Characterising 0-Day Exploit Brokers

Matthias Dellago
University of Innsbruck, Austria

Daniel W. Woods
Univesity of Edinburgh, UK

Andrew C. Simpson
University of Oxford, UK

June 2, 2022

Abstract

0-day brokers are market makers who serve both adversaries seeking to exploit computer systems and researchers who develop the means to do so. This involves searching for buyers/sellers, negotiating prices and contracts, and monitoring the contract. In this paper we characterise the search aspect of 0-day broking. We extracted longitudinal data on two brokers who list prices on a public website and then plotted how the price of different types of exploit and targeted systems changed over time. As the data is not updated sufficiently regularly or frequently to build a time-series model, we conducted a regression analysis of the most recent snapshot of prices. The results suggest that properties of the exploit (e.g. the functionality it achieves) provide the most explanatory power, and that the system targeted by the exploit provides less explanatory power. We compare the price of exploit to three metrics (number of CVEs, detected 0-days, and user base) over time. Finally, we discuss what inferences we can make about systems security and the operations of adversaries, hypothesising a trade-off between secrecy and the competitiveness of the supply-side. 0-day brokers who publicly advertise prices offer cheap exploits but little secrecy.

This paper will be presented at the 21^{st} Workshop on the Economics of Information Security (WEIS'22) in Tulsa, Oklahoma, USA.

1 Introduction

The idea that computer security vulnerabilities can be traded has a long history [1, 2, 3, 4]. Vulnerability information is bought and sold via many business models and institutional structures. Theoretical contributions have considered schemes including bug auctions [2], bug bounties [5], exploit derivatives [6], vulnerability brokers [7], and cyber cat bonds [8]. The associated institutional differences are relevant to security economics, and, as such, we should continue to study what institutional details become common practice.

Take, for example, bug bounties and bug auctions. While both facilitate payments to researchers who can disclose a novel vulnerability, they do so with different market structures. A competitive auction leads to multiple competing bids, whereas researchers only receive bids from a single vendor in a bug bounty program. This helps to explain why the complaints about bug bounty programs sometimes sound like the critiques of monopsony—Apple's program, for example, was described as "a bug bounty program where the house always wins" [9]. This suggests the institutional details impact incentives for security research.

Institutional details also determine what inferences can be made about the security of the associated products and services. Bug bounties and vulnerability brokers only trade existing vulnerabilities, whereas exploit derivatives are traded in anticipation of a future exploit. As a result, exploit derivative trades embody private beliefs about the likelihood of developing exploits of specific systems [6]. These details influence the inference framework that can be applied to observations of market data.

Exploit brokers differ from the aforementioned schemes. There is no secondary market as in exploit derivatives [6] or cyber cat bonds [8]—one cannot transact with 0-day exploit brokers unless one holds or wants to hold an exploit. The broker buys exploits of multiple vendors unlike bug bounties that focus on the vendor's only, and there is no competitive bidding (unlike in bug auctions) unless a researcher collects quotes from multiple brokers. To see the difference to vulnerability brokers, consider that exploits are software engineering projects that are evaluated on properties such as speed and reliability and must be maintained on an ongoing basis [10], whereas a pure vulnerability is a one-time information good.

Rather than propose a theoretical model of exploit brokers, this paper provides empirical insights on 0-day exploit broking. We extract the prices quoted by two 0-day brokers for a range of exploits over time. We subsequently display longitudinal plots to understand how different exploits varied in price with the mean price of exploits growing by 1240% over the 6-year observation period. The data is not updated sufficiently regularly or frequently to build a time-series model, so, instead, we conduct a regression analysis of the most recent snapshot of prices. The results show that properties of the exploit (e.g. the functionality it achieves) provide the most explanatory power, and that the system targeted by the exploit provides comparably little explanatory power.

Section 2 provides background on 0-day brokers using a mixture of researcher testimonies, research articles and the brokers' websites. Section 3 describes related work in vulnerability markets and making security inferences from market indicators. Section 4 describes how we collected data. Section 5 presents our results. In Section 6 we discuss what might be inferred from 0-day brokers and the business strategy of brokers. Finally, Section 7 summarises the contribution of the paper and offers conclusions.

2 Background

In this section, we explain the market structure of 0-day markets drawing on a range of sources [11, 12, 10, 13, 14, 15]. Exploit brokers function as market makers by contracting with suppliers (security researchers), managing an inventory of exploits, and selling to buyers (actors who deploy offensive cyber operations). In doing so, brokers can more efficiently manage transaction costs relative to suppliers and buyers directly contracting with each other.

Search Suppliers and buyers face a matching problem as exploits are heterogeneous due to differences in targeted product, version, and the exploit's capabilities. Search costs for researchers are high because exploits are transitory [16], which means the expected value of an exploit decreases with time. Further, actively searching in an open market reveals which system the researcher has exploited and which system the offensive actor wants to exploit, creating liability and operational problems for both. Thus, brokers maintaining a stock of exploits efficiently solves the matching problem *and* also limits information leakage. Additionally brokers provide a layer of insulation against reputation and legal fallout. For example, brokers may have legal entities in multiple countries to deal with export regulations [11].

Until recently, brokers did not publicly advertise prices. Instead, market participants needed to navigate informal professional networks—a sign of an immature market. More recently, two brokers began publicly advertising prices paid to researchers (notably not advertising what buyers pay). There is no centralised exchange for exploits.

Negotiation Brokers acting as market makers allow suppliers and buyers to contract differently. The discovery and lifetime of exploits is unpredictable, necessitating one-time contracts. While infrequent payments are less of a problem for individual engineers, offensive actors cannot accept the resulting interruptions. A break in intelligence collection while a new exploit is procured could endanger national security or hamper law enforcement. Brokers maintaining a stock of exploits means buyers can sign continuous contracts, even though the stock is unpredictably replenished. Brokers also provide the function of verifying the exploit.

The negotiation between broker and seller typically proceeds as follows. The seller contacts the broker, whether through connections or directly, with a specification sheet of the exploit. The broker then responds with a non-binding preliminary offer, usually less than the maximum payout, after taking limitations into account. The seller may then submit their exploit for evaluation by the broker. It is customary to allow for an assessment period of up to two weeks. In this time the broker tests the exploit, and compares their result to the previously provided specifications. If the broker is interested in buying the exploit following review, both parties sign a contract including payment terms (warranty), intellectual property rights and exclusivity. The payment is typically

spread out over the course of a few months to a year.

Monitoring Exploits are made useless when a supplier patches the targeted system. This necessitates monitoring the behaviour of both suppliers (who can re-sell to another actor) and buyers (who can use exploits widely and noisily). In theory, brokers could punish such irresponsible behaviour by suspending trading with either suppliers or buyers. In practice, brokers focus on deterring irresponsible suppliers by including contractual terms about re-sale or publication of the exploit, as well as stopping payment if a patch is released that fixes the vulnerability underlying the exploit. The brokers' ability to restrain buyers is less clear, although some report only selling to supposedly reliable governments (Zerodium mention "Western governments" [10]).

