WhiteSource

cyr3con
be in the know, now.

VULNERABILITY PRIORITIZATION THROUGH THE EYES OF HACKERS

What Can We Learn From the Hacker Community?

The need for vulnerability prioritization

Software development teams are constantly bombarded with an increasingly high number of security alerts.

Unfortunately, there is currently no agreed-upon strategy or a straightforward process for vulnerabilities' prioritization. This results in a lot of valuable development time wasted on assessing vulnerabilities, while the critical security issues remain unattended.

Since fixing all vulnerabilities is unrealistic, it's imperative that teams find a method to zero in on the security vulnerabilities that matter. Prioritization is the only way to ensure quick remediation of the most critical issues first, without slowing down development.

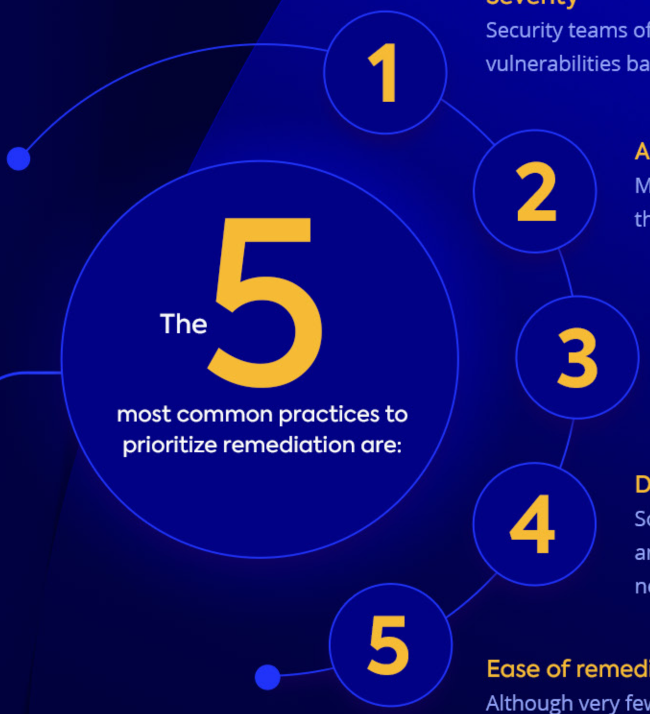
How did we conduct the research?

How is CYR3CON™ tracking hacker communities?

How do organizations prioritize security vulnerabilities?

Since no development team can possibly address all security alerts, different security and development teams look to a number of parameters in their attempt to determine what to remediate first.

We surveyed 300 of our customers in January 2020 to learn how security and development teams are prioritizing vulnerability alerts. These are the five most common considerations:



The 5 most common practices to prioritize remediation are:

1 **Severity**
Security teams often focus on critical and high severity vulnerabilities based on CVSS rating.

2 **Application type**
Mission critical apps or apps with sensitive data are often the first teams address when a vulnerability is discovered.

3 **Popularity**
Since the more popular open source projects provide hackers with the most exploit opportunities, this is another parameter that teams take under consideration.

4 **Disclosure date**
Some organizations can't handle the backlog of vulnerabilities alerts and define a cut-off point from which to start properly addressing newly reported vulnerabilities.

5 **Ease of remediation**
Although very few developers will admit it, the difficulty level of remediation is a deciding factor when it comes to determining what to fix first.

In search of a best practice for prioritization, we decided to look at the dilemma from an additional angle: the hacker community.

We joined forces with CYR3CON, a company which specializes in predicting cyber attacks based on AI gathered from various hacker communities including the dark web and the deep web. CYR3CON's CyRating™ takes factors like discussions in hacker communities, availability of exploits, and exploitation of similar vulnerabilities into account. It captures all information in a single metric, as it is designed to communicate how many times more likely a vulnerability is to be exploited than average.

Cross-referencing the data about the most common open source vulnerabilities in 2019 and CYR3CON's CyRating helped us examine how we should prioritize vulnerabilities from the perspective of hackers.

