Patching 101

© *Rob Hulsebos*

Version 1.0, 6-January-2022

Introduction

Organizations looking for advice on cybersecurity probably encounter the advice to "patch" their equipment. Patching is one of the most effective methods to keep malware out (but not a 100% guarantee). But exactly what *is* patching? Everyone who ever owned a PC, laptop or tablet or smartphone undoubtedly has seen the message that "This PC must be restarted to install updates", usually in the second week of each month. This is the final step, Windows has already done all the work, and the restart of the PC allows the installation to be finished.

The user doesn't have to do much, the nuisance of the necessary restart is all the we see of the installation of the new software. But for industrial equipment, patching is not so automatic. It requires more knowledge from the user, more work, more planning, sometimes additional equipment, and more can go wrong. Also, an industrial system may need to be stopped while the patch is in progress. This is a much unwanted side-effect of patching.

Figure 1: Message on an iPhone about a new update being available.

Because I could find no documentation for novice users on how to patch industrial systems, I decided to write this document "Patching 101" for anyone interested in the subject. The "Patching 102" explains the patch process. Finally, "Patching 103" describes what IEC-62243 has to say about patching, and a short summary on current insights.

Why is patching done?

Patching is necessary when there's something wrong with software. A new version of the software needs to be installed to fix that software. This is called (depending on the vendor) a patch, a hotfix, or an update. The existing functionality remains the same.

This differs from the situation where the vendor has a new version of the software, with new features (perhaps you need to pay for this). This is a new release, either a major release (say from version 3.5 to 4.0) or a minor release (from 3.11 to 3.12).

Reasons for patching are usually bugs (programming errors) in the software. Sometimes a bug is dangerous, meaning it could be exploited by hackers. It is then called a "vulnerability". A definition of a vulnerability (as used in the National Vulnerability Database) is:

> *"A weakness in the computational logic (i.e. code) found in software and hardware components that, when exploited, results in a negative impact to confidentiality, integrity or availability".*

A "weakness" in code can be due to:

- A programming error ("bug"). There are a few hundred possible types of bugs known.
- An architectural issue. For example, the WiFi "WPS" algorithm supposedly has 10^8 possible pin codes, but in practice it has only 20000.
- A design issue. For example, a timing issue which can be used to circumvent certain checks on allowing access to a device.
- A configuration issue. For example, a device has a hard-coded and unmodifiable password which everybody in the world knows (after some time).
- The passage of time. For example, a device has a password of 6 characters which was strong enough 20 years ago, but is easy to crack today.

Note that vulnerabilities can also occur in hardware, for example in a CPU, or a complex chip (i.e for Ethernet, WiFi or Bluetooth).

Not all bugs are created as vulnerability

A bug in software is not necessarily also a vulnerability. Indeed, most bugs are not. They do make an application work otherwise as expected, but it may have no negative effect at all on a production system. For example, a programmer may forget[1] that some years are leap years, not allowing you to enter 29 February as a date.

How are vulnerabilities discovered

Where does knowledge about vulnerabilities originate? Many researchers and hackers are looking into the firmware of all sorts of devices in the hope of finding a new vulnerability. Why? For honour and glory, but also to make money in a legal way (after reporting them, a financial reward may follow) or to make an illegal way (exploiting them via malware or ransomware).

When a vulnerability is found, the following can happen:

1) It may be misused, in malware, to attack equipment. Luckily, most vulnerabilities are not used in malware. But there is never a guarantee.
2) It can be sold to a company that buys this knowledge, for re-selling them to others.
3) It can be reported to the vendor of the device. The vendor can either use this information to create a patch, or ignore the information (do nothing). The time a vendor needs to fix a vulnerability and issue a patch is usually several months.
4) It can be reported to the CVE or ICS-CERT, who publishes it, and reports it to the vendor.
5) It is disclosed in a publication or a talk at a security conference. Responsible researchers give the vendor of the device a few months to fix the vulnerability

[1] Quite common, as an example see: https://developer.apple.com/forums/thread/134090 and https://codeofmatt.com/list-of-2016-leap-day-bugs/

before going public, but if the vendor is slow or does nothing, the publication / talk proceeds anyway.

