The Economics of Cybersecurity — Lecture 13 Notes

Adam Hastings

April 16, 2024

1 Introduction: Types of Exploits

Let's clarify some terminology:

- **Bug** — unintended behavior. Could be a security issue, often times is not. Example:
- **Vulnerability** — A specific instance of a bug that has the potential to be exploited. Makes the device running the code vulnerable. Example: Memory safety violations, integer overflow, executing untrusted unsanitized input (e.g. SQL injection). *What is the name of the database that tracks vulnerabilities?* Technically the NVD (National Vulnerability Database — run by NIST!) is the *database*, but the entries are generally known by their *CVE* number (e.g. CVE-2021-44228).
- **Exploit** — Something that takes advantage of a vulnerability to achieve some goal. Important to note though that oftentimes there is a large gap between a vulnerability and an exploit! *Why?* A vulnerability can be thought of as a "foot in the door", but a vulnerability alone is rarely enough to do anything useful.

2 Characterising 0-Day Exploit Brokers

Ask: What's a 0-day? Why are they important in security and in security economics?

This paper is an empirical study of the marketplace for 0-day exploits. Specifically, it tracks prices from public 0-day brokers, which facilitate exchanges beteween 0-day buyers and 0-day sellers. *This is fairly similar to the Hack-for-Hire paper we read, but what are some of the key differences?* (Empirical, not experimental; 0-day market, not hacking-as-a-service market.)

- *Who are the buyers?* Zerodium claims to sell to "Western governments". Don't know how to verify this claim. Generally as part of the terms of agreement, the buyer is supposed to get exclusive rights to the exploit. *How do you ensure no double-selling?* I don't know. Maybe a research project.

- *Who are the sellers?* A bit unclear; My understanding is that there are private hackers out there who are good at finding and exploiting vulnerabilities and are willing to sell to the highest bidder.
- *What is the role of the broker? Are they just a middleman?* They serve a key role actually! Stockpiling and modulating the rate of 0day releases. Useful for offensive actors to have a steady supply of resources, because they usually have a limited shelf life. Who does this help economically? The sellers! The brokers create a market and can keep prices high. Don't need to worry about gluts or timing the market. Also, the buyer wants a steady supply too. So yes a middleman but one that actually improves outcomes for both buyers and sellers. Not just rent seeking!!)

2.1 Methods

Figure 1: Data was collected from snapshots on the Wayback Machine.

Threats to the validity of this data?

- Shows max prices only

2.2 Results

Figure 2: Prices are in the millions and have been for several years.

Figure 3: Spikes are an artifact of data collection. *Which types of applications have the highest-priced 0-days?* Browser and messengers apps. *Why are browser and messenger exploits worth more?*

- Maybe because more people use them?
- more universal?
- Others?

They actually quantitatively address this later in the paper.

Figure 4: *Which types of exploits are most valuable?* (zero click, persistence). What does this mean? Let's define some of these acronyms:

- **rce** — *Remote Code Execution.* Attacker has the ability to execute code on the victim's device.
- **lpe** — *Local Privilege Escalation.*
- **sbx** — *Sandbox Escape or Bypass. Can someone tell me what a sandbox is here in this case?* E.g. web browser, containers.

- **persistence** — Some types of malware may only be active for a few seconds or less; others may become malicious processes that inject themselve into memory and may try to hide themselves. *If a process exists only in memory and the machine is rebooted, what happens?* The malware is gone. If the attacker was trying to spy on the user, or just keep the malware alive with the intention of doing something malicious later, then simply rebooting will remove this malicious process from the device.
- **vme** — *Virtual Machine Escape.* A form of sbx?
- **zero click** — Exactly what it sounds like. Victim requires no interaction or "mistake".
- **requires local access** — Exactly what it sounds like. (Note the price is lower)
- **bypass** — bypasses specific security mechanisms (ASLR? NX-bit? Not sure which ones; unspecified)

Is it surprising or expected that zero-click and persistence are the most valuable? Why or why not?

How does this data compare to what is currently on Zerodium's home page?

What other trends do we notice? Growth (although they show max prices only? So of course these will be monotonic.)

Figure 5: *Which types of OS are most valuable?* IoS and Android. Both mobile! *Why?* Phone calls are sensitive; things are said that deliberately don't get written in text, perhaps? *Or is it because these platforms are harder to attack?* Probably not. General impression is that iOS is pretty secure, Android is insecure. Yet prices are very similar! I find this interesting.