Chapter 1

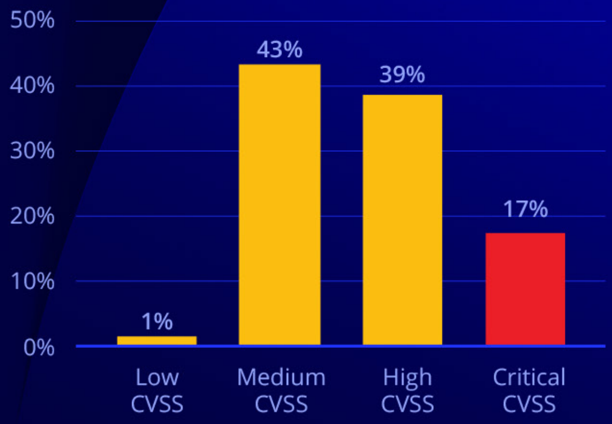
INSIGHTS ON VULNERABILITY PRIORITIZATION

1 Prioritizing based on severity

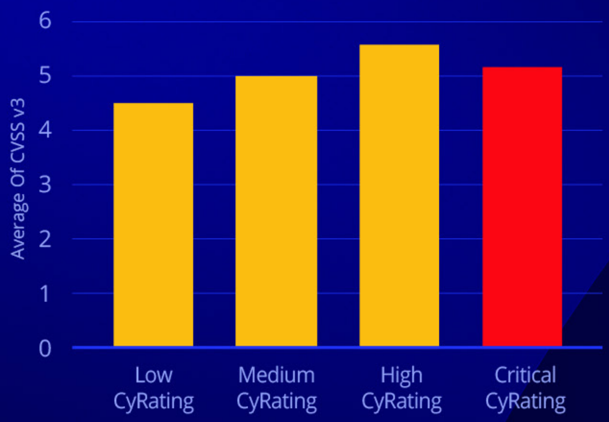
Severity score is the go-to parameter for many security professionals when deciding which vulnerability to fix first, mainly because they want to stick to a standard, and the information is easily accessible and free.

We found that the open source vulnerabilities with higher severity scores didn't get more attention from hackers (higher **CyRating**), meaning hackers aren't focusing on the vulnerabilities with the highest severity score.

Severity distribution for open source vulnerabilities in 2019



CVSS v3 average per CyRating



This is an important reminder that while the severity scoring system aims to cover the characteristics and impact of a security vulnerability, it does not indicate risk. **Risk** is the **impact** times the **likelihood**, but severity only indicates the impact.

2 Prioritization based on application type

Many organizations prioritize vulnerabilities remediation in critical applications, where they know the risk of a breach can be fatal to the company. After all, not all of your applications are equal: some might contain more sensitive data than others.

The problem is not all attackers target only mission critical applications. In addition, it is very difficult to build a standard methodology based on application type when there are so many subjective variables like user audience, mobile or web application, and more.

