

# Patching Optimization

© Rob Hulsebos

Version 11-August-2022

## I ntroduction

Anyone active in (industrial) cybersecurity knows that the cadence of our work is often set by the publication of vulnerabilities. When a system is affected by the vulnerability, the next step would then be to patch it as soon as possible.

But this is not always an effective way of working. First, in a larger system, there are far more vulnerabilities than one can handle. Installing patches in an industrial system requires more work than in an IT environment, and while this activity is done no production can be run. In an industrial environment, being able to produce is priority #1. Stopping production for installing a patch can't be done too often, so it is advised to install only the patches for really dangerous vulnerabilities ASAP, and leave the others for a more convenient moment (i.e. December 25<sup>th</sup>).

Second, having a vulnerability in a system doesn't necessarily mean that a hacker somewhere in the world is exploiting it already. Many vulnerabilities are never exploited, for the simple reason that nobody finds it worthwhile to hack. But we don't know in advance which vulnerability is exploitable (statistics: 2% - 7% of all published vulnerabilities are exploited). Because of the uncertainty, we just (for surety) patch the vulnerability, which sometimes is a total waste of effort. But we never know this.

### **Prioritization**

Clearly, a method is needed to prioritize the available patches for all new vulnerabilities. There is no standard accepted method for this, not even in the IEC-62243 standard for industrial cybersecurity. Some use the CVSS-score (i.e. "All CVSS scores > 9"), other look for available (publicly known) exploits, others only patch the vulnerabilities in the "crown jewel" devices, and others rely on their vendor's advice, or patch only the 'crown jewels'. And some users never patch.

### **A better way**

To assist those working on this subject, the "Exploit Prediction Scoring System" (EPSS) has been developed by FIRST (first.org) to help prioritize the remediation of vulnerabilities. A vulnerability will be given a score (in the range 0..100%). The higher the score, the higher the likelihood that the vulnerability is exploited, and is this reason to keep a tab on it.

The score is calculated by collecting data from multiple sources, feeding it into the EPSS model, which is trained with 1164 variables. Based on the results, using EPSS v2 helps organizations to patch fewer than 20% of the vulnerabilities while simultaneously reducing the risk.

# 1 The challenge

The number of vulnerabilities in products rises each year. The following figure shows the year-to-year rise in published vulnerabilities ("CVEs"). As can be seen, 2022 has a 20% increase in comparison to 2021.

