Blog4 - The Rising Number of CVEs

Michael Ippolito

2022-11-02

Contents

Background

As anyone familiar with the field of cybersecurity will tell you, it's hard to keep up with the hackers. If your organization has data worth exploiting, someone will try to find a way to exploit it. Even if you do all the things you're supposed to do (e.g. patch your systems, scan for vulnerabilities, perform penetration testing, train your users not to click suspicious links), someone will always try to develop a new way to break in. And with the growing number of applications and devices running those applications, the number of ways to exploit them is growing at an increasing rate.

As a cybersecurity professional, I was interested in trying to gauge that rate in quantitative terms. This is helpful for a number of reasons, not least of which is how to evaluate the number of staff hours needed to assess the growing number of vulnerabilities. To that end, I built a model to analyze the trend in the number of vulnerabilities over time, as well as to predict the number of vulnerabilities we might see in the coming years.

Data

The data I used comes from my organization's vulnerability management system, which includes a knowledge base listing all the vulnerabilities it knows about. The list is contributed to by the collaborative cybersecurity community at large and is catalogued by the MITRE corporation with funding from the US federal government's Cybersecurity and Infrastructure Security Agency (CISA). For tracking purposes, vulnerabilities are designated a CVE (common vulnerabilities and exposures) number.

Each CVE entry contains a number of properties that quantify the type and severity of the vulnerability: things like whether there is functional or proof-of-concept code available to exploit the vulnerability, whether the software vendor has patched the vulnerability, whether the vulnerability can be exploited over the network or requires a physical presence on the machine, and whether the vulnerability requires local credentials on the device or if it can be exploited without authenticating.

Based on the above (and other) criteria, CVEs are scored in terms of severity (from 0 to 10); the higher the score, the more serious the vulnerability. These scores are called common vulnerability scoring system (CVSS) scores and can be either version 2 or 3; most recently published vulnerabilities contain both versions, while older vulnerabilities published before v3 was released only have v2 scores.

Scores of 7 to 10 typically indicate flaws that can be remotely exploited without needing credentials and which allow a remote attacker to execute arbitrary code on the device. Vulnerabilities with lower scores are often difficult to execute, require certain narrow criteria to be effective, or require a secondary attack to be successful (e.g. first luring a user to click on a malicious link, which would then direct the user to the attacker's site where the exploit would be triggered).

