.

[2] Schneier: Security is as strong as its weakest link, 2005. [online] https://www.schneier.com/blog/archives/2005/12/weakest_link_se.html.

[3] Software vulnerability disclosure: The chilling effect, 2007. [online] http://www.csoonline.com/article/2121727/application-security/software-vulnerability-disclosure--the-chilling-effect.html.

[4] Shopping for zero-days: A price list for hackers' secret software exploits, 2012. [online] http://www.forbes.com/sites/andygreenberg/2012/03/23/shopping-for-zero-days-an-price-list-for-hackers-secret-software-

[5] Cisco vulnerability disclosure policy, 2014. [online] http://www.cisco.com/web/about/security/psirt/security_vulnerability_policy.html#asr.

[6] Google project zero, 2014. [online] http://googleprojectzero.blogspot.se.

[7] Defeat the casual attacker first!!, 2015. [online] `http://blogs.gartner.com/anton-chuvakin/2015/01/28/defeat-the-casual-attacker-first`.

[8] Meet paunch: The accused author of the blackhole exploit kit, 2015. [online] `http://krebsonsecurity.com/2013/12/meet-paunch-the-accused-author-of-the-blackhole-exploit-kit/`.

[9] NIST National Vulnerability Database (NVD), 2015. [online] `http://nvd.nist.gov`.

[10] Open Sourced Vulnerability Database (OSVDB), 2015. [online] `http://osvdb.org`.

[11] An overview of exploit packs (update 22) jan 2015, 2015. [online] `http://contagiodump.blogspot.it/2010/06/overview-of-exploit-packs-update.html`.

[12] When google squares off with microsoft on bug disclosure, only users lose, 2015. [online] `http://arstechnica.com/security/2015/01/google-sees-a-bug-before-patch-tuesday-but-windows-users-remain-vulne`

[13] George A. Akerlof. The market for "lemons": Quality uncertainty and the market mechanism. *The Quarterly Journal of Economics*, 84:pp. 488–500, 1970.

[14] Miriam R Albert. E-buyer beware: Why online auction fraud should be regulated. *American Business Law Journal*, 39(4):575–644, 2002.

[15] Omar Alhazmi and Yashwant Malaiya. Modeling the vulnerability discovery process. In *Proceedings of the 16th IEEE International Symposium on Software Reliability Engineering (ISSRE'05)*, pages 129–138, 2005.

[16] Omar Alhazmi and Yashwant Malaiya. Application of vulnerability discovery models to major operating systems. *IEEE Transactions on Reliability*, 57(1):14–22, 2008.

[17] Luca Allodi. Attacker economics for internet-scale vulnerability risk assessment. In *Presented as part of the 6th USENIX Workshop on Large-Scale Exploits and Emergent Threats*. USENIX, 2013.

[18] Luca Allodi, Vadim Kotov, and Fabio Massacci. Malwarelab: Experimentation with cybercrime attack tools. In *Proceedings of the 2013 6th Workshop on Cybersecurity Security and Test*, 2013.

[19] R. Anderson, C. Barton, R. Böhme, R. Clayton, M.J.G. van Eeten, M. Levi, T. Moore, and S. Savage. Measuring the cost of cybercrime. In *Proceedings of the 11th Workshop on Economics and Information Security*, 2012.

[20] A. Arora, R. Krishnan, A. Nandkumar, R. Telang, and Y. Yang. Impact of vulnerability disclosure and patch availability-an empirical analysis. In *Proceedings of the 3rd Workshop on Economics and Information Security*, 2004.

[21] Ashish Arora, Ramayya Krishnan, Rahul Telang, and Yubao Yang. An empirical analysis of software vendors' patch release behavior: Impact of vulnerability disclosure. *Information Systems Research*, 21(1):115–132, March 2010.

[22] Ashish Arora, Ramayya Krishnan, Rahul Telang, and Yubao Yang. An empirical analysis of software vendors; patch release behavior: Impact of vulnerability disclosure. *Information Systems Research*, 21(1):115–132, 2010.

[23] Ashish Arora, Rahul Telang, and Hao Xu. Optimal policy for software vulnerability disclosure. *Management Science*, 54(4):642–656, 2008.