2.3 Linear regression

I think CS people think of themselves as quantitative people yet are often a bit lacking when it comes to knowing quantitative methods (or maybe I'm just telling on myself). I could easily skip over the next part of this paper but it's so common in economics but also so common in data science and machine learning that I'm going to cover it here too.

Poll: Who knows what linear regression is? This is a really fundamental technique in data analysis you all should know. OLS is "curve fitting" — finding a line that minimzes distance betwen line and datapoints.

When fitting a curve, you want to find the line that most closely follows the data and "punish" the distance between the line and the datapoint. One common method is RSS (residual sum of squares):

$$RSS(b) = \sum_{i=1}^{N} (y_i - x_i^T b)^2$$

where β is the coefficients to the polynomial.

In OLS you are trying to minimize RSS, i.e. find

$$\hat{\beta} = arg\, min_b RSS(b) = (X^T X)^{-1} X^T y$$

which fortunately has a closed for equation.

2.3.1 Results

Source: https://semrasevi342192471.files.wordpress.com/2021/01/guide-to-interpreting-regression-tables.pdf

Let's take a look at the regression table.

- The numbers not in parentheses are the coefficients $\beta_0, \beta_1, ...$ for each chosen independent variable. Note that β_0 is the y-intercept.
- If the coefficient is positive, this means it has as positive relationship on the dependent variable. If negative, then a negative effect on the dependent variable.
- The size of the coefficient represents the magnitude of an effect this variable has.
- The numbers in the parentheses is the standard error, which is the estimate of the standard deviation of the coefficient. (Big standard error = less confidence in this parameter)
- There are also asterisks for "p-values"–what are these? They are statistical significance of the coefficient. This is based off of statistical sampling theory. At $p < 0.05$, there is a 5 in 100 chance that there is no relationship between the independent variable and the dependent variable.
- The lower the p-value, the higher your confidence is that you're observing a true relationship and didn't just get unlucky and are looking at statistical noise. Says nothing about the size of the effect.
- *Where do the p-values come from?* They come from the specific hypothesis test you're running, in this case is was the F-test. Go read about this if you want to learn more about the various ways of doing statistical hypothesis testing.

Based on this regression table, which coefficients are the largest? Which ones have the most impact on the final regression estimate?

The next part talks about R^2 values. *Show of hands—who knows what an R^2 score is?* R^2 is a measure between 0 (no correlation) and 1 (perfect correlation) that tells you the proportion of the

variance that is explained by your regression estimate. It's a measure of the goodness of fit of data. It's defined as:

$$R^2 = 1 - RSS/TSS$$

where TSS is the Total Sum of Squares, defined as

$$TSS = \sum_{i=1}^{N} (y_i - \bar{y})^2$$

where $\bar{y}$ is the mean the observed dependent variables.

Think about this — if your RSS is very close to 0 that means that the line perfectly fits the data, and your R^2 will be close to 1. *What does it take to have an R^2 of 0? What is the intuition behind this?* It means RSS = TSS. *If you have a bad fit, why would RSS be close to TSS?*

2.3.2 Log-linear regression results

They try out a few different models, each which chooses a different set of variables.

- Virtual Machine escapes have the biggest effect sizes
- Properties of the exploit (functionality it achieves) has the most explanatory power
- Targeted system has comparably little explanatory power.
- Local access coefficient is negative — makes sense, but not statistically significant.
- R^2 is good not great. Means there are some missing variables most likely.

2.4 Extended discussion

- *Why advertise prices at all?* Authors pose this question. For attention perhaps? Might be surprising that buyers are OK with information leakeage (e.g. upped prices in niche email application)
- *Can we use prices as a predictor of risk?* If Zerodium is upping the price of exploiting some piece of software that you write, how should you react? What if you use the software in question?

3 Hacking for good: Leveraging HackerOne data to develop an economic model of Bug Bounties

Let's talk about bug bounties. Many companies have bug bounty programs where they will pay researchers money for disclosing bugs (especially security bugs). This is a mechanism for measuring the costs of security (not a very good one though).

4 Background

What is HackerOne? Or Bugcrowd? Crowdsourced bug bounty programs.

How is this different from Zerodium or Crowdfense?

- Recall: Bugs != exploits (although some bug bounties require proof of exploit before paying).
- Biggest difference is the buyers. In crowdsourced bug bounty programs, the product vendor themselves are the ones paying for the disclosure, with the intention of **patching** rather than **exploiting**.
- Another big difference — bug bounties are often web exploits that leak information about the hosting service. The incentives are aligned! It's not about protecting someone else.
- *Is there a difference in the sellers?* My impression from being around the cybersecurity community is that bug bounties are a nice little extra cash for people but no one treats them as a job.