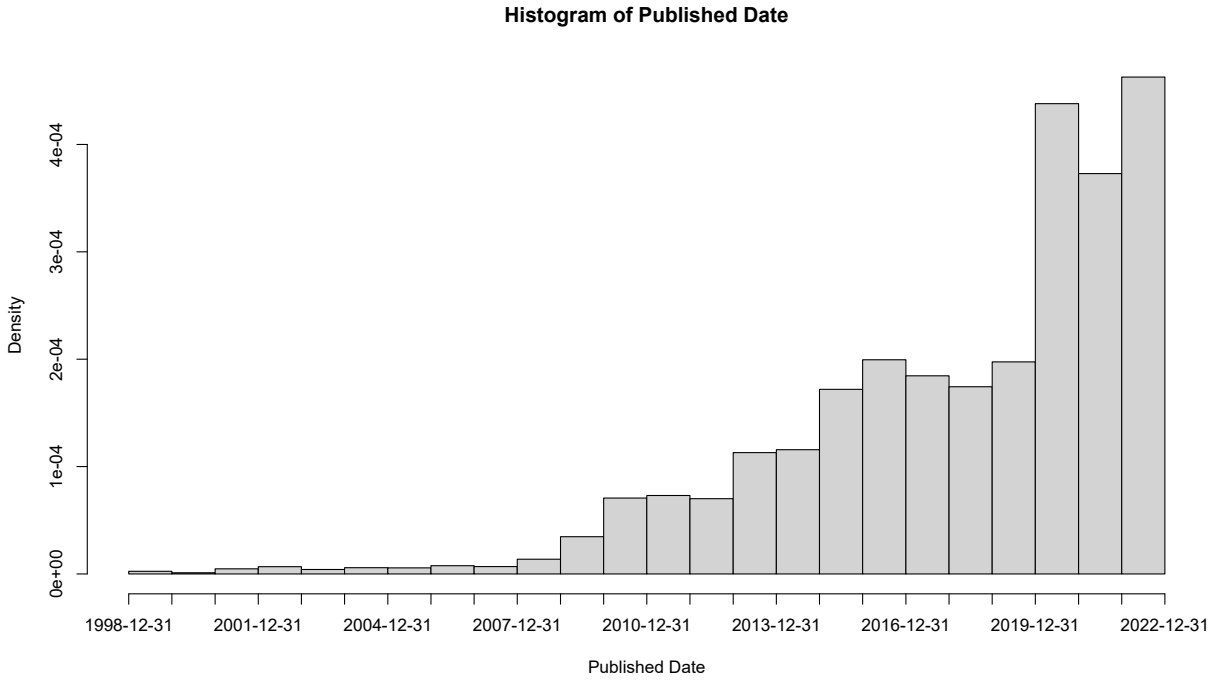
The following is a summary of relevant data catalogued in our vulnerability management system's knowledgebase:

```
##       uid           category          cve            cvss.access.complexity
##  Min.   :    1   Local  : 65937   Length:309623      Low   :192076
##  1st Qu.:21634   SUSE   : 59951   Class :character   Medium:111062
##  Median :43614   OEL    : 19084   Mode  :character   High  :  6485
##  Mean   :43624   RedHat : 17815
##  3rd Qu.:65332   Ubuntu : 17679
##  Max.   :87632   Debian : 17509
##                  (Other):111648
##         cvss.access.vector cvss.authentication   cvss.base
##  Local           : 42426   None    :277189     Min.   : 0.000
##  Adjacent Network: 14161   Single  : 23276     1st Qu.: 6.400
##  Network         :253036   Multiple:  9158     Median : 7.500
##                                                Mean   : 7.411
##                                                3rd Qu.: 9.300
##                                                Max.   :10.000
##
##        cvss.exploitability cvss.impact.availability cvss.impact.confidentiality
##  Unproven        :213334   None    : 32366          None    : 45893
##  Proof-of-Concept: 83603   Partial :136823          Partial :130354
##  Functional      : 10125   Complete:140434          Complete:133376
##  High            :  2561
##  Not Defined     :     0
##
##
##  cvss.impact.integrity   cvss.remediation_level    cvss.report_confidence
##  None    : 41766       Official Fix :305480     Unconfirmed   :  1301
##  Partial :127946       Temporary Fix:    33     Uncorroborated:   923
##  Complete:139911       Workaround   :   805     Confirmed     :307399
##                        Unavailable  :  3305     Not Defined   :     0
##                        Not Defined  :     0
##
##
##  cvss.temporal    cvss.vector_string patchable    severity_level
##  Min.   : 0.000   Length:309623      No :  4098   1:   469
##  1st Qu.: 4.700   Class :character   Yes:305525   2:  8440
##  Median : 5.500   Mode  :character                3:120132
##  Mean   : 5.611                                  4:128347
##  3rd Qu.: 7.300                                  5: 52235
##  Max.   :10.000
##
##    title                            vuln_type         modified
##  Length:309623      Confirmed              :295828   Min.   :1999-01-01
##  Class :character   Confirmed or Potential:  2983   1st Qu.:2016-06-14
##  Mode  :character   Info                   :     1   Median :2020-01-23
##                     Potential              : 10811   Mean   :2018-09-16
##                                                      3rd Qu.:2021-08-30
```