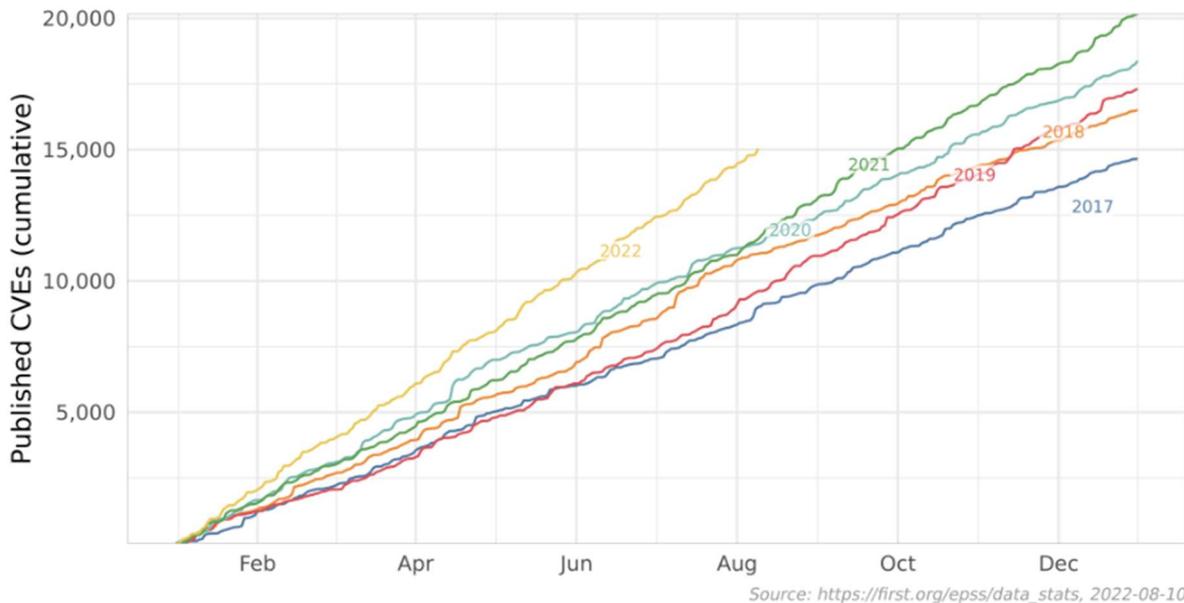


Figure 1: Cumulative count of vulnerabilities published per year

What is the root-cause of the increase? It could be that more vulnerabilities are reported (whereas they were kept secret in the past), but also an increase in research and/or better tooling.

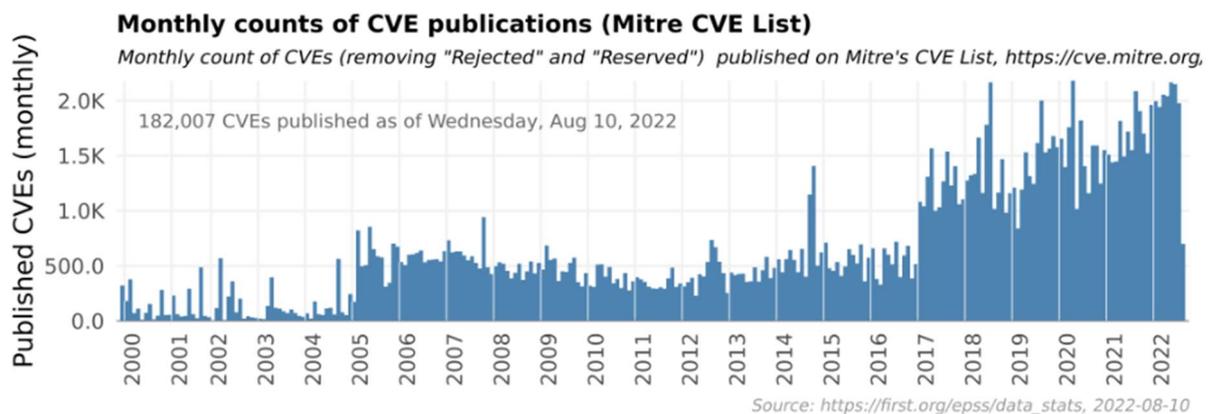


Figure 2: Vulnerabilities published per month

Anyway, for an asset-owner it means more work: keeping track of all vulnerabilities, and then checking if a vulnerability is applicable to his systems (= do I have the affected hardware / software) ? If yes, an available patch needs to be installed, or the vulnerability may need a mitigation (if possible).

# 2 The landscape

Of all the vulnerabilities in hardware and software, only a fraction is ever published. The others are known only to hackers and/or the companies of the affected products. For the hackers, there is money to be made by selling vulnerabilities<sup>1</sup>. For companies, a reason not to publish about vulnerabilities is the fear of loss of reputation. In both cases, there's not much for an asset owner to do about it (so for the remainder of this document, we'll ignore the "unknown" vulnerabilities).

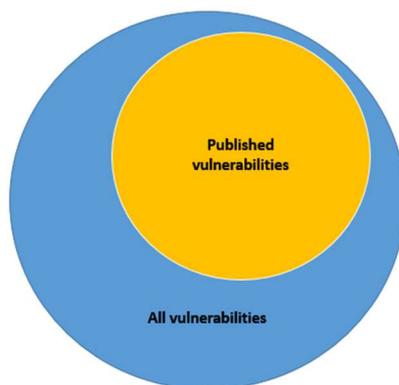


Figure 3: There's more vulnerabilities than those published (not to scale)

The best known source for information about published vulnerabilities is NVD – National Vulnerability Database (nvd.org), which is tracking vulnerabilities since 1999. In 2021 there were 20142 new vulnerabilities added, up from 18351 in 2020.