3 Prioritizing based on popularity

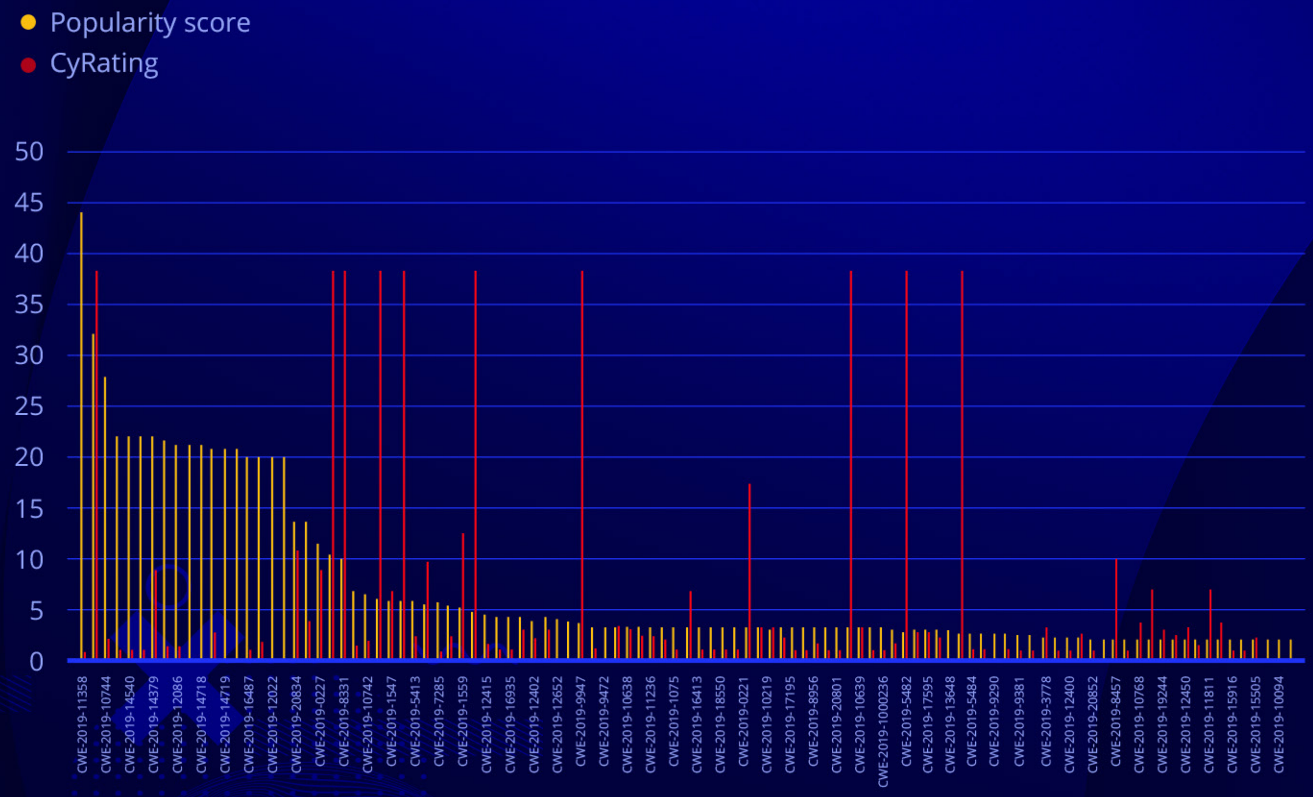
When it comes to open source vulnerabilities, it is believed that attackers favor vulnerabilities in popular open source projects, because one vulnerability could equal millions of targets.

We found that **85** out of the 100 most common open source security vulnerabilities in 2019 had some level of discussion between hackers in their online communities.

However, we didn't find a correlation between how popular an open source project is and the level of discussion in hacker communities.

While a vulnerability in a popular project has a likelihood of getting hackers' attention, there seem to be other parameters that are more significant.

Top open source vulnerabilities and their CyRating



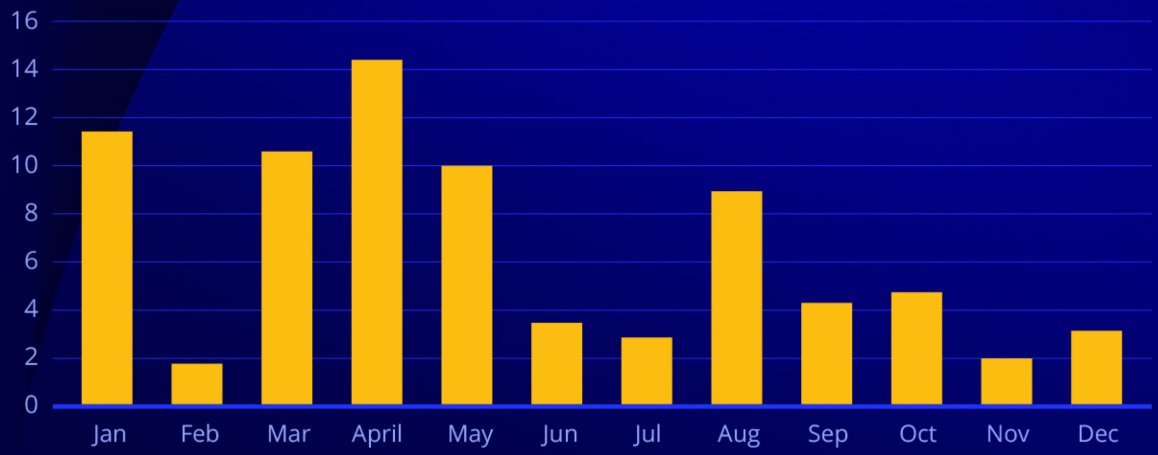
4 Prioritizing based on disclosure date

Our survey results show that organizations that can't handle the backlog caused by years of neglect, make the decision to start fresh from an arbitrary cut-off point.