[24] Hadi Asghari, Michel Van Eeten, Axel Arnbak, and Nico Van Eijk. Security economics in the https value chain. *Available at SSRN 2277806*, 2013.

[25] W. Baker, M. Howard, A. Hutton, and C. David Hylender. 2012 data breach investigation report. Technical report, Verizon, 2012.

[26] Jonell Baltazar. More traffic, more money: Koobface draws more blood. Technical report, TrendLabs, 2011.

[27] Johannes M Bauer and Michel JG Van Eeten. Cybersecurity: Stakeholder incentives, externalities, and policy options. *Telecommunications Policy*, 33(10):706–719, 2009.

[28] Leyla Bilge and Tudor Dumitras. Before we knew it: an empirical study of zero-day attacks in the real world. In *Proceedings of the 19th ACM Conference on Computer and Communications Security (CCS'12)*, pages 833–844. ACM, 2012.

[29] J Martin Bland and Douglas G Altman. Multiple significance tests: the bonferroni method. 310(6973):170, 1995.

[30] Mehran Bozorgi, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. Beyond heuristics: Learning to classify vulnerabilities and predict exploits. In *Proceedings of the 16th ACM International Conference on Knowledge Discovery and Data Mining*, July 2010.

[31] B. Brykczynski and R.A. Small. Reducing internet-based intrusions: Effective security patch management. *Software, IEEE*, 20(1):50–57, Jan 2003.

[32] Ahto Buldas, Peeter Laud, Jaan Priisalu, Mart Saarepera, and Jan Willemson. Rational choice of security measures via multi-parameter attack trees. In Javier Lopez, editor, *Proceedings of the 1st International Workshop on Critical Information Infrastructures Security*, volume 4347 of *Lecture Notes in Computer Science*, pages 235–248. Springer Berlin / Heidelberg, 2006.

[33] Mary M Calkins. My reputation always had more fun than me: The failure of ebay's feedback model to effectively prevent online auction fraud. *Rich. JL & Tech.*, 7:33–34, 2001.

[34] Hasan Cavusoglu, Huseyin Cavusoglu, and Jun Zhang. Security patch management: Share the burden or share the damage? *Management Science*, 54(4):657–670, 2008.

[35] Huseyin Cavusoglu, Hasan Cavusoglu, and Jun Zhang. Economics of security patch management. In *WEIS*, 2006.

[36] Pei-yu Chen, Gaurav Kataria, and Ramayya Krishnan. Correlated failures, diversification, and information security risk management. *MIS Quaterly-Management Information Systems*, 35(2):397–422, 2011.

[37] Erika Chin, Adrienne Porter Felt, Vyas Sekar, and David Wagner. Measuring user confidence in smartphone security and privacy. In *Proceedings of the Eighth Symposium on Usable Privacy and Security*, page 1. ACM, 2012.

[38] Steve Christey and Brian Martin. Buying into the bias: why vulnerability statistics suck. `https://www.blackhat.com/us-13/archives.html#Martin`, July 2013.

[39] Sandy Clark, Stefan Frei, Matt Blaze, and Jonathan Smith. Familiarity breeds contempt: the honeymoon effect and the role of legacy code in

zero-day vulnerabilities. In *Proceedings of the 26th Annual Computer Security Applications Conference*, pages 251–260, 2010.

[40] PCI Council. Pci dss requirements and security assessment procedures, version 2.0., 2010.

[41] Bill Curtis, Herb Krasner, and Neil Iscoe. A field study of the software design process for large systems. *Commun. ACM*, 31(11):1268–1287, November 1988.

[42] D. Dagon, C. Zou, and W. Lee. Modeling botnet propagation using time zones. In *Proceedings of the 13th Annual Network and Distributed System Security Symposium*, 2006.

[43] Russell Davidson and James G MacKinnon. *Econometric theory and methods*, volume 5. Oxford University Press New York, 2004.

[44] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198 – 208, mar 1983.

[45] Richard Doll and A Bradford Hill. Smoking and carcinoma of the lung. *British Medical Journal*, 2(4682):739–748, 1950.

[46] Tudor Dumitras and Petros Efstathopoulos. Ask wine: are we safer today? evaluating operating system security through big data analysis. In *Proceeding of the 2012 USENIX Workshop on Large-Scale Exploits and Emergent Threats*, LEET'12, pages 11–11, 2012.