These 20142 vulnerabilities relate to 25223 different products, as known to the NVD. An example of affected software (for CVE-2022-30333) is shown below:

### Configuration 1 ([hide](#))

<b>cpe:2.3:a:rarlab:unrar:*:*:*:*:*</b> <a href="#">Hide Matching CPE(s)</a> ▲ <ul style="list-style-type: none"><li>cpe:2.3:a:rarlab:unrar:0.0.1:*:*:*:*</li><li>cpe:2.3:a:rarlab:unrar:5.5.4:*:*:*:*</li><li>cpe:2.3:a:rarlab:unrar:5.5.6:*:*:*:*</li><li>cpe:2.3:a:rarlab:unrar:5.6.1.2:*:*:*:*</li><li>cpe:2.3:a:rarlab:unrar:5.6.1.3:*:*:*:*</li><li>cpe:2.3:a:rarlab:unrar:6.0.3:*:*:*:*</li></ul>	<b>Up to (excluding)</b> <b>6.12</b>
<b>Running on/with</b>	
<b>cpe:2.3:a:linux:linux_kernel:*:*:*:*</b> <a href="#">Show Matching CPE(s)</a> ▼	
<b>cpe:2.3:o:opengroup:unix:*:*:*:*</b> <a href="#">Show Matching CPE(s)</a> ▼	

Figure 4: Products / releases affected by a vulnerability

<sup>1</sup> To know more about this market, read the excellent book "This is how they tell me the world ends: the cyber weapons arms race" by Nicole Perlroth (ISBN 978-1635576054).

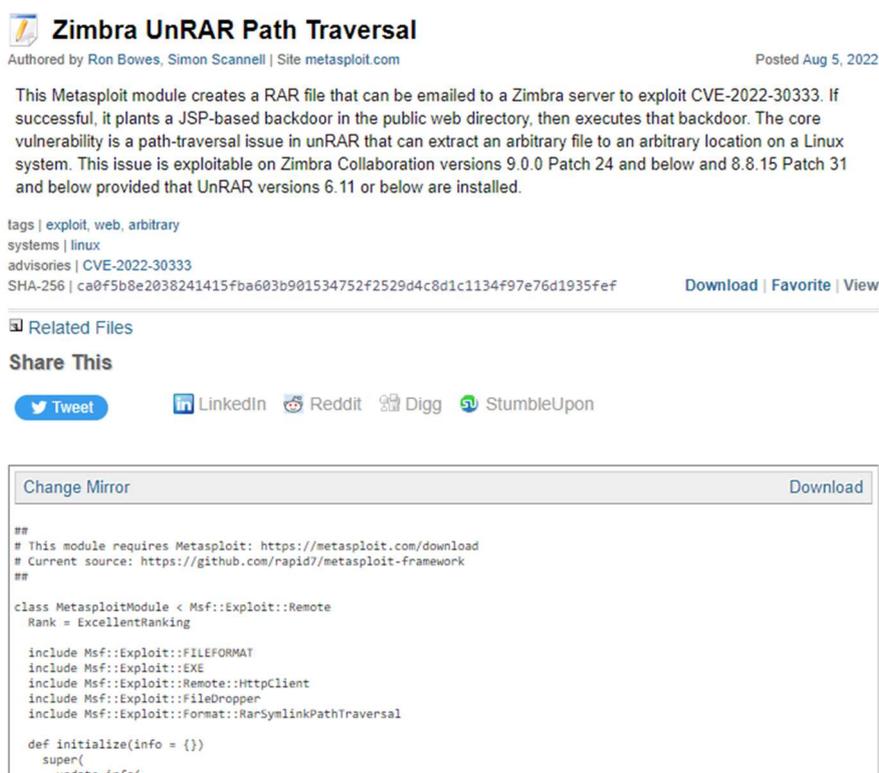
## Exploitable vulnerabilities

Some vulnerabilities are more interesting for hackers than others. For example, Windows vulnerabilities grab more attention than a vulnerability in a little-used product. Interesting other targets are Android, software-libraries (SSL, Log4J) that are incorporated in many other products, and products with millions of users (WordPress, Apache, Java, Oracle, VMWare, etc.), and products of market leaders (i.e. Cisco).