Summary

Bug bounties and 0-day brokers represents relatively illiquid and opaque markets. Zerodium paid out just $50 million since being founded in 2015 [17]. Low sales volume means prices are not updated regularly—certainly not continuously. Price discovery is further limited because advertised prices are actually maximum pay-outs, with the real prices depending on discretionary factors such as which versions are affected or the exploit's reliability and/or time to execute. Notably, these institutional factors suggest that our data source is a noisy proxy for the true cost of compromise.

3 Related Work

Our study contributes to the literature on the structure of vulnerability markets, which we review in Section 3.1. But it also moves towards an alternative to systems security metrics created by engineers, which we review in Section 3.2. Section 3.3 outlines related works making inferences from market indicators with relevance to security.

3.1 Vulnerability Market Structure

Selling exploits to a broker is but one of many ways for an independent security researcher to share information. Sales channels have differing levels of legitimacy. Some vendors offer bug bounty programs (BBPs)—a monetary reward for reporting directly to the vendor—which have been shown to be effective [18] and efficient [19] security interventions. Institutional BBPs, in which dedicated staff set policy and evaluate submissions, are run by large technology vendors such as Google and Apple [20]. Such fixed costs can be avoided by subscribing to a platform that operates the BBP. The HackerOne platform leads to less sophisticated vulnerabilities (predominantly web vulnerabilities [21]) discovered via automated tools [22, 23, 24].

Black markets in which criminals offer financial rewards for exploits sit at the illicit end of the spectrum [25, 26, 27]. While it has been suggested that dishonesty amongst thieves would undermine market function [28], there is evidence that underground markets have developed enforcement mechanisms that prevent dishonest practices [27]. Freelancers have declined from 80% to 20% of total participants (as of 2014) as criminal organisations form [25]. We also see that exploits procured in black markets are used by threat actors in-the-wild [29].

Exploit brokers exist somewhere between bug bounties and black markets in terms of legitimacy depending on who the broker sells to. An early theoretical model of vulnerability brokers suggested unregulated brokers would be incentivised to leak the vulnerability information inappropriately [7]. There are few empirical studies of such brokers, which suggests that alternative business models won out. For example, ExploitHub was a trusted intermediary for exploits of published CVEs between 2010 and 2015. The firm's website reports that the industry changed as the business model was no longer viable [30].

Indeed, many vulnerability markets offer seemingly low financial incentives: exploit kits [25, Table 2.2] are priced in thousands of dollars on Dark-Web forums; the mean price across 181 ExploitHub transactions was "a little above [a] hundred dollars" [31]; the average HackerOne payout is in the hundreds of dollars [22, 23, 24]. In contrast, 0-day exploits can be priced in the millions of dollars [32]. This motivates us to turn our attention to 0-day brokers, who largely operate in the shadows [14]. Ablon and Bogart [15] assembled a "sparse and inconsistent" table of prices for 0-days and Meakins [33] provide a snapshot of pricing across four different brokers. Both of these papers display prices as examples, rather than collecting data for a systematic analysis—such an analysis is one contribution of this paper. The second contribution involves using this data to make inferences about systems security.

3.2 Security Metrics

Traditionally, engineers design security metrics in a bottom-up way by focusing on a sub-component of security [34]. For example, there is a rich body of work on how to measure the strength of an individual password [35, 36, 37], but far less knowledge about how the strength of an individual password relates to security outcomes [38]—not least because to develop such knowledge would require reliable metrics of system-level outcomes. The most common approach is to measure security incidents such as network abuse at the ISP-level [39] or cyber incidents at the firm-level [40]. Incident-oriented approaches rely on firms publicly reporting incidents, which is notoriously inconsistent [41], and also infrequent, which undermines statistical power [42]. The research community has largely failed to link security practices to incident-based outcome metrics [43].

One alternative is attacker-centric metrics such as the difficulty of compromising the system of interest [34]. Again, the bottom-up engineering approach is to map out all possible combinations of vulnerabilities in an attack graph, and to then reason about the difficulty of each path [44]—which is challenging

in practice. Exploit markets provide a top-down alternative because the price of a given exploit is equivalent to the cost of compromise—in a monetary unit, advertised publicly, and updated regularly [6]. It is conceivale that exploit market data could be directly inputted into the Return on Attack [45] and other metrics that incorporate attacker costs [46].

3.3 Security Inferences from Market Data

Our aim is to contribute to a body of work using economic indicators to study security. The most common approach involves studying stock market reactions to information security events [47, 48, 49, 50, 51]. Such studies show breaches have a negative impact on stock market value, although the effect depends on the event window, industry and sample of breaches [52, 53]. Announcements that firms will pursue cybersecurity certifications and standards can also lead to a positive effect on stock market value [54, 55]. More exotic indicators include cyber insurance prices, which can be reverse engineered to reveal actuarial expectations regarding cyber risk outcomes [56].

Even before BBPs existed, it had been argued that higher prices are a signal of more secure products [57]. In a similar vein, Allodi [58] argues data from criminal markets for exploit kits can inform risk assessments, noting that technical severity is a poor predictor of how widely a vulnerability will be exploited [26, 59]. Specifically, Allodi shows that higher priced exploits are *less* likely to be exploited at scale [60], which can in theory be fed into risk-management decisions. This seems to confirm the theory that increasing cost of compromise decreases the probability of attack [45].

4 Data Collection

Historic 0-day transactions are difficult to find because publishing such data would reveal details about which exploits exist and who has access. Instead, we used the price lists advertised by brokers, accepting that these are actually upper bounds with the associated limitations (see Section 2). We collected prices from Zerodium and Crowdfense, who are, to the best of our knowledge, the only two companies that publicly advertising specific prices for 0-days. Due to the heterogeneous nature and brevity of these lists, we extracted the prices manually, together with short descriptions of the exploits.

We used the Internet Archive's Wayback Machine to investigate the long-term evolution of these prices [61]. It provided us with frequent snapshots of the programme's websites, all the way back to their respective inceptions. The 'Comparison' feature of the Wayback Machine shows which points in times the websites were altered. We manually inspected, and noted, the date at which the Wayback Machine registered a change in the website. Unless the programme provided the specific date they updated the price list, we used the snapshot dates to estimate the duration for which these prices were valid. The Wayback Machine sampled much more frequently than the website was updated, and so