[47] Tudor Dumitras and Darren Shou. Toward a standard benchmark for computer security research: The worldwide intelligence network environment (wine). In *Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*, pages 89–96. ACM, 2011.

[48] Kathleen M Eisenhardt. Agency theory: An assessment and review. *Academy of management review*, 14(1):57–74, 1989.

[49] L. Evans. The effectiveness of safety belts in preventing fatalities. *Accident Analysis & Prevention*, 18(3):229–241, 1986.

[50] FBI. Internet crime report 2013. Technical report, Internet Crime Complaint Center, 2013.

[51] Matthew Finifter, Devdatta Akhawe, and David Wagner. An empirical study of vulnerability rewards programs. In *Presented as part of the 22nd USENIX Security Symposium (USENIX Security 13)*, pages 273–288, Washington, D.C., 2013. USENIX.

[52] J. Franklin, V. Paxson, A. Perrig, and S. Savage. An inquiry into the nature and causes of the wealth of internet miscreants. In *Proceedings of the 14th ACM Conference on Computer and Communications Security (CCS'07)*, pages 375–388, 2007.

[53] Stefan Frei, Dominik Schatzmann, Bernhard Plattner, and Brian Trammell. Modeling the security ecosystem - the dynamics of (in)security. In Tyler Moore, David Pym, and Christos Ioannidis, editors, *Economics of Information Security and Privacy*, pages 79–106. Springer US, 2010.

[54] Thomas Gerace and Huseyin Cavusoglu. The critical elements of the patch management process. *Commun. ACM*, 52(8):117–121, August 2009.

[55] Lawrence A. Gordon and Martin P. Loeb. The economics of information security investment. *ACM Transactions on Information and System Security*, 5(4):438–457, 2002.

[56] Avner Greif. Contract enforceability and economic institutions in early trade: The maghribi traders' coalition. *The American Economic Review*, pages 525–548, 1993.

[57] Mark Greisiger. Cyber liability & data breach insurance claims a study of actual claim payouts. Technical report, NetDiligence, 2013.

[58] Chris Grier, Lucas Ballard, Juan Caballero, Neha Chachra, Christian J. Dietrich, Kirill Levchenko, Panayiotis Mavrommatis, Damon McCoy, Antonio Nappa, Andreas Pitsillidis, Niels Provos, M. Zubair Rafique, Moheeb Abu Rajab, Christian Rossow, Kurt Thomas, Vern Paxson, Stefan Savage, and Geoffrey M. Voelker. Manufacturing compromise: the emergence of exploit-as-a-service. In *Proceedings of the 19th ACM Conference on Computer and Communications Security (CCS'12)*, pages 821–832. ACM, 2012.

[59] Julian B. Grizzard, Vikram Sharma, Chris Nunnery, Brent ByungHoon Kang, and David Dagon. Peer-to-peer botnets: overview and case study. In *Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, 2007.

[60] Salim Hariri, Guangzhi Qu, Tushneem Dharmagadda, Modukuri Ramkishore, and Cauligi S Raghavendra. Impact analysis of faults and attacks in large-scale networks. *IEEE Security & Privacy*, (5):49–54, 2003.

[61] Trevor Hastie and Robert Tibshirani. Generalized additive models. *Statistical science*, pages 297–310, 1986.

[62] C. Herley. When does targeting make sense for an attacker? *IEEE Security and Privacy*, 11(2):89–92, 2013.

[63] C. Herley and D. Florencio. Nobody sells gold for the price of silver: Dishonesty, uncertainty and the underground economy. *Economics of Information Security and Privacy*, 2010.

[64] Cormac Herley. Why do nigerian scammers say they are from nigeria? In *Proceedings of the 11th Workshop on Economics and Information Security*, 2012.

[65] Thomas J Holt, Deborah Strumsky, Olga Smirnova, and Max Kilger. Examining the social networks of malware writers and hackers. *International Journal of Cyber Criminology*, 6(1):891–903, 2012.