Why is patching needed?
Software that has vulnerabilities may be exploited by hackers. That's why it is important to install a patch as soon as possible, it closes the opportunity for abuse. When a vulnerability becomes publicly known, hackers may be able to exploit it within one day. Luckily, this doesn't happen with all vulnerabilities[2].

Is patching *always* needed?
The general advice is: yes. But in practice there are just too much devices with vulnerabilities and their corresponding patches. You'd be busy *every* day in a larger installation. Because the installation of a patch often means that production must be stopped (downtime!), there is a growing concern about the impracticality of this. Also, installations may sometimes go wrong, causing more downtime. In total you might have more downtime than a hacker could ever cause.

Risk-based patching
A new way of thinking about how to install patches is to do *risk-based patching*. Based on how dangerous a vulnerability is, and the possible impact on a system (i.e. negatively affecting process safety), new patches are divided in groups: those to be installed immediately, those to be installed later, or those to be never installed.

What is patching ?

Patching is the activity where software with a vulnerability is replaced by (new) software *without* the vulnerability. Usually this is done by overwriting the older software with the newer software, i.e. by copying the a new file over the older file, or overwriting the contents of flash memory (in embedded devices).

Some vendors deliver the patch as a set of one or more files, and possibly an installation utility. By executing the installation utility, the correct files are patched. With larger software programs only the affected files are replaced, it is not a full installation. This saves a lot of time (both in download and installation). For embedded devices, the patch usually replaces the complete content of the flash memory.

Cost
Patches are usually provided without cost by vendors, at least as long a product is under support. After the support period, patches are sometimes made only available for paying customers (with an extended support contract).

For industrial equipment, the support period for a device is something to watch. If the vendor stops support after 5 years but you still plan to use to device to coming 10 years, there is a certain risk in not getting support for (as of yet unknown) vulnerabilities. An alternative may be to replace the device by a newer version.

Restart / reboot
It is not always possible to replace software while it is running. This means that in order to install a patch, the currently active software must be stopped. For industrial systems this may mean that a machine or production line must be stopped; in a building it may mean that access control systems are inactive so entry to certain rooms or floors is not

[2] See https://www.zdnet.com/article/only-5-5-of-all-vulnerabilities-are-ever-exploited-in-the-wild/

possible, or that security camera's are inoperable, etc. It is always inconvenient to have systems "down" while installing a patch.

In some systems, it is possible to install a patch without having to shut down the device first. For example, on Windows PC's the patch is installed on a side track (elsewhere on disk). But it must still be installed on the definitive location, for which Microsoft requires a reboot (restart) of the PC.

The requirement to stop applications, or restart / reboot a device, is a major hindrance in the installation of patches. In some companies it leads to a situation where patches are not installed at all, or sometimes not for years.

Functional updates

Some companies do not distribute patches standalone. Instead, they combine them with functional updates of their software. Sometimes this has a drawback, because the new functionality may require that you need to make changes to your application software too. This is more (usually unexpected) work, which may delay the installation of the new software with the patch for a vulnerability.

Note that some companies do not allow functional updates to be installed without additional licensing. If you do not buy this, the functional update cannot be installed and you will not get the patch!

Workaround

For some vulnerabilities, a patch may not be provided, for example because a particular product is out of support. Instead, a vendor may suggest a workaround procedure, i.e. disable a particular feature of a device. Whether this is feasible or not, depends on how the device is used in your system.

Terminology

In relation to vulnerabilities and patching, the cybersecurity community uses specific terminology that you will encounter in documentation, websites and advisories. We will briefly explain in this chapter:

- CVE Common Vulnerability Enumeration
- NVD National Vulnerability Database
- CVSS Common Vulnerability Scoring System
- ICS CERT Industrial Control Systems CERT[3]

What is a CVE ?