"In a perfectly competitive market, we would expect the prevailing market price to be the marginal cost of producing a good" — why might this not be true for bugs reported in bug bounties? See above—many bugs may be found as a result of other security research endeavors or by accident. Contributes to a non-functioning market (along with incentive alignments) and lowers the price, and disincentivizes those who could be bug bounty hunters from doing so full time.

4.1 Methodology

They also do an OLS regression! See—it really shows up everywhere **(pull up regression formula)**. "Industry" is one-hot encoded. The dependent variable Y_{ij} is the number of valid vulnerabilities submitted to program i in month j.

Problem with first attempt is "endogeneity". *What does "endogenous" mean in everyday language?* It means something that comes from within. Example — some amino acids are endogenous (your body can synthesize them from other ingredients) whereas others are not (i.e. exogenous). Endogenous has the same root word (piecewise doublet to be precise) as "indigenous", which you've probably heard.

So then what might we expect "endogeneity" to mean in statistics? It means that the dependent variable Y is correlated with the error. This can happen when the dependent variable and the regressor variables can causally influence each other. Example: imagine that you are trying to predict income based on education level. We know that higher eduction $\rightarrow$ higher pay. But these variables interact both directions: If you have low pay, you might not be able to afford an education. This is one source of endogeneity and is called "simultaneity". In this paper, *TimeToResolution* suffers from simultaneity: "If a company receives a lot of valid reports [in a given month], it will overwhelm their internal security team and their average resolution time will naturally go up". *Ask: Does this make sense?*

Why is this an issue? It doesn't prevent you from doing OLS. But it can introduce bias to statistical tests done on your OLS. You can fool yourself (or others) if you don't account for this.

To resolve their endogeneity, they did two things: First, they used a method called instrumental variable estimation (IV). To resolve, they use a 2SLS (two stage least squares) regression, which essentially is just doing a regression on some of the variables in question (*NewPrograms* and *AveragetimetoResoution*).

But they suspected lingering endogeneity, so they also run a fixed effects regression via a least square dummy variable estimator (LSDV) which drops all time-invariant variables (industry, revenue, and brand profile).

4.2 Results

First, they do some statistical tests of exogeneity. The tests pass. I do not know much about these tests but it seems like a good thing they passed.

- Finds that "hackers" are price insensitive (elasticity at the median of between 0.1 and 0.2). *What is elasticity? What does this mean? Why do we think this is?* My personal intuition here explains most of this—people submit bug bounties for nice bonus cash but few rely on this as a source of income; naturally you're going to take what you can get, hence the price insensitivity. Often submitting for reputation.
- "Brand profile and revenue have an economically insignificant impact on reports companies receive" — *where in the data is this conclusion supported?* The authors use Twitter followers (and also compare this to web traffic, with they consider to be a worse metric of brand profile (i.e. eBay vs Walmart)).
- What do we see in the R^2 values? Very, very low...meaning that the dependent variable "explains" very little of the variance. This means that there were other unmeasured variables not included in the analysis (omitted variable bias).

4.3 Extended Discussion

5 Post-Paper Reading Discussion

Some questions to chew on:

- *Why do exploits pay so much more than bug bounties?* There's almost no limit to the amount the US will pay to read Putin's emails. Whereas companies have no liability for vulnerabilities (yet—remember National Cybersecurity Strategy?) so paying bug bounties at all is almost just pure altruism (or is it?).

- Let's revisit the "axiom" that "many eyes make all bugs shallow"—*what does this mean?* Often used in reference to open source software. *Is it true?* Think about long tails in fuzzer bug finding. "Many eyes" may catch the ones that are easy to spot but that doesn't mean they don't exist.

5.1 Private BBPs

Large tech companies also often have their own private bug bounty programs (see linked websites on Courseworks). Pay is in the $1000 — $60,000 range. **Compare Zerodium prices with bug bounty prices.**

6 Pwn2Own

- Pwn2Own is an twice-yearly hacking competition with prize money in the low millions.
- Competitors are usually elite CTF teams or soloists.
- Supposed to incentivize ethical + responsible vulnerability disclosure
- Sponsored by the Zero Day Initiative (now owned by Trend Micro, a Japanese cybersecurity firm)
- Vibe is that pwn2own is for show + reputation (+ lower-tier exploits), but if you're really good you're going to sell to an exploit broker who typically will give you more money.
- Incentive? ZDI handles vulnerability disclosure with vendor, makes sure the bug is patched