CVEs usually do not contain much usable information about the vulnerability, to give malicious actors no head-start. Nevertheless, both hackers and researchers want to know every detail, as this is needed to write detection scripts (i.e. in Intrusion Detection Systems, firewalls, tools like Snort, Metasploit, etc.). Depending on the complexity of the vulnerability, some exploits appear within a day, but usually it takes about a week. Publications about it follow each other in quick succession, with variations of the exploit. Sometimes this research leads to the detection of new a new vulnerability (as happened with log4J).

An exploit usually is a small piece of software, often written in Python, that shows that the vulnerability is really there, and can be used for nefarious purposes: for example, to reboot a device, install other software, extract interesting data, erase the disk, etc. However, most exploits do not damage anything.

An example of an exploit for CVE-2022-30333 (as found on packetstormsecurity.com):



The screenshot shows a Metasploit module page titled "Zimbra UnRAR Path Traversal". The page includes the author information (Ron Bowes, Simon Scannell), the date (Aug 5, 2022), and a detailed description of the exploit. The description states that the module creates a RAR file that can be emailed to a Zimbra server to exploit CVE-2022-30333. It also provides the core vulnerability details and the versions of Zimbra Collaboration and UnRAR that are affected. Below the description, there are tags for "exploit, web, arbitrary systems | linux" and "advisories | CVE-2022-30333". The SHA-256 hash is also provided. There are links for "Download", "Favorite", and "View". A "Related Files" section is visible, followed by a "Share This" section with social media icons for Twitter, LinkedIn, Reddit, Digg, and StumbleUpon. At the bottom, there is a "Change Mirror" button and a "Download" button. A code block shows a small part of the Python code for the module.

```
##
# This module requires Metasploit: https://metasploit.com/download
# Current source: https://github.com/rapid7/metasploit-framework
##

class MetasploitModule < Msf::Exploit::Remote
  Rank = ExcellentRanking

  include Msf::Exploit::FILEFORMAT
  include Msf::Exploit::EXE
  include Msf::Exploit::Remote::HttpClient
  include Msf::Exploit::FileDropper
  include Msf::Exploit::Format::RarSymlinkPathTraversal

  def initialize(info = {})
    super(
      update info{
```

Figure 5: Exploit for CVE-2022-30333 and a small part of the Python code

## How many exploits are there?

Not all vulnerabilities lead to an exploit. This is good news for users, as a vulnerability that cannot be exploited<sup>2</sup> is not dangerous.

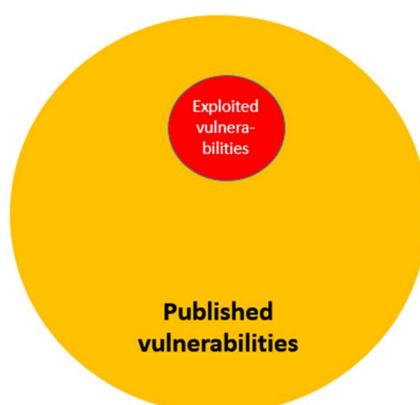


Figure 6: Not all vulnerabilities are exploitable (not to scale).

As mentioned, only a subset of all public known vulnerabilities will have an exploit developed, only 2% - 7% (according to FIRST). Small as it may seem, given the number of vulnerabilities published, this is still 40 – 140 per year. But this doesn't mean that your organization also 'only' has a 2% - 7% likelihood of being vulnerable – one vulnerable device could be enough for a hacker to strike.

Exploits are collected by the site exploit-db.com (containing over 45000 entries). Other sources for exploits are github.com, or on researcher's websites.