At the end of the previous century, every cybersecurity company in the world used its own system of describing a vulnerability. The same vulnerability could have different names and descriptions depending on the company. Apart from the inevitable confusion, it made it difficult to users to keep track of what was going on, depending upon who they talked to.

Since 1999, the "CVE – Common Vulnerability Enumeration" system is in operation. Its mission is to identify, define, and catalog publicly disclosed vulnerabilities. Each vulnerability has a unique identification, for example "CVE-2021-24488". This is called the "CVE ID" (Identifier). The first number is the year, the second number is randomly assigned.

The catalog of all CVE ID's is kept on the website https://nvd.nist.gov: the National Vulnerability Database. In contrast to its name not only the US vulnerabilities are catalogued here, but of any country.

Currently (2022), the CVE list grows by more than 20000 per year[4].

[3] Computer Emergency Response Team

[4] Despite this, you may encounter CVE-ID's with six-digit numbers due to the way ranges of ID's are assigned to organizations, for example CVE-2018-1000622.

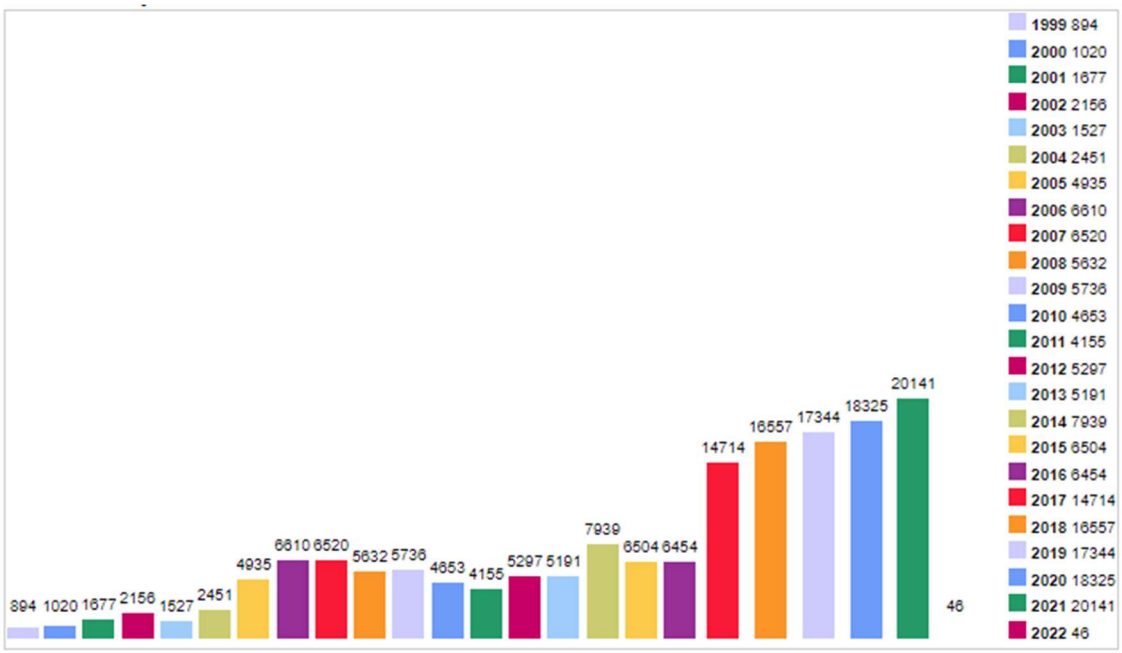


Figure 2: Number of issued CVE's per year (and 46 for the first three days in 2022). Source: cvedetails.com

Each CVE follows the same format (see figure 3), which makes them easy to interpret. In the middle, section "Severity", a severity value called "CVSS Score"is shown, which we will discuss below. This particular vulnerability has a CVSS Score of 7.8, which is rated as "High" (dangerous!).