The tally of really old CVEs that are exploited years later suggests that the oldies are still goodies, reads Verizon's Data Breach Investigations Report (DBIR). "Hackers use what works, and what works doesn't seem to change all that often. Secondly, attackers automate certain weaponized vulnerabilities and spray and pray them across the internet, sometimes yielding incredible success."

CYR3CON data validated the above statement as it shows that vulnerabilities are often discussed over six months following exploitation. It also shows that older vulnerabilities re-emerge in hacker community discussions as they appear in new exploits or malware.

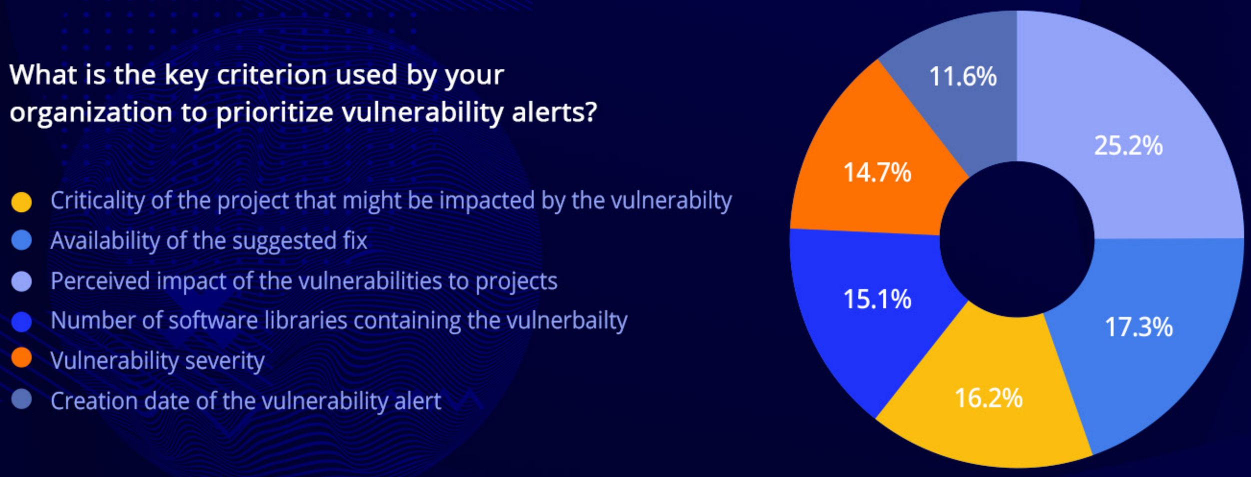
CyRating vs Disclosure



5 Prioritizing based on ease of remediation

Lacking a standard or set process for prioritization, developers often look to the most readily available data when prioritizing remediation, like the availability of a fix. However, this leaves the vulnerabilities that pose the most critical risk open to exploit.

What is the key criterion used by your organization to prioritize vulnerability alerts?



Chapter 2

NEW APPROACHES TO PRIORITIZATION

The fact that we found little to no correlation between any of the common prioritization practices used by development and security teams is concerning.

It tells us that beyond lacking a clear and standardized process for prioritization, the different considerations currently in use barely cover the vulnerabilities that feature in the hacker communities' discussion.