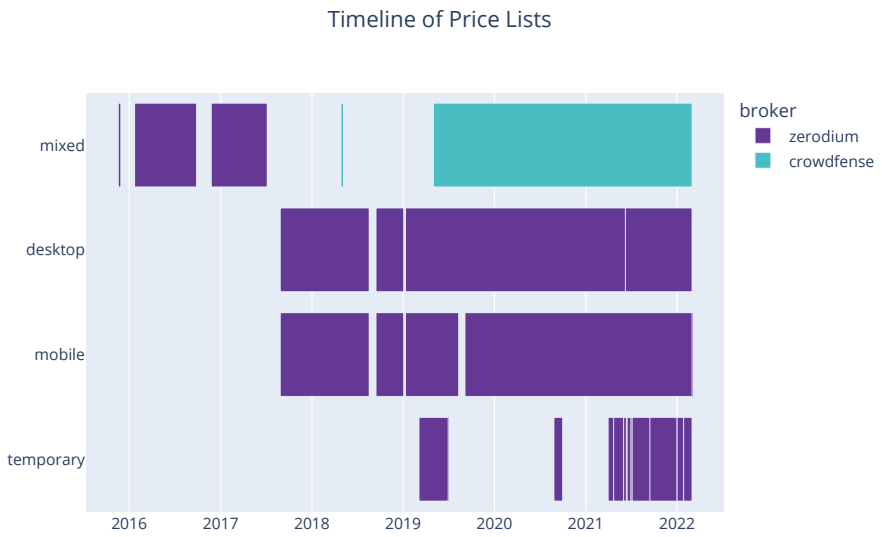

Figure 1: Labels on the y-axis indicate the type exploits on the price list, mobile, desktop or mixed (meaning both desktop and mobile). Temporary is a special category of purchase offers by Zerodium, lasting only in the order of weeks or months. Presumably these exist to respond quickly specific demand, without having to alter the price list.

we are confident that the gaps between snapshots will not significantly affect our results. Figure 1 displays the windows of validity of all price lists we collected.

We then extracted descriptive information about each exploit price directly from the broker, which included:

- Properties of the exploit (e.g. remote code execution, local privilege escalation, persistence, sandbox escape);
- target operating system (e.g. Android, iOS, Windows, Linux, and MacOS); and
- target application (e.g. Facebook Messenger, Signal, Google Chrome, antivirus).

The exploit properties were directly extracted, only applying discretion in classifying full chain with persistence exploits as *remote code execution*, *local privilege escalation*, and *persistence* all set to true. We directly extracted the targeted operating system, applying discretion to merge Linux and OpenBSD into a single category, unix-like. We used discretion in grouping target applications into the following categories:

- OS (when the exploit targets the operating system, not an application)
- Browser (e.g. Google Chrome, Mozilla Firefox)
- Document (e.g. MS Office, Adobe Acrobat, 7-zip)
- Messenger (e.g. Facebook Messenger, Signal, WhatsApp)
- Mail (e.g. Thunderbird, Exim)
- Security (e.g. antivirus, VMware ESXi)
- Web (e.g. MS IIS, Apache, phpBB)

We removed router exploits from our dataset, since they are difficult to compare to software exploits.

We also collected supplemental data about the operating systems for which 0-day exploit prices are advertised. We extracted the number of 0-days for specific operating systems from a spreadsheet maintained by Google Project Zero, which collects data from "a range of public sources". We extracted the number of CVEs per operating system from the National Vulnerability Database (NVD), accepting the minor errors in accuracy [62]. Combining the number of Internet users [63] and OS market share of those users [64], we estimated the number of active Internet users by OS from 2015 to 2021.

5 Results

Section 5.1 displays how prices developed over time pooling both price lists. Section 5.2 presents a regression analysis of the most recent snapshot of prices across both brokers. Section 5.3 compares exploit prices to three security metrics for each of four different operating systems.

5.1 Longitudinal Price Development

Figure 2 displays a scatter plot of the prices against time. To account for temporary prices (see Figure 1), we sample all prices that were valid in the calendar year. The two major developments are the increasing scope of the Zerodium program as exploits of new systems were added and the increase in dispersion of prices over time. The increased dispersion is illustrated by the maximum price quoted by Zerodium rising from $500k in 2015 to $1.5 million in 2017 and then to $2.5 million in 2019. Some intermediate values also increased in value, although many exploits retained the same price throughout the sample window, which means the minimum price did not increase.

This inflation among a subset of exploits can be seen in Figure 3, which shows the mean price for different kinds of exploits. Price increases from 2018 to 2020 were most dramatic among messenger applications. Exploits of web browsers and email applications also increased in price, whereas some exploits (such as anti-virus exploits) flat-lined.

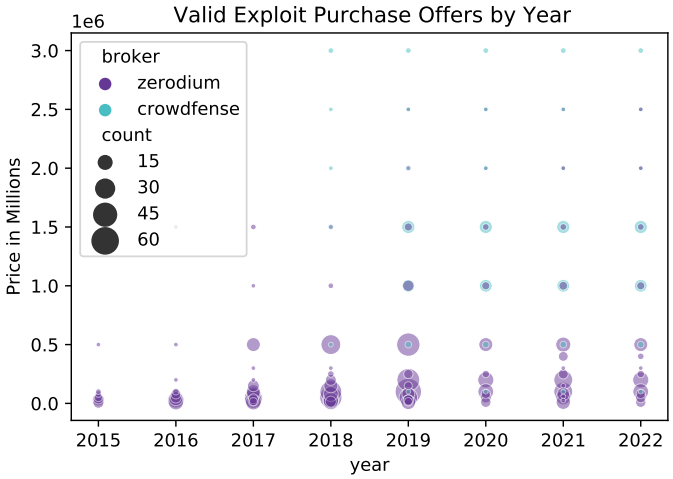

Figure 2: Distribution of all prices in each calendar year. The area of points is proportional to the number of exploits with that price, at that time.

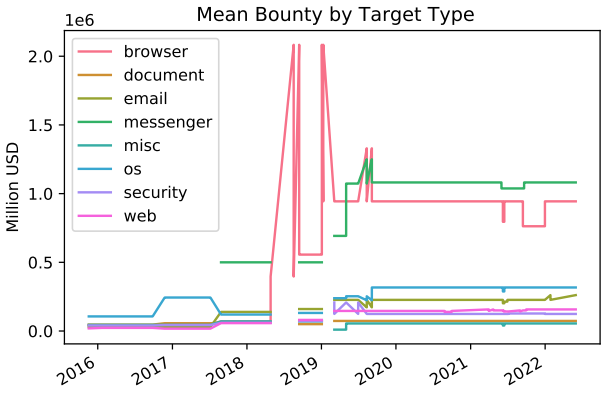

Figure 3: Comparing the average prices of exploits for different applications, from 2015 until present. The spikes an artefact of our data collection method; for certain periods the price lists of containing the cheaper exploits were unavailable on the Wayback Machine.

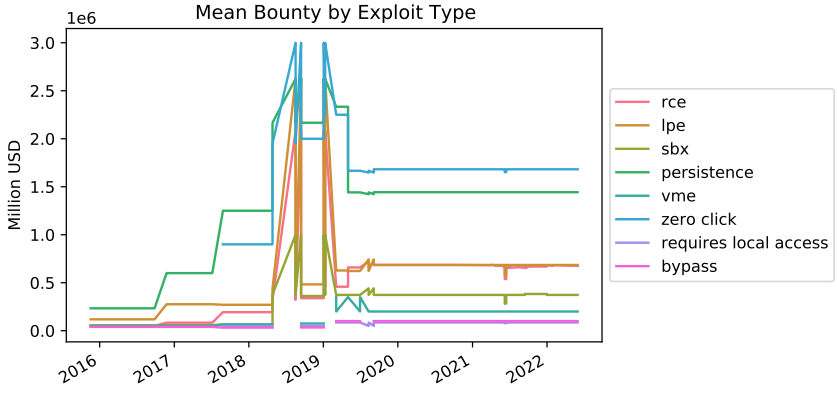

Figure 4: Comparing the average prices of different kinds of exploits, from 2015 until present. The spikes an artefact of our data collection method; for certain periods the price lists of containing the cheaper exploits were unavailable on the wayback machine. The average is reacting accordingly.