[66] Andrei Homescu, Steven Neisius, Per Larsen, Stefan Brunthaler, and Michael Franz. Profile-guided automated software diversity. In *2013 IEEE/ACM International Symposium on Code Generation and Optimization (CGO)*, pages 1–11. IEEE, 2013.

[67] M. Howard, J. Pincus, and J.M. Wing. Measuring relative attack surfaces. *Computer Security in the 21st Century*, pages 109–137, 2005.

[68] Michael Howard, Jon Pincus, and Jeannette M. Wing. Measuring relative attack surfaces. In *Proceedings of Workshop on Advanced Developments in Software and Systems Security*, 2003.

[69] A.E. Howe, I. Ray, M. Roberts, M. Urbanska, and Z. Byrne. The psychology of security for the home computer user. In *Proceedings of the 33rd IEEE Symposium on Security and Privacy*, pages 209–223, 2012.

[70] Group IB. State and trends of the russian digital crime market. Technical report, Group IB, 2011.

[71] Christos Ioannidis, David Pym, and Julian Williams. Information security trade-offs and optimal patching policies. *European Journal of Operational Research*, 216(2):434 – 444, 2011.

[72] Chris Kanich, Christian Kreibich, Kirill Levchenko, Brandon Enright, Geoffrey M. Voelker, Vern Paxson, and Stefan Savage. Spamalytics: an empirical analysis of spam marketing conversion. In *Proceedings of the 15th ACM Conference on Computer and Communications Security (CCS'08)*, CCS '08, pages 3–14. ACM, 2008.

[73] Chaim Kaufmann. Threat inflation and the failure of the marketplace of ideas: The selling of the iraq war. *International Security*, 29(1):5–48, 2004.

[74] Barbara Kitchenham, Lesley Pickard, and Shari Lawrence Pfleeger. Case studies for method and tool evaluation. *IEEE Software*, 12(4):52–62, 1995.

[75] Vadim Kotov and Fabio Massacci. Anatomy of exploit kits. preliminary analysis of exploit kits as software artefacts. In *Proc. of ESSoS 2013*, 2013.

[76] Thomas Kurt, Grier Chris, Paxson Vern, and Song Dawn. Suspended accounts in retrospect:an analysis of twitter spam. In *Proceedings of the ACM 2011 Internet Measurement Conference*. ACM, 2011.

[77] M86 Labs. Security labs report july-december 2011 recap. Technical report, M86 Security Labs, 2011.

[78] J.D. McCalley, V. Vittal, and N. Abi-Samra. An overview of risk based security assessment. In *Power Engineering Society Summer Meeting, 1999. IEEE*, volume 1, pages 173–178 vol.1, Jul 1999.

[79] Peter Mell, Karen Scarfone, and Sasha Romanosky. A complete guide to the common vulnerability scoring system version 2.0. Technical report, FIRST, Available at `http://www.first.org/cvss`, 2007.

[80] Mikhail I Melnik and James Alm. Does a seller's ecommerce reputation matter? evidence from ebay auctions. *The journal of industrial economics*, 50(3):337–349, 2002.

[81] C. Miller. The legitimate vulnerability market: Inside the secretive world of 0-day exploit sales. In *Proceedings of the 6th Workshop on Economics and Information Security*, 2007.

[82] Marti Motoyama, Damon McCoy, Stefan Savage, and Geoffrey M. Voelker. An analysis of underground forums. In *Proceedings of the ACM 2011 Internet Measurement Conference*, 2011.

[83] Mendes Naaliel, Duraes Joao, and Madeira Henrique. Security benchmarks for web serving systems. In *Proceedings of the 25th IEEE International Symposium on Software Reliability Engineering (ISSRE'14)*, 2014.

[84] Antonio Nappa, Richard Johnson, Leyla Bilge, Juan Caballero, and Tudor Dumitras. The attack of the clones: A study of the impact of shared code on vulnerability patching. 2015.

[85] Kartik Nayak, Daniel Marino, Petros Efstathopoulos, and Tudor Dumitraş. Some vulnerabilities are different than others. In *Proceedings of the 17th International Symposium on Research in Attacks, Intrusions and Defenses*, pages 426–446. Springer, 2014.

[86] Viet Hung Nguyen and Fabio Massacci. An independent validation of vulnerability discovery models. In *Proceeding of the 7th ACM Sympo-*

sium on Information, Computer and Communications Security (ASIACCS'12), 2012.