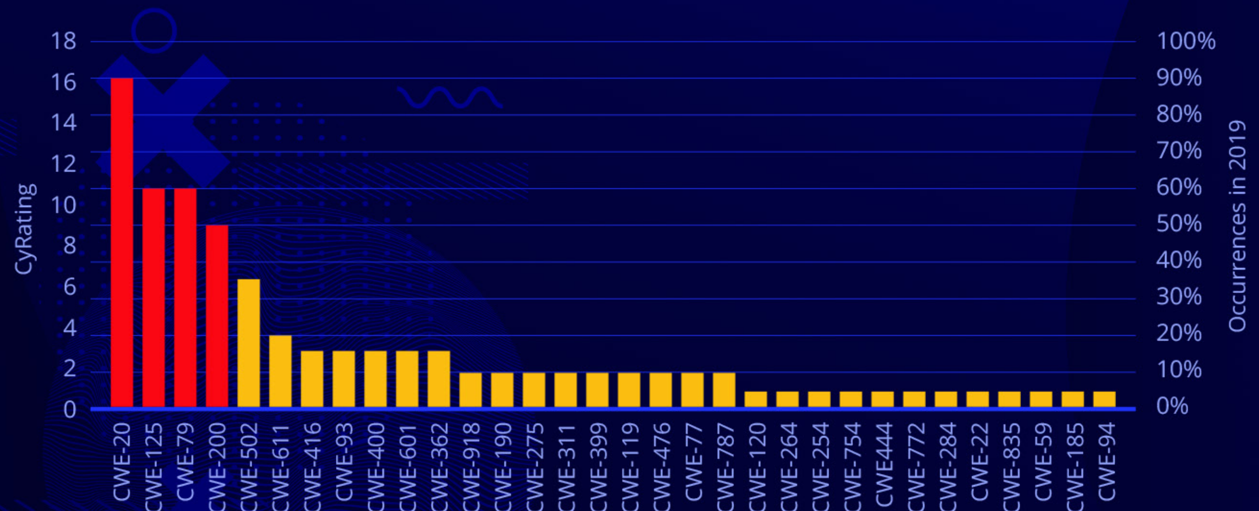
Teams need to find a way to ensure that they are addressing the biggest risks first, and stop wasting valuable time on low-risk vulnerabilities.

In order to achieve this goal, we decided to explore other strategies for prioritization.

6 Prioritizing based on vulnerability type (CWE)

We decided to check the correlation between the vulnerability types (CWEs) and the volume of discussion about those vulnerabilities in hacker communities (CyRating). Results showed that the four CWEs that got the highest CyRating were among the top five most common CWEs in 2019. This teaches us that specific CWEs are targeted by hackers more than others.

Top CWEs in 2019 based on discussions in hacker communities (CyRating)



6 Most common CWEs in open source vulnerabilities in 2019

2019	1	2	3	4	5
	CWE-79	CWE-20	CWE-119	CWE-125	CWE-200
	Cross-Site Scripting (XSS)	Improper Input Validation	Buffer Errors	Out-of-Bounds Read	Information Exposure

The high number of discussions about a handful of CWEs teaches us that hackers prefer to learn one skill and stick to it: they are proficient at very specific types of exploits.

There are number of reasons why these types of vulnerabilities are popular in hacker community discussions.

One is that many automated tools make these vulnerabilities easier to exploit, even for the script kiddies. Another is somewhat of a "chicken-and-egg" issue — the more common a CWE, the more hackers will study it, learn to exploit it, and discuss it. The security research and open source communities, in turn, may focus on discovering and fixing these CWEs to avoid future exploits.

7 Prioritizing based on vulnerability effectiveness

In addition to the methods that we listed and analyzed, there is another key factor that organizations can use in order to prioritize the open source security vulnerabilities that pose a real threat to their applications.

This new factor cuts up to 85% of security alerts, so it can have a significant impact on your developers' remediation speed.

Vulnerability Effectiveness is a new approach in which applications are scanned to detect the direct routes between an organization's proprietary code and the vulnerable method within the open source libraries in use. When proprietary code calls a vulnerable component, the vulnerability is considered effective, because it can impact the proprietary code.

If the proprietary code is calling a harmless component within the library, the vulnerability within this library is treated as non-effective. The vulnerability doesn't pose a risk, since the application is not using it.

Effective
vs
Ineffective

Effective Vulnerability
If the proprietary code is making calls to the vulnerable functionality

Ineffective Vulnerability
If the proprietary code is **NOT** making calls to the vulnerable functionality

SUMMARY

Prioritization: Focus on what matters

> *Perfect security is impossible. Zero risk is impossible. We must bring continuous risk and trust-based assessment and prioritization of application vulnerabilities to DevSecOps.*
>
> 10 Things to Get Right for Successful DevSecOps
> **Neil MacDonald, Gartner**

As the volume of security alerts that development teams deal with continues to rise, it's nearly impossible to remediate every vulnerability. The burning question is: how do we best prioritize remediation?