Similar patterns can be seen in the mean price of different kinds of exploits over time in Figure 4. The average price of a full chain with persistence exploit grew from $250k to almost $2 million in less than five years. Less dramatic price increases were also observed for exploits that provide local privilege escalation (escaping access control permissions) or remote code execution (running arbitrary code on the device). Exploits that can escape virtual machines and application sandboxes were also expensive. It is also notable that, across both graphs, prices do not go down for any exploit. This is likely because the quoted prices are maximum bids (see Section 2) and any price decreases result in the negotiated price falling even lower beneath the maximum.

Turning to how prices varied across operating systems, Figure 5 shows how the total price across all exploits of a given operating system changed over time. This figure is affected by both the price of each exploit in the category and also the number of exploits in the category. The most rewards were offered for mobile exploits following their introduction in mid 2017. iOS and Android subsequently saw the most increase. In terms of desktop and server exploits, there were more rewards offered for exploits of Windows than Linux. Zerodium stopped advertising for certain macOS exploits in mid 2018. Many of the Linux exploits related to server technologies, rather than the desktop OS.

5.2 0-day Broker Prices

Beyond the time trends, the previous analysis suggests properties of the exploit and targeted system can explain differences in prices. We explore these factors with a log-linear model that tries to explain variation of prices in the most recent

Table 1: Summary Statistics

Variable	N	Mean	Std. Dev.	Min	Pctl. 25	Pctl. 75	Max
broker	140						
... crowdfense	36	25.7%					
... zerodium	104	74.3%					
price	140	581357.143	644352.567	10000	1e+05	1e+06	3e+06
os	140						
... windows	24	17.1%					
... android	41	29.3%					
... ios	44	31.4%					
... macos	5	3.6%					
... unix-like	26	18.6%					
mobile	85	60.7%					
rce	114	81.4%					
lpe	50	35.7%					
sbx	7	5%					
persistence	35	25%					
vme	2	1.4%					
zero.click	22	15.7%					
requires.local.access	7	5%					
bypass	7	5%					
target.type	140						
... os	39	27.9%					
... browser	16	11.4%					
... document	6	4.3%					
... email	15	10.7%					
... messenger	43	30.7%					
... security	7	5%					
... web	14	10%					
temporary	4	2.9%					
billion.os.users.2021	140	1.126	0.726	0.046	0.81	1.997	1.997

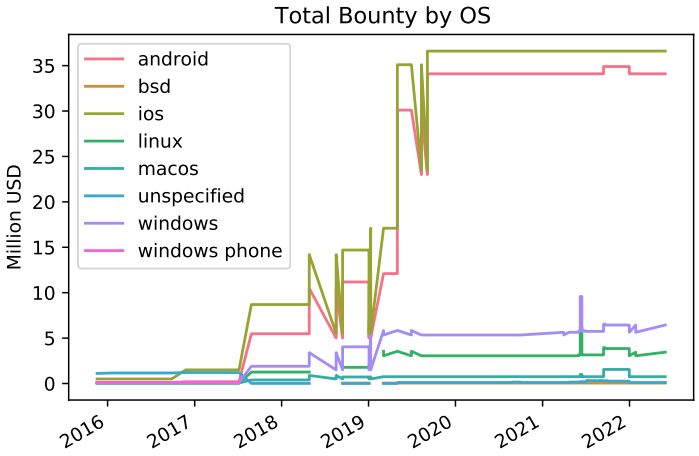

Figure 5: The sum of all exploit bounties by OS, from 2015 until present.

snapshot of exploits. We specify the following regression:

$$log(Y_i) = \beta_0 + \beta_1 x_{i1} + ... + \beta_N x_{i19} + \epsilon_i \quad (1)$$

In our regression, Y is the price of each exploit in dollars. The variables $x_1, \ldots, x_{19}$ are dummy variables related to the properties of the exploit, the targeted system and whether the prices are temporary. We run regressions for a subset of these variables, as well as replacing the operating system variable with: (i) the userbase of the OS; and (ii) a dummy for whether the OS is mobile (e.g. iOS or Android). Table 1 displays the descriptive statistics for each variable.

Table 2 displays the estimated effects of the properties of the exploit, the targeted operating system, and targeted application on the exploit prices. We run models for each of the variables independently. We then combine all variables representing the targeted OS as a categorical variable (Model 4), numeric variable in the number of users (Model 5) and as a binary variable capturing whether it is a mobile or desktop/server (Model 6). All models are statistically significant under an F-test ($p = 0.001$). Across all models we see that *temporaryTrue* is positive and statistically significant ($p < 0.05$).

When comparing the models with just one variable (Models 1–3), we see that the model including only the properties of the exploit explains the most variance ($R^2 = 0.641$) compared to variables for the targeted application ($R^2 = 0.510$) and OS ($R^2 = 0.366$). Adding the other two variables to Model 1 only boosts R^2 by 0.08. We tried replacing *os* with the number of users (Model 5) or whether the exploit targets mobile (Model 6), but including *os* as a full categorical variable leads to the most adjusted R^2 and so we proceed by analysing it.

The variables with the largest effect size are all properties of the exploit, apart from *TemporaryTrue*. Exploits achieving *persistence*, which means the