Date	D	A	V	Title	Type	Platform	Author
2022-01-05	+			Siemens S7 Layer 2 - Denial of Service (DoS)	DoS	Hardware	RoseSecurity
2019-11-14	+			Siemens Desigo PX 6.00 - Denial of Service (PoC)	DoS	Hardware	LiquidWorm
2019-07-10	+			Siemens TIA Portal - Remote Command Execution	Remote	Hardware	Joseph Bingham
2019-04-23	+		✓	Linux - Missing Locking in Siemens R3964 Line Discipline Race Condition	DoS	Linux	Google Security Research
2018-05-30	+			Siemens SIMATIC S7-300 CPU - Remote Denial of Service	DoS	Linux	14rk3vilz
2018-05-23	+			Siemens SCALANCE S613 - Remote Denial of Service	DoS	Linux	14rk3vilz
2018-05-22	+			Siemens SIMATIC S7-1500 CPU - Remote Denial of Service	DoS	Linux	14rk3vilz
2018-05-22	+			Siemens SIMATIC S7-1200 CPU - Cross-Site Request Forgery	WebApps	Linux	14rk3vilz
2018-05-21	+			Siemens SIMATIC S7-1200 CPU - Cross-Site Request Forgery	WebApps	Linux	14rk3vilz
2018-02-16	+			Siemens SIPROTEC 4 and SIPROTEC Compact EN100 Ethernet Module < 4.25 - Denial of Service	DoS	Hardware	M. Can Kurnaz
2016-08-19	+		✓	SIEMENS IP Cameras (Multiple Models) - Credential Disclosure / Configuration Download	WebApps	CGI	Todor Donev

Figure 7: Collection of exploits on Exploit Database website

## Which exploits are dangerous for me?

Luckily, many exploits are no reason for concern, as you may not have the affected hardware / software in your system. For example, if there is an exploit for Apache's webserver software, but you do not have this software, there is nothing to worry<sup>3</sup>

<sup>2</sup> Of course, we never know if there is an exploit developed by a hacker who'd rather keep it secret.

<sup>3</sup> You'd better be 100% sure that you really don't have this software – experience shows that in practice many users have no idea what's in their systems. But that is a topic for another article.

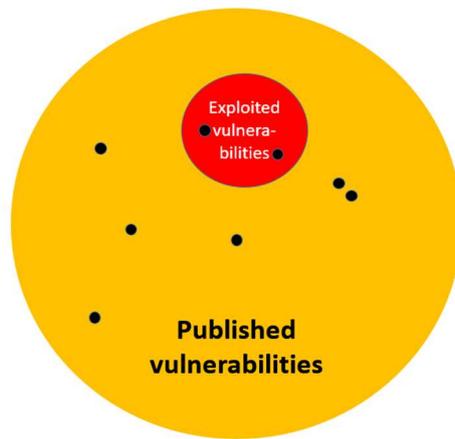


Figure 8: Vulnerabilities (black spots) present

As an example, figure 8 shows that there are 8 vulnerabilities<sup>4</sup> in a certain system, of which 2 are exploitable.

Our first priority would be to patch these two. To identify the 2 exploitable vulnerabilities out of the 10 (or likely many more in a large organization!), EPSS can help.

### **Focus on exploitable vulnerabilities**

The EPSS is intended to help users prioritize which vulnerabilities to patch. A vulnerability is given a score in the range 0 – 100%; the higher the value, the greater the likelihood that the vulnerability will be exploited within the next 30 days.

The EPSS model ingests data from various sources, on a daily base. This means that the EPSS score of a vulnerability can change from day to day, depending on new information that becomes available. Some sources of information are:

- The MITRE CVE list (cve.org), for all “published” CVEs
- Certain (text-based) tags in the CVE description or other sources
- The age (in days) since the CVE has been published
- Count of how many references are listed in the CVE
- Published exploit code, as found in Metasploit, ExploitDB or Github
- Results from security scanners (like Jaeles, Intrigue, Nuclei, sn1per)
- CVSSv3 vector as published by NVD (nvd.nist.gov)
- CPE (vendor) information as published by NVD
- Daily observations of exploitation “in the wild” as reported via AlienVault or FortiNet

Altogether, there are 1164 variables in the EPSS model (now at version 2022.01.01). Some are simple, like “Is Microsoft the vendor?”, or “Is it supported by Metasploit?”, which gives a higher likelihood for exploitation. In case a CVE is referenced multiple times the score increases too. The age of a CVE is also taken into account (peaking around 6 years and then reducing). We cannot discuss all 1160 other variables, for more information about this refer to the research paper listed on the EPSS website.