The data shows some of the most popular prioritization methods don't address the security vulnerabilities that the hacker community focuses on. Relying on the most accessible parameters rather than what actually exposes an organization to risk leaves applications open to attacks.

Organizations must implement a solid prioritization method in order to ensure that they are focusing on the most critical issues first, and not wasting valuable time fixing low-risk vulnerabilities while leaving high risk windows wide open.

How did we conduct the research?

We pulled the top 100 most common open source vulnerabilities reported in 2019 based on more than 100,000 applications with their CVSS rating, popularity, CWE, disclosure date, and more.

CYR3CON then provided the CyRating for each vulnerability. The CyRating score indicates the level of discussions among hackers on each vulnerability and the intent of the hacker community to exploit that vulnerability.

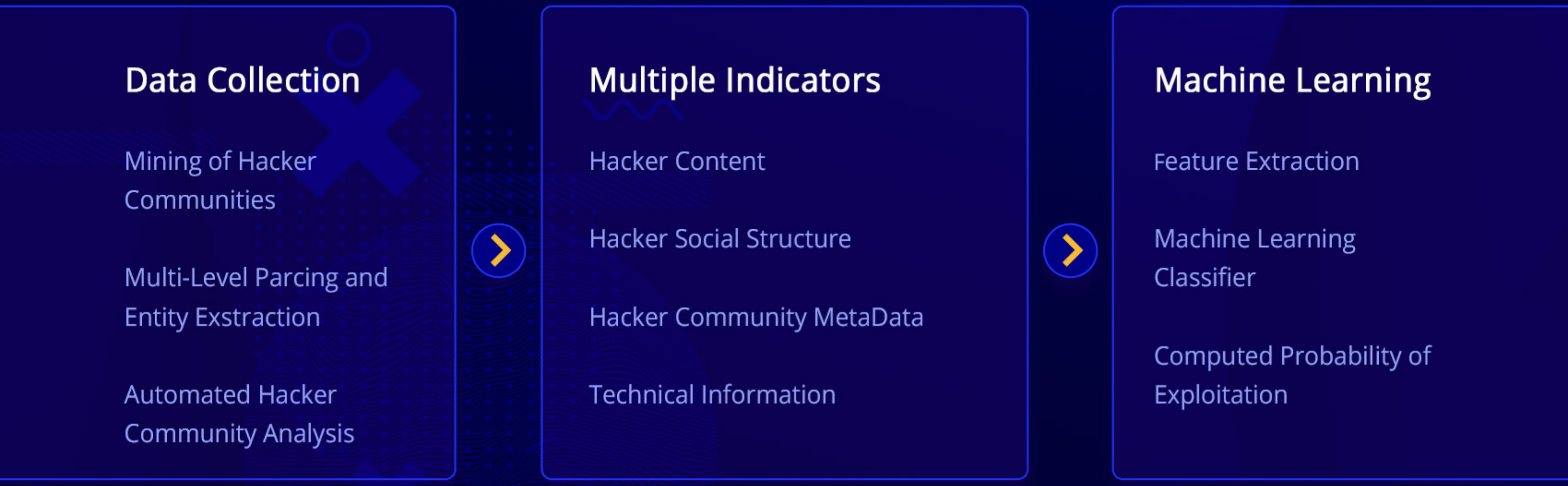
How is CYR3CON tracking hacker communities?

cyr3con be in the know, now. specializes in predicting and preventing cyber attacks based on AI gathered from the various hacker communities including the dark web and the deep web.

It predicts cyber attacks by calculating the CyRating score based on threat intelligence across three phases: data collection, indicator extraction, and model training.

The CyRating score is determined through the use of a peer-reviewed supervised machine learning approach where threat intelligence from various hacker community sources is aligned with information on exploits in the wild to produce the prediction.

CYR3CON(R) and CyRating(R) are registered trademarks of Cyber Reconnaissance, Inc., all rights reserved.



Data Collection

Mining of Hacker Communities

Multi-Level Parcing and Entity Exstraction

Automated Hacker Community Analysis

Multiple Indicators

Hacker Content

Hacker Social Structure

Hacker Community MetaData

Technical Information

Machine Learning

Feature Extraction

Machine Learning Classifier

Computed Probability of Exploitation