	log(price)					
	(1)	(2)	(3)	(4)	(5)	(6)
rceTrue	1.267***			0.981***	1.146***	1.007***
	(0.241)			(0.253)	(0.252)	(0.254)
lpeTrue	1.145***			0.523**	0.584***	0.537**
	(0.168)			(0.196)	(0.198)	(0.196)
sbxTrue	1.339***			0.898*	0.838*	0.917*
	(0.361)			(0.378)	(0.377)	(0.371)
persistenceTrue	2.028***			1.082***	1.190***	1.075***
	(0.212)			(0.247)	(0.248)	(0.248)
vmeTrue	1.923***			2.139***	2.258***	2.111***
	(0.645)			(0.648)	(0.662)	(0.653)
zero.clickTrue	0.728***			0.711***	0.691***	0.729***
	(0.238)			(0.212)	(0.216)	(0.213)
requires.local.accessTrue	−0.226			−0.297	−0.358	−0.283
	(0.361)			(0.330)	(0.334)	(0.329)
bypassTrue	1.327***			0.485	0.813*	0.517
	(0.416)			(0.401)	(0.391)	(0.404)
osandroid		1.415***		0.362		
		(0.294)		(0.277)		
osios		1.436***		0.409		
		(0.291)		(0.276)		
osmacos		−0.341		−0.497		
		(0.550)		(0.407)		
osunix-like		−0.630*		−0.505		
		(0.318)		(0.256)		
billion.os.users.2021					0.177	
					(0.103)	
mobile.osTrue						0.637**
						(0.233)
target.typebrowser			1.087***	0.356	0.259	0.387
			(0.289)	(0.264)	(0.266)	(0.265)
target.typedocument			−0.814	−0.643	−0.965*	−0.470
			(0.427)	(0.426)	(0.383)	(0.415)
target.typeemail			−0.122	0.285	−0.100	0.191
			(0.315)	(0.348)	(0.310)	(0.332)
target.typemessenger			1.837***	0.536*	0.574*	0.539*
			(0.215)	(0.226)	(0.231)	(0.227)
target.typesecurity			−0.634	−0.447	−0.814*	−0.385
			(0.400)	(0.426)	(0.389)	(0.419)
target.typeweb			−0.954***	−0.277	−0.779*	−0.467
			(0.306)	(0.387)	(0.327)	(0.354)
temporaryTrue	1.277***	1.225*	1.173*	1.061*	1.124**	1.314***
	(0.436)	(0.586)	(0.534)	(0.422)	(0.419)	(0.406)
Constant	10.283***	11.760***	11.984***	10.767***	10.621***	10.480***
	(0.259)	(0.238)	(0.156)	(0.317)	(0.288)	(0.288)
Observations	140	140	140	140	140	140
Adjusted R^2	0.641	0.366	0.510	0.723	0.709	0.719

*p<0.05; **p<0.01; ***p<0.005

Table 2: Log-linear regressions with price as the dependent variable and a Windows OS exploit as the reference.

adversary retains control after restarting the device or logging off, are 194% more expensive. Although the effect size for exploits that require no user interaction (*zero.clickTrue*) is comparably small, in all of our observations this variable is only true if *persistence* is also true. In particular, the statistically significant effect size on *zero.clickTrue* suggests that user interaction is not cheap. In a world where getting users to click on arbitrary links is costless, this variable would presumably not be significant.

We also see that virtual machine escapes have the biggest effect size. The ordering of the effect sizes for local privilege escalation (*lpeTrue*), sandbox escape (*sbxTrue*), and virtual machine escape (*vmeTrue*) makes sense on a technical level—LPE only requires overcoming the device's access control system, SBX requires escaping an application designed specifically to contain code run within it, and VME additionally requires escaping a virtual operating system. The coefficient for exploits that require local access is negative, which would make sense given the security model differs when one has local access (e.g. side channel attacks become possible), however this effect is not statistically significant in any of the models. We also see that exploits that bypass security mechanisms (*bypassTrue*) are more expensive, but again this effect is not statistically significant in the full model.

In the full model, the coefficients for the targeted operating system were not statistically significant when represented as either categorical data (Model 4) or in terms of user base (Model 5). However, the coefficient for whether the exploit targets a mobile device (Model 6) is statistically significant ($p < 0.01$). To interpret this, consider that brokers offer prices for an exploit of an application like Facebook Messenger on both iOS and Android. This means that the OS that the applications runs on does not seem to affect price, although exploits of mobile apps are more expensive. This could be because of the increased control mobile platforms have via app stores or alternatively due to architectural design using application permissions [65, p. 210]. However, we should not read too much into this given the individual coefficients for Android/iOS are not significant in Model 4.

A similar story is true for the application that is targeted. Noting that the reference exploit targets the Windows OS, the coefficients are not statistically significant for any variable apart from *target.typemessenger*. This suggests that exploits targeting messenger applications are more expensive than the other categories. Nevertheless, it is worth noting that this variable required the most researcher interpretation in building it.

5.3 Exploits, Vulnerabilities and Broker Prices

It is natural to ask what drives exploit prices over time or indeed whether exploit prices drive or predict any other security relevant metrics. As we have neither the prior theory to propose a model of this relationship nor the data availability to fit it with sufficient granularity, we instead display four security metrics per year for four popular operating systems. We display: (i) the number of CVEs affecting that OS per year; (ii) the number of in-the-wild 0-days affecting

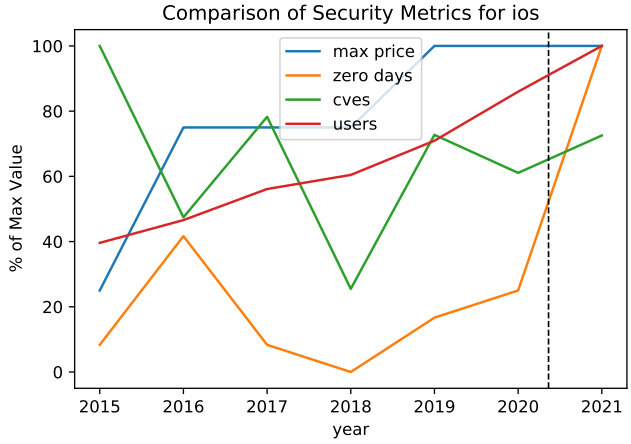

Figure 6: Normalised annual count for the number of CVEs, 0-days and users, along with the maximum price of an exploit of iOS. The dashed vertical line indicates May 13th 2020, when zerodium temporarily suspended purchases of iOS exploits.

that OS per year; (iii) the number of users of that OS per year; and (iv) the maximum price across all exploits of that OS in a year. All are normalised by their maximum value.

For Apple iOS, Figure 6 shows that the maximum price increased by over 350% and user share increased by around 150%. The number of CVEs displayed no clear trend with a lot of year-on-year variation. Meanwhile, observed iOS 0-days spiked in 2016 and then fell back down until another spike in 2021. Notably, in May 2020 Zerodium announced that they were no longer buying iOS exploits because of a surge of submissions. This was followed by a 500% year on year increase in observed 0-days, which suggests brokers suspending trading could be a signal to look out for in the future.

Much like iOS, Figure 7 shows Android saw both an increasing user base and an increasing maximum exploit price, as well as fluctuation in yearly CVE numbers. Android also saw a 2016 spike, as well as a 200% increase in the number of 0-days in 2021. This casts doubt over whether Zerodium suspending trading of iOS exploits was relevant, given a similar pattern can also be observed in Android.

Turning to desktop OS, Figure 8 shows that the Windows OS had little user base growth, a CVE count that seemed to grow consistently from 2015–2019 and then fell away, and a fluctuating but relatively stable 0-day count. The maximum price increased comparably gradually from 2016–2019. Figure 9 shows that the Apple desktop OS shows the least growth in price of exploits, a declining CVE count and a steadily increasing user base. Again, the number of 0-day exploits is highest in 2015–2016 and then falls until a dramatic spike in

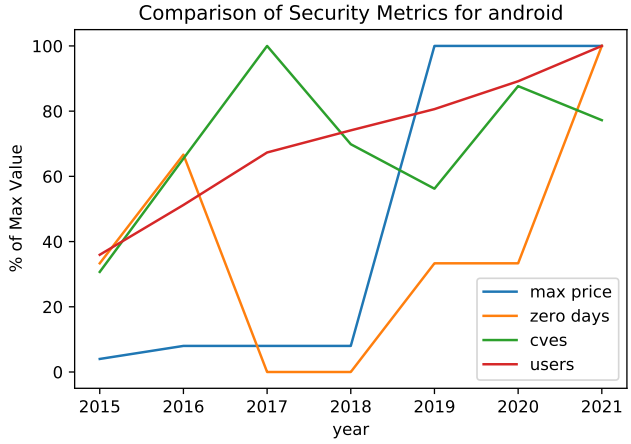

Figure 7: Normalised annual count for the number of CVEs, 0-days and users, along with the maximum price of an exploit of the Android mobile OS.