[87] Russian Ministry of Internal Affairs (MVD). Arrested 13 members of criminal society who "earned" 70m rubles with internet virus. `http://mvd.ru/news/item/1387267/`, December 2013.

[88] Hamed Okhravi and D Nicol. Evaluation of patch management strategies. *International Journal of Computational Intelligence: Theory and Practice*, 3(2):109–117, 2008.

[89] Daniela Oliveira, Marissa Rosenthal, Nicole Morin, Kuo-Chuan Yeh, Justin Cappos, and Yanyan Zhuang. It's the psychology stupid: How heuristics explain software vulnerabilities and how priming can illuminate developer's blind spots. In *Proceedings of the 30th Annual Computer Security Applications Conference*, pages 296–305. ACM, 2014.

[90] Andy Ozment. Bug auctions: Vulnerability markets reconsidered. In *Third Workshop on the Economics of Information Security*, 2004.

[91] Andy Ozment. The likelihood of vulnerability rediscovery and the social utility of vulnerability hunting. In *Proceedings of the 4th Workshop on Economics and Information Security*, 2005.

[92] Ponemon. The state of risk-based security management. Technical report, Ponemon / Tripwire, 2013.

[93] Niels Provos, Panayiotis Mavrommatis, Moheeb Abu Rajab, and Fabian Monrose. All your iframes point to us. In *Proceedings of the 17th USENIX Security Symposium*, pages 1–15, 2008.

[94] Niels Provos, Dean McNamee, Panayiotis Mavrommatis, Ke Wang, and Nagendra Modadugu. The ghost in the browser analysis of web-based

malware. In *Proceedings of the first conference on First Workshop on Hot Topics in Understanding Botnets*, pages 4–4, 2007.

[95] Stephen D. Quinn, Karen A. Scarfone, Matthew Barrett, and Christopher S. Johnson. Sp 800-117. guide to adopting and using the security content automation protocol (scap) version 1.0. Technical report, National Institute of Standards & Technology, 2010.

[96] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.

[97] R.J. Ratterman, R. Maltzman, and J.D. Knepfle. Determining a community rating for a user using feedback ratings of related users in an electronic environment, 2000. US Patent 8,290,809.

[98] Paul Resnick and Richard Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system. *Advances in applied microeconomics*, 11:127–157, 2002.

[99] Colin Robson. *Real world research*, volume 2. Blackwell publishers Oxford, 2002.

[100] Sasha Romanosky. Email exchange with author. Personal email communication, July 2012.

[101] Christian Rossow, Christian J. Dietrich, Chris Grier, Christian Kreibich, Vern Paxson, Norbert Pohlmann, Herbert Bos, and Maarten van Steen. Prudent practices for designing malware experiments: Status quo and outlook. In *Proceedings of the 33rd IEEE Symposium on Security and Privacy*, 2012.

[102] Per Runeson and Martin Höst. Guidelines for conducting and reporting case study research in software engineering. *Empirical Software Engineering*, 14(2):131–164, 2009.

[103] Karen Scarfone and Peter Mell. An analysis of cvss version 2 vulnerability scoring. In *Proceedings of the 3rd International Symposium on Empirical Software Engineering and Measurement*, pages 516–525, 2009.

[104] Bruce Schneier. `https://www.schneier.com/blog/archives/2012/02/verisign_hacked.html`.

[105] Guido Schryen. A comprehensive and comparative analysis of the patching behavior of open source and closed source software vendors. In *Proceedings of the 2009 Fifth International Conference on IT Security Incident Management and IT Forensics*, IMF '09, pages 153–168, Washington, DC, USA, 2009. IEEE Computer Society.

[106] Edward J Schwartz, Thanassis Avgerinos, and David Brumley. Q: Exploit hardening made easy. In *USENIX Security Symposium*, 2011.

[107] Edoardo Serra, Sushil Jajodia, Andrea Pugliese, Antonino Rullo, and VS Subrahmanian. Pareto-optimal adversarial defense of enterprise systems. *ACM Transactions on Information and System Security (TISSEC)*, 17(3):11, 2015.