### **Some results**

The EPSS website provides a daily update. The following snapshots were taken on Friday, August 5, 2022. It states that the EPSS model knows about 181576 CVEs, of which 58 were new on that day.

---

<sup>4</sup> This assumes that you exactly know which hardware and software is present in your system.

The first statistic shows the 24 top rated CVEs from the last 2 days. As can be seen, the EPSS score per CVE is (still) low, showing that not much information is publicly available.

### Top rated CVEs from the last two days

We selected the 24 highest rated CVEs published in the last 48 hours. They are shown here with the CVE and EPSS score.

<b>CVE-2022-32292</b> 3.9%	<b>CVE-2022-25168</b> 2.7%	<b>CVE-2022-37030</b> 2.5%	<b>CVE-2022-31793</b> 1.1%	<b>CVE-2022-32964</b> 1.1%	<b>CVE-2022-35864</b> 1.1%
<b>CVE-2022-35619</b> 3.9%	<b>CVE-2022-28684</b> 2.7%	<b>CVE-2022-28668</b> 2.3%	<b>CVE-2022-2272</b> 1.1%	<b>CVE-2022-34871</b> 1.1%	<b>CVE-2022-35866</b> 1.1%
<b>CVE-2022-35620</b> 3.9%	<b>CVE-2022-35867</b> 2.6%	<b>CVE-2022-32293</b> 1.5%	<b>CVE-2022-27621</b> 1.1%	<b>CVE-2022-34872</b> 1.1%	<b>CVE-2022-21186</b> 1.0%
<b>CVE-2022-35865</b> 3.9%	<b>CVE-2022-37396</b> 2.6%	<b>CVE-2022-32965</b> 1.2%	<b>CVE-2022-32963</b> 1.1%	<b>CVE-2022-35216</b> 1.1%	<b>CVE-2022-34974</b> 1.0%

Source: [https://first.org/epss/data\\_stats](https://first.org/epss/data_stats), 2022-08-05

Figure 9: EPSS statistic showing top rated CVEs

When vulnerabilities are older, more information about them can become available (i.e. an exploit is published, more details about the vulnerability become known, etc.). This is shown in the following figure.

### CVEs with shifting EPSS scores

EPSS scores can shift around because of new information (e.g CPE data is available now, an exploit is published, etc)

<b>CVE-2022-30333</b> 73.3% <b>+67.5%</b>	<b>CVE-2013-1861</b> 21.8% <b>+6.2%</b>	<b>CVE-2010-1184</b> 16.4% <b>-2.9%</b>	<b>CVE-2022-34120</b> 1.2% <b>-1.2%</b>	<b>CVE-2022-21785</b> 1.0% <b>-1.5%</b>	<b>CVE-2022-26426</b> 1.0% <b>-2.5%</b>
<b>CVE-2022-26138</b> 16.5% <b>+13.8%</b>	<b>CVE-2022-24675</b> 19.2% <b>+4.3%</b>	<b>CVE-2008-0020</b> 55.1% <b>-0.7%</b>	<b>CVE-2022-34556</b> 1.2% <b>-3.6%</b>	<b>CVE-2022-21790</b> 1.0% <b>-0.5%</b>	<b>CVE-2022-26427</b> 1.0% <b>-2.5%</b>
<b>CVE-2022-31206</b> 1.4% <b>-0.1%</b>	<b>CVE-2022-29078</b> 12.5% <b>+3.5%</b>	<b>CVE-2010-4180</b> 9.9% <b>+2.3%</b>	<b>CVE-2012-2271</b> 8.6% <b>-1.5%</b>	<b>CVE-2022-21791</b> 1.0% <b>-1.5%</b>	<b>CVE-2022-26428</b> 1.0% <b>-2.5%</b>
<b>CVE-2022-31207</b> 1.4% <b>-0.1%</b>	<b>CVE-2022-36946</b> 4.5% <b>+3.1%</b>	<b>CVE-2022-26437</b> 1.2% <b>-3.6%</b>	<b>CVE-2022-21785</b> 1.0% <b>-1.5%</b>	<b>CVE-2022-21792</b> 1.0% <b>-2.5%</b>	<b>CVE-2022-26429</b> 1.0% <b>-2.5%</b>