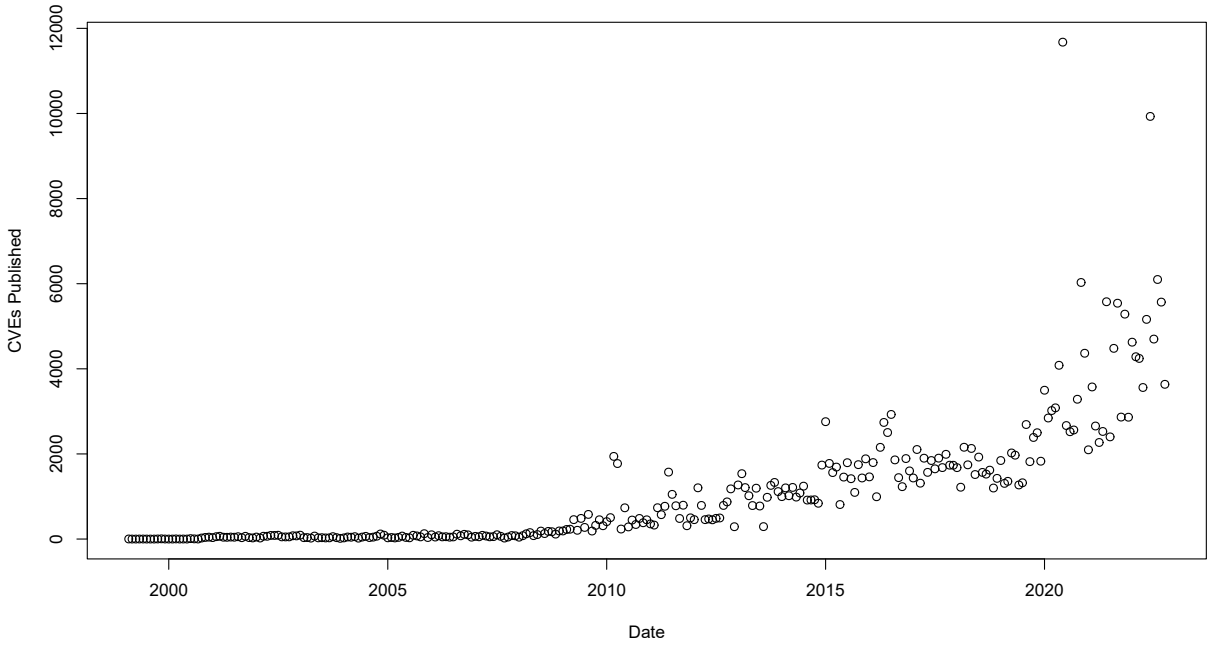
```
##                                                 Max.   :2022-11-01
##
##    published           u_published          u_modified          pub_month
##  Min.   :1999-01-01   Min.   :9.151e+08   Min.   :9.151e+08   Min.   : 1.000
##  1st Qu.:2015-11-18   1st Qu.:1.448e+09   1st Qu.:1.466e+09   1st Qu.: 4.000
##  Median :2019-08-08   Median :1.565e+09   Median :1.580e+09   Median : 6.000
##  Mean   :2018-04-01   Mean   :1.523e+09   Mean   :1.537e+09   Mean   : 6.459
##  3rd Qu.:2021-06-17   3rd Qu.:1.624e+09   3rd Qu.:1.630e+09   3rd Qu.: 9.000
##  Max.   :2022-11-01   Max.   :1.667e+09   Max.   :1.667e+09   Max.   :12.000
##
##     pub_year
##  Min.   :1999
##  1st Qu.:2015
##  Median :2019
##  Mean   :2018
##  3rd Qu.:2021
##  Max.   :2022
##
```

The data required some cleaning and wrangling to get it into this state, e.g. factoring the categorical variables, converting the published and modified dates, and converting CVSS scores to numeric values. Because older vulnerabilities don't have CVSS v3 scores, I opted to discard v3 data and only use CVSS v2 data, which the vast majority of CVEs have (of the approximately 309,000 CVEs, only 14 didn't contain fully populated CVSS v2 information, whereas 76,000 CVEs lacked CVSS v3 data).

While there was a lot of data to work with, I focused on total CVE counts over time. As shown on the following histogram, the number of CVEs is growing at an exponential rate:



Converting the published date of each CVE to a Unix timestamp, I generated a scatter plot of vulnerability counts over time:



Having these data points, I created a model to fit the data points.