2021.

Taking these trends together, we see that 0-day exploit prices increased to a maximum in 2019 across all four operating systems, and that 0-day exploits observed in the wild spiked in 2021. User share increased steadily from 2015–2020 for all operating systems but windows. CVEs display a lot of year-to-year variation and do not show a clear trend across platforms.

6 Discussion

Returning to the related work of Section 3, we first discuss the potential to infer security properties from the quoted prices in Section 6.1. We then turn to exploit brokers as an institution, which we discuss in Section 6.2.

6.1 Exploit Prices for Inference

So how much information could we extract from the observed exploit prices? Table 2 explains how brokers price different exploits, providing evidence for statements such as "exploits that require no user interaction are more expensive" or "exploits of messenger apps are more expensive". There is, perhaps, an argument for such information being fed back into risk-based decisions. The following reasoning is supported by our results, "opting to communicate sensitive information over a messenger app seems to raise the cost of compromise of an adversary procuring exploits on the market relative to communicating the same information over email", but the same statement could be supported by reasoning about design of each mode of communication.

We can also backwards rationalise some of the effect sizes of Figure 2. That

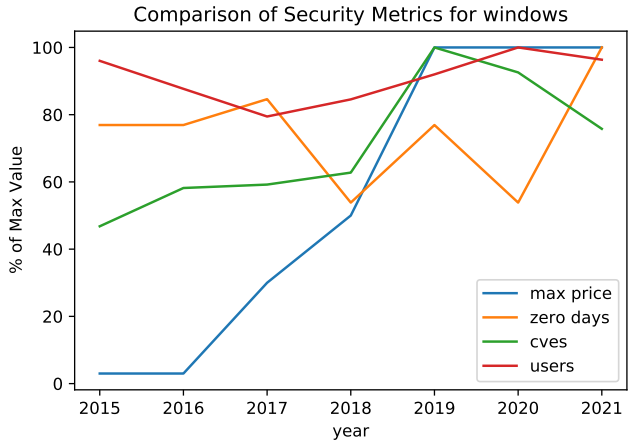

Figure 8: Normalised annual count for the number of CVEs, 0-days and users, along with the maximum price of an exploit of the Windows desktop OS.

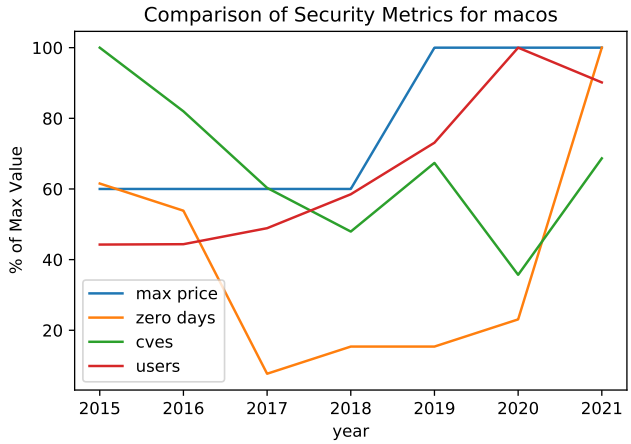

Figure 9: Normalised annual count for the number of CVEs, 0-days and users, along with the maximum price of an exploit of the mac desktop OS.

temporary prices are higher could be explained by the fixed cost of researchers switching to researching a new system. Similarly, it makes intuitive sense that escaping a virtual machine is more difficult than escaping a sandbox and so the price of exploits doing so are higher (see $sbxTrue < vmeTrue$ in all models). Other results are hard to explain, such as the non-significant variable related to whether the exploit requires local access—one would expect that such exploits should be cheaper based on similar logic to the explanation that zero-click exploits are more expensive because user interaction is costly to manipulate. Such confounding results should cast doubt on backwards rationalising these effects and concluding the market prices capture some real signal.

The above reasoning assumes the cost of an exploit is driven by systems architecture making exploits more difficult to find. This follows the logic imagined by early writing on bug bounties in which higher prices reflect the difficulty of finding such exploits [2]. But whereas bug bounties are one-sided markets, brokers are two-sided in which prices are also influenced by buyers. If supply is limited, then prices could instead be driven by which systems adversaries hope to exploit. This is most clearly the case for temporary prices, which can only be explained by a buyer requesting a specific exploit.

Finally, Section 5.3 displayed four security metrics (using admittedly shaky data). However, it is not even clear what theoretical framework could predict the relationship between such metrics: do exploit prices rise when 0-days are discovered and patched, or do rising exploit prices signal increased cyber operations and thereby predict future 0-days? If the latter held, it would make exploit broker data a useful forward looking indicator of risk. Interpreting prices is notoriously difficult [66]. Going forward, it is unclear whether we require better theories or better data, or if it is simply the case that the relationship does not exist.

6.2 Exploit Brokers as Institutions

In Section 2, we argued that 0-day brokers function to reduce transaction costs. Our data does not allow us to analyse how brokers facilitate negotiation or monitoring, and so we reflect on what we learned about search.

That the broker's search is available over the Web is perhaps most surprising, especially with temporary prices about niche products that seem to reveal operational details. For example, from June 15th 2021 to August 31st 2021, Zerodium offered up to $25k for a remote code execution exploit of the Moodle learning management software. An even more temporary price was up to $60,000 for exploits of the IceWarp web server from June 15 2021 to June 30th 2021. The developers and users of such software could use this information to revise perceived threat level.

Given Zerodium has sold at least $50 million worth of exploits, we should ask why buyers accept the associated information leakage. It could be that broad advertisements increase the pool of potential researchers, likely driving down price: Zerodium estimate there are 1500 active researchers [17], whereas Schwartz [11] estimates just 400. Greater competition between researchers should lead to

lower prices, all else being equal. Reasoning from first principles suggests public brokers represent a cheap but conspicuous option. Brokers selling to a limited number of vetted participants offer more secrecy at the cost of higher prices, and internal development provides yet more secrecy.