Source: [https://first.org/epss/data\\_stats](https://first.org/epss/data_stats), 2022-08-05

Figure 10: EPSS statistic showing increase / decrease of CVEs

Here we see that CVE-2022-30333, about the unRAR tool on Linux, has a 67.5% higher score. So there is interest in this vulnerability, and definitely a vulnerability to check (not) being present on your systems.

### Is EPSS always right?

A score of 100% for a vulnerability doesn't give a 100% certainty that there will be an exploit. There can be a lot of interest in the vulnerability, but in the end nobody found it worthwhile to put work in it. So in this case we have a "false" positive.

Similarly, a vulnerability for which nothing could be found, can still have an exploit, it takes just one hacker silently working on it. This is a "false" negative (the EPSS model said there wouldn't be an exploit, but there came one anyway).

# 3 Some points to note

## **OT vulnerabilities**

The EPSS puts a heavy emphasis on CVEs, but for OT systems this doesn't give a good coverage. Certain CVEs are applicable to OT products, but these are not listed by the NVD (the so-called "CPE"), and are thus not taken into account into the EPSS score calculation.

*An example of this is a vulnerability in the SSL library. This library is used by many industrial vendors, for example Siemens. This company publishes its own advisory SSA-712929 about the vulnerability CVE-2022-0778, listing 22 pages (!) with affected products. None of these are listed by the NVD; it also doesn't list other industrial vendors affected by this vulnerability. Peculiar is that the Siemens advisory is mentioned by MITRE on <https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-0778>.*

It also regularly occurs that a vendor updates their own advisory, but the updates are not propagated to the CVE.

## **Vulnerabilities without CVE**

Not all OT vulnerabilities get assigned a CVE (a choice of the vendor).

*An example of this is Yokogawa advisory<sup>5</sup> YSAR-22-0002. It is unclear why exactly this advisory has no CVE, while Yokogawa does assign CVEs usually.*

A peculiar example is an advisory by Moxa which does not have a CVE, but is published in Russia (at FSTEC) because the vulnerability was discovered by Russian researchers.

## **ICS-CERT**

An alternative database for industrial vulnerabilities is ICS-CERT<sup>6</sup>, founded in 2003 to collect and publish information about vulnerable industrial products at a time when vendors published nothing. Now, this has changed; some vendors publish their vulnerabilities at ICS-CERT, but not always, and some vendors never publish at ICS-CERT.

## **Summary**

For OT systems the EPSS score may not accurately reflect the real-life situation of the threat landscape. So it is advised to not solely rely on the EPSS score of vulnerabilities, and monitor your vendor(s) directly (some advisories may not end up in the EPSS model).

# 4 Conclusion

The EPSS helps to reduce the workload of patching systems by identifying vulnerabilities which have a high likelihood of being exploited. Tackle these vulnerabilities first, and the rest later. According to EPSS v2, organizations have to patch fewer than 20% of the vulnerabilities they would have mitigated, compared to using a strategy based on CVSS scores.

For more information about the EPSS model: see <https://www.first.org/epss/model>.

---

<sup>5</sup> [web-material3.yokogawa.com/1/32133/files/YSAR-22-0002-E.pdf](https://web-material3.yokogawa.com/1/32133/files/YSAR-22-0002-E.pdf)

<sup>6</sup> [www.cisa.gov/uscert/ics/advisories](https://www.cisa.gov/uscert/ics/advisories)

*This article is likely not complete in describing all EPSS details.  
If you have any suggestions, comments or additions,  
please do not hesitate to contact me (email: [rh\[at\]enodenetworks.com](mailto:rh[at]enodenetworks.com)).*