[108] Muhammad Shahzad, Muhammad Zubair Shafiq, and Alex X. Liu. A large scale exploratory analysis of software vulnerability life cycles. In *Proceedings of the 34th International Conference on Software Engineering*, pages 771–781. IEEE Press, 2012.

[109] Herbert A Simon. Theories of decision-making in economics and behavioral science. *The American Economic Review*, 49(3):253–283, 1959.

[110] Kevin Z Snow, Fabian Monrose, Lucas Davi, Alexandra Dmitrienko, Christopher Liebchen, and Ahmad-Reza Sadeghi. Just-in-time code

reuse: On the effectiveness of fine-grained address space layout randomization. In *Security and Privacy (SP), 2013 IEEE Symposium on*, pages 574–588. IEEE, 2013.

[111] Brett Stone-Gross, Marco Cova, Bob Gilbert, Richard Kemmerer, Christopher Kruegel, and Giovanni Vigna. Analysis of a botnet takeover. *IEEE Sec. & Priv. Mag.*, 9(1):64–72, 2011.

[112] Symantec. *Analysis of Malicious Web Activity by Attack Toolkits*. Symantec, Available on the web at http://www.symantec.com/threatreport/topic.jsp?id=threat_activity_trends&aid=analysis_of_malicious_web_activity, online edition, 2011. Accessed on June 1012.

[113] Symantec. Symantec corporation internet security threat report 2013. Technical Report 18, 2012 Trends, April 2013.

[114] Cisco Systems. Cisco 2015 annual security report. Technical report, Cisco, 2015.

[115] P.A. Taylor. *Hackers: crime in the digital sublime*. Psychology Press, 1999.

[116] CVSS SIG Team. Common vulnerability scoring system v3.0: Specification document. Technical report, First.org, 2015.

[117] Rahul Telang and Sunil Wattal. An empirical analysis of the impact of software vulnerability announcements on firm stock price. *, IEEE Transactions on Software Engineering*, 33(8):544–557, 2007.

[118] W. Tirenin and D. Faatz. A concept for strategic cyber defense. In *Military Communications Conference Proceedings, 1999. MILCOM 1999. IEEE*, volume 1, pages 458–463 vol.1, 1999.

[119] Jean Tirole. Cognition and incomplete contracts. *The American Economic Review*, 99:265–294, 2009.

[120] O Turgeman-Goldschmidt. Hackers' accounts: Hacking as a social entertainment. *Social Science Computer Review*, 23(1):8, 2005.

[121] Michel Van Eeten and Johannes Bauer. Economics of malware: Security decisions, incentives and externalities. Technical report, OECD, 2008.

[122] Verizon. Verizon 2014 pci compliance report. Technical report, Verizon Enterprise, 2014.

[123] Verizon. Pci compliance report. Technical report, Verizon Enterprise, 2015.

[124] Lingyu Wang, Tania Islam, Tao Long, Anoop Singhal, and Sushil Jajodia. An attack graph-based probabilistic security metric. In *Proceedings of the 22nd IFIP WG 11.3 Working Conference on Data and Applications Security*, volume 5094 of *Lecture Notes in Computer Science*, pages 283–296. Springer Berlin / Heidelberg, 2008.

[125] Rick Wash. Folk models of home computer security. In *Proceedings of the Sixth Symposium on Usable Privacy and Security*, 2010.

[126] Halbert White. A heteroskedasticity-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica*, 48(4):pp. 817–838, 1980.

[127] Branden R Williams and Anton Chuvakin. *PCI Compliance: Understand and implement effective PCI data security standard compliance.* Syngress Elsevier, 2012.

[128] Claes Wohlin, Per Runeson, Martin Höst, Magnus C Ohlsson, Björn Regnell, and Anders Wesslén. *Experimentation in software engineering*. Springer Science & Business Media, 2012.

[129] Sung-Whan Woo, Omar Alhazmi, and Yashwant Malaiya. Assessing vulnerabilities in Apache and IIS HTTP servers. In *Proceedings of the 2nd IEEE International Symposium on Dependable, Autonomic and Secure Computing*, 2006.

[130] Michael Yip, Nigel Shadbolt, and Craig Webber. Why forums? an empirical analysis into the facilitating factors of carding forums. 2013.