Speculating on how public brokers reduce information leakage provides an interesting line of thought. Maintaining permanent price tables reduces leakage—one cannot tell whether there is a temporal spike in usage, as we could in IceWarp web server via the temporary price—but does so at a cost. A broker would have to purchase qualifying exploits and maintain an inventory until a buyer emerges, which is risky if that exploit is infrequently traded given exploits are transitory [16]. Brokers can only profitably maintain permanent prices for exploits that are sufficiently liquid, otherwise they are forced to publish temporary prices for niche exploits, and accept the associated information leakage. This suggests that the concentration in software markets may even facilitate the secrecy of offensive cyber operations—adversaries can procure exploits without leaking information providing the market is sufficiently liquid, which further suggests a force towards concentration in exploit brokers. Again, all of this suggests avenues for further theoretical work.

7 Conclusion

0-day brokers represent one of many channels by which vulnerabilities and exploits are bought and sold. Some brokers quote exploit prices publicly, which provided an opportunity for our study. The longitudinal data shows that the mean exploit price quoted by Zerodium increased by 1240% over 6 years, however this inflation was not uniformly distributed across all exploits. The mean price of exploits achieving total control of mobile devices (*full chain with persistence*) grew by over 1071%, whereas exploits that require local access to the device remained relatively steady. Similarly, the mean price of messenger and web browser applications displayed the most inflation, meanwhile Antivirus and file compression exploits changed not so much.

We also ran a log-linear model on the most recent snapshot of prices. The results show that properties of the exploit (e.g. the functionality it achieves) provide the most explanatory power, and that the system targeted by the exploit provides comparably little explanatory power. In fact, the variable describing whether the exploit targets messenger apps was the only significant variable related to which system was targeted. Temporary prices were higher than permanent prices, possibly to account for the bug researcher's switching costs.

That we could even study such prices is most surprising. Adopting this business model allowed one broker to pay out over $50 million to researchers. Information leakage is clearest when it comes to temporary prices, such as a $60,000 price for a niche email application that was only available for two weeks in June 2021. In this specific example it is clear to see how defenders can use this information to adjust risk-management postures, such as hardening systems deploying that web server in the months following this announcement. More

generally, we hope the idea that exploit broker prices reveal information about systems security can help strengthen defensive postures, while recognising that there is a reliance on future work to build the surrounding theory.

Acknowledgements

We would like to thank Rainer Böhme, Max Smeets, Simon Röck, and the two anonymous reviewers for their insightful comments and useful feedback. This research was funded by the Air Force Office of Scientific Research. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 894799.

References

[1] L Jean Camp and Catherine Wolfram. Pricing security. In *Economics of information security*, pages 17–34. Springer, 2004.

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

[3] Dmitri Nizovtsev and Marie Thursby. Economic analysis of incentives to disclose software vulnerabilities. In *Workshop on the Economics of Information Security*, 2005.

[4] Charlie Miller. The legitimate vulnerability market: Inside the secretive world of 0-day exploit sales. In *Workshop on the Economics of Information Security*, 2007.

[5] Mingyi Zhao, Aron Laszka, and Jens Grossklags. Devising effective policies for bug-bounty platforms and security vulnerability discovery. *Journal of Information Policy*, 7(1):372–418, 2017.

[6] Rainer Böhme. A comparison of market approaches to software vulnerability disclosure. In *International Conference on Emerging Trends in Information and Communication Security*, pages 298–311. Springer, 2006.

[7] Karthik Kannan and Rahul Telang. Market for software vulnerabilities? think again. *Management Science*, 51(5):726–740, 2005.

[8] Maochao Xu and Yiying Zhang. Data breach cat bonds: Modeling and pricing. *North American Actuarial Journal*, 25(4):543–561, 2021.

[9] Reed Albergotti. Apple pays hackers six figures to find bugs in its software. then it sits on their findings. *Washington Post*, 2021.

[10] Zerodium. Frequently Asked Questions. `www.zerodium.com/faq.html`, 2021.

[11] Maor Shwartz. Selling 0-days to governments and offensive security companies. Blackhat, 2019.

[12] Vlad Tsyrklevich. Hacking team: a zero-day market case study. tsyrklevich.net, 2015.

[13] Crowdfense. Bug Bounty program. `www.crowdfense.com/bug-bounty-program.html`, 2021.

[14] Serge Egelman, Cormac Herley, and Paul C. van Oorschot. Markets for zero-day exploits: Ethics and implications. In *Proceedings of the 2013 New Security Paradigms Workshop*, NSPW '13, page 41–46, New York, NY, USA, 2013. Association for Computing Machinery.

[15] Lillian Ablon and Andy Bogart. Zero days, thousands of nights. *RAND Corporation, Santa Monica, CA*, 2017.

[16] Max Smeets. A matter of time: On the transitory nature of cyberweapons. *Journal of Strategic Studies*, 41(1-2):6–32, 2018.

[17] Zerodium. `www.zerodium.com`, 2022.

[18] Sam Ransbotham, Sabyaschi Mitra, and Jon Ramsey. Are markets for vulnerabilities effective? *MIS Quarterly*, pages 43–64, 2012.

[19] Matthew Finifter, Devdatta Akhawe, and David Wagner. An empirical study of vulnerability rewards programs. In *USENIX Security Symposium*, pages 273–288, 2013.

[20] Suresh S Malladi and Hemang C Subramanian. Bug bounty programs for cybersecurity: Practices, issues, and recommendations. *IEEE Software*, 37(1):31–39, 2019.

[21] Mingyi Zhao, Jens Grossklags, and Peng Liu. An empirical study of web vulnerability discovery ecosystems. In *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pages 1105–1117. ACM, 2015.

[22] Donatello Luna, Luca Allodi, and Marco Cremonini. Productivity and patterns of activity in bug bounty programs: Analysis of hackerone and google vulnerability research. In *Proceedings of the 14th International Conference on Availability, Reliability and Security*, page 67. ACM, 2019.

[23] Thomas Walshe and Andrew Simpson. An empirical study of bug bounty programs. In *2020 IEEE 2nd International Workshop on Intelligent Bug Fixing (IBF)*, pages 35–44. IEEE, 2020.

[24] Kiran Sridhar and Ming Ng. Hacking for good: Leveraging hackerone data to develop an economic model of bug bounties. *Journal of Cybersecurity*, 7(1):tyab007, 2021.

[25] Lillian Ablon, Martin C Libicki, and Andrea A Golay. *Markets for cybercrime tools and stolen data: Hackers' bazaar*. RAND Corporation, 2014.

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

[27] Luca Allodi, Marco Corradin, and Fabio Massacci. Then and now: On the maturity of the cybercrime markets the lesson that black-hat marketeers learned. *IEEE Transactions on Emerging Topics in Computing*, 4(1):35–46, 2015.

[28] Cormac Herley and Dinei Florêncio. Nobody sells gold for the price of silver: Dishonesty, uncertainty and the underground economy. In *Economics of information security and privacy*, pages 33–53. Springer, 2010.

[29] Chris Grier, Lucas Ballard, Juan Caballero, Neha Chachra, Christian J Dietrich, Kirill Levchenko, Panayiotis Mavrommatis, Damon McCoy, Antonio Nappa, Andreas Pitsillidis, et al. Manufacturing compromise: the emergence of exploit-as-a-service. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 821–832. ACM, 2012.