Modeling

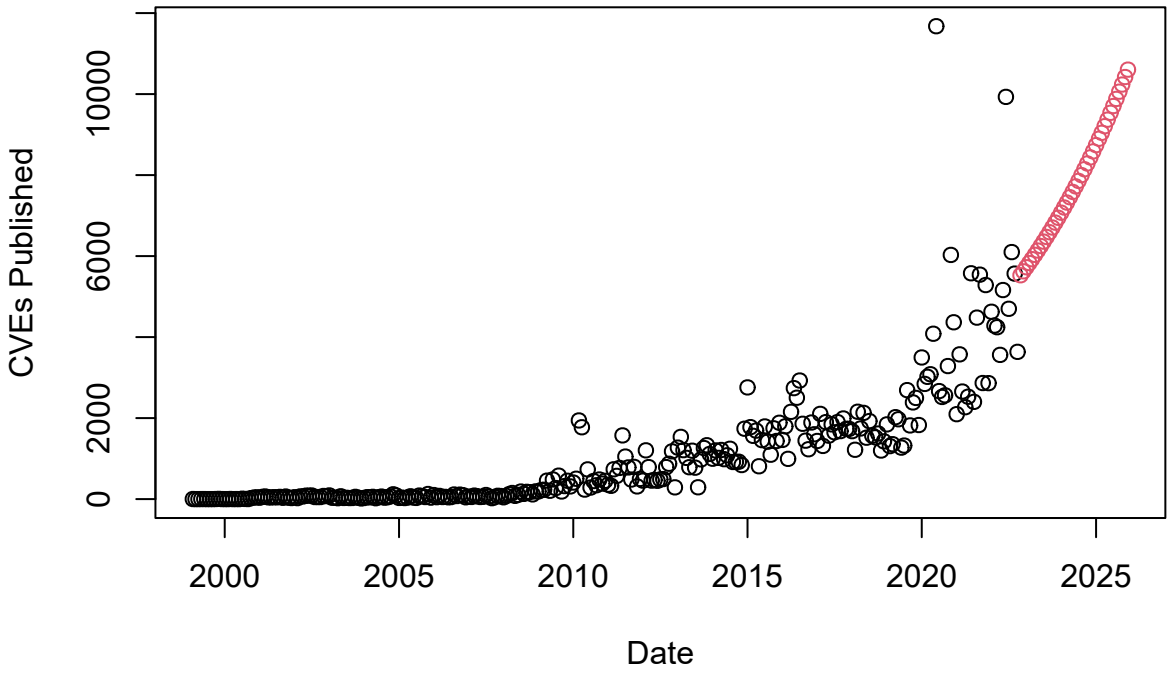
Because this is rate-based data (vulnerabilities per month), I opted to use a generalized linear model using the Poisson family:

```
##
## Call:
## glm(formula = n ~ pub_yrmo, family = poisson(), data = df4)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -30.402   -9.235   -5.044    2.289  112.597
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.550e+00  2.019e-02  -126.3   <2e-16 ***
## pub_yrmo     6.698e-09  1.322e-11   506.7   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 467386  on 284  degrees of freedom
## Residual deviance:  63732  on 283  degrees of freedom
## AIC: 65842
##
## Number of Fisher Scoring iterations: 5
```

Using the model output parameters, I ran predictions for the next two years. The following table summarizes the predicted values for December of each of the preceding three years and the next three years:

```
##   pub_year pub_month published_cve_count future_value
## 1     2019        12                1800           No
## 2     2020        12                4400           No
## 3     2021        12                2900           No
## 4     2022        12                5600          Yes
## 5     2023        12                6900          Yes
## 6     2024        12                8600          Yes
## 7     2025        12               10600          Yes
```

As shown on the following plot, there is a sharp upward trend in the number of vulnerabilities to be published in the near future:



Conclusion

As expected, the number of vulnerabilities published is growing dramatically over time. Based on the model's predictions, we can expect approximately 6,900 CVEs to be published in 2023, 8,600 in 2024, and 10,600 in 2025, compared with the 5,600 expected by the end of 2022. While a more rigorous analysis should be performed if these projections are to be used for concrete budget or staffing purposes, these numbers may serve as a general estimate.