[30] ExploitHub. `www.exploithub.com/`, 2022.

[31] Jukka Ruohonen, Sami Hyrynsalmi, and Ville Leppänen. Trading exploits online: A preliminary case study. In *2016 IEEE Tenth International Conference on Research Challenges in Information Science (RCIS)*, pages 1–12. IEEE, 2016.

[32] Maylin Fidler. Anarchy or regulation: Controlling the global trade in zero-day vulnerabilities. *PhD diss., Freeman Spogli Institute for International Studies, Stanford University*, 2014.

[33] Joss Meakins. A zero-sum game: the zero-day market in 2018. *Journal of Cyber Policy*, 4(1):60–71, 2019.

[34] Marcus Pendleton, Richard Garcia-Lebron, Jin-Hee Cho, and Shouhuai Xu. A survey on systems security metrics. *ACM Computing Surveys (CSUR)*, 49(4):1–35, 2016.

[35] Joseph Bonneau. Statistical metrics for individual password strength. In *International Workshop on Security Protocols*, pages 76–86. Springer, 2012.

[36] Michelle L Mazurek, Saranga Komanduri, Timothy Vidas, Lujo Bauer, Nicolas Christin, Lorrie Faith Cranor, Patrick Gage Kelley, Richard Shay, and Blase Ur. Measuring password guessability for an entire university. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 173–186, 2013.

[37] William Melicher, Blase Ur, Sean M Segreti, Saranga Komanduri, Lujo Bauer, Nicolas Christin, and Lorrie Faith Cranor. Fast, lean, and accurate: Modeling password guessability using neural networks. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 175–191, 2016.

[38] Cormac Herley and Paul C Van Oorschot. Sok: Science, security and the elusive goal of security as a scientific pursuit. In *2017 IEEE symposium on security and privacy (SP)*, pages 99–120. IEEE, 2017.

[39] Samaneh Tajalizadehkhoob, Tom Van Goethem, Maciej Korczyński, Arman Noroozian, Rainer Böhme, Tyler Moore, Wouter Joosen, and Michel van Eeten. Herding vulnerable cats: a statistical approach to disentangle joint responsibility for web security in shared hosting. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 553–567, 2017.

[40] Sasha Romanosky. Examining the costs and causes of cyber incidents. *Journal of Cybersecurity*, 2(2):121–135, 2016.

[41] Stefan Laube and Rainer Böhme. Strategic aspects of cyber risk information sharing. *ACM Computing Surveys (CSUR)*, 50(5):1–36, 2017.

[42] Frank Nagle, Sam Ransbotham, and George Westerman. The effects of security management on security events. In *Workshop on the Economics of Information Security*, 2017.

[43] Daniel W. Woods and Rainer Böhme. SoK: Quantifying cyber risk. In *IEEE Symposium on Security and Privacy*, pages 909–926, Oakland, CA, May 2021.

[44] Kyle Ingols, Richard Lippmann, and Keith Piwowarski. Practical attack graph generation for network defense. In *22nd Annual Computer Security Applications Conference (ACSAC'06)*, pages 121–130. IEEE, 2006.

[45] Marco Cremonini and Patrizia Martini. Evaluating information security investments from attackers perspective: the return-on-attack (ROA). In *Workshop on the Economics of Information Security*, 2005.

[46] Daniel Schatz and Rabih Bashroush. Economic valuation for information security investment: a systematic literature review. *Information Systems Frontiers*, 19(5):1205–1228, 2017.

[47] Huseyin Cavusoglu, Birendra Mishra, and Srinivasan Raghunathan. The effect of internet security breach announcements on market value: Capital market reactions for breached firms and internet security developers. *International Journal of Electronic Commerce*, 9(1):70–104, 2004.

[48] Karthik Kannan, Jackie Rees, and Sanjay Sridhar. Market reactions to information security breach announcements: An empirical analysis. *International Journal of Electronic Commerce*, 12(1):69–91, 2007.

[49] Alessandro Acquisti, Allan Friedman, and Rahul Telang. Is there a cost to privacy breaches? an event study. *ICIS 2006 Proceedings*, page 94, 2006.

[50] Kevin M Gatzlaff and Kathleen A McCullough. The effect of data breaches on shareholder wealth. *Risk Management and Insurance Review*, 13(1):61–83, 2010.

[51] Lawrence A Gordon, Martin P Loeb, and Lei Zhou. The impact of information security breaches: Has there been a downward shift in costs? *Journal of Computer Security*, 19(1):33–56, 2011.

[52] Georgios Spanos and Lefteris Angelis. The impact of information security events to the stock market: A systematic literature review. *Computers & Security*, 58:216–229, 2016.

[53] Syed Emad Azhar Ali, Fong-Woon Lai, PDD Dominic, Nicholas James Brown, Paul Benjamin Benjamin Lowry, and Rao Faizan Ali. Stock market reactions to favorable and unfavorable information security events: A systematic literature review. *Computers & Security*, 110:102451, 2021.

[54] Jason K Deane, David M Goldberg, Terry R Rakes, et al. The effect of information security certification announcements on the market value of the firm. *Information Technology and Management*, 20(3):107–121, 2019.

[55] Dennis D Malliouris and Andrew C Simpson. The stock market impact of information security investments: The case of security standards. In *Workshop on the Economics of Information Security (WEIS)*, 2019.

[56] Daniel W Woods, Tyler Moore, and Andrew C Simpson. The county fair cyber loss distribution: Drawing inferences from insurance prices. *Digital Threats: Research and Practice*, 2(2):1–21, 2021.

[57] Stuart Schechter. How to buy better testing using competition to get the most security and robustness for your dollar. In *International Conference on Infrastructure Security*, pages 73–87. Springer, 2002.

[58] Luca Allodi. Underground economics for vulnerability risk. *; login:*, 43(1), 2018.

[59] Leyla Bilge and Tudor Dumitraş. Before we knew it: an empirical study of zero-day attacks in the real world. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 833–844, 2012.

[60] Luca Allodi. Economic factors of vulnerability trade and exploitation. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 1483–1499, 2017.

[61] The Internet Archive. The Wayback Machine. `web.archive.org`, 2021.

[62] Afsah Anwar, Ahmed Abusnaina, Songqing Chen, Frank Li, and David Mohaisen. Cleaning the nvd: Comprehensive quality assessment, improvements, and analyses. *IEEE Transactions on Dependable and Secure Computing*, 2021.

[63] International Telecommunication Union (ITU). Key ict indicators for developed and developing countries, the world and special regions (totals and penetration rates). `https://www.itu.int/en/ITU-D/Statistics/Pages/stat/default.aspx`, October 2021.

[64] Statcounter Global Stats. Operating system market share worldwide. `https://gs.statcounter.com/os-market-share`, 2022.

[65] Ross Anderson. *Security engineering: a guide to building dependable distributed systems*. John Wiley & Sons, 2020.

[66] Scott Sumner. Never reason from a price change. `www.themoneyillusion.com/never-reason-from-a-price-change/`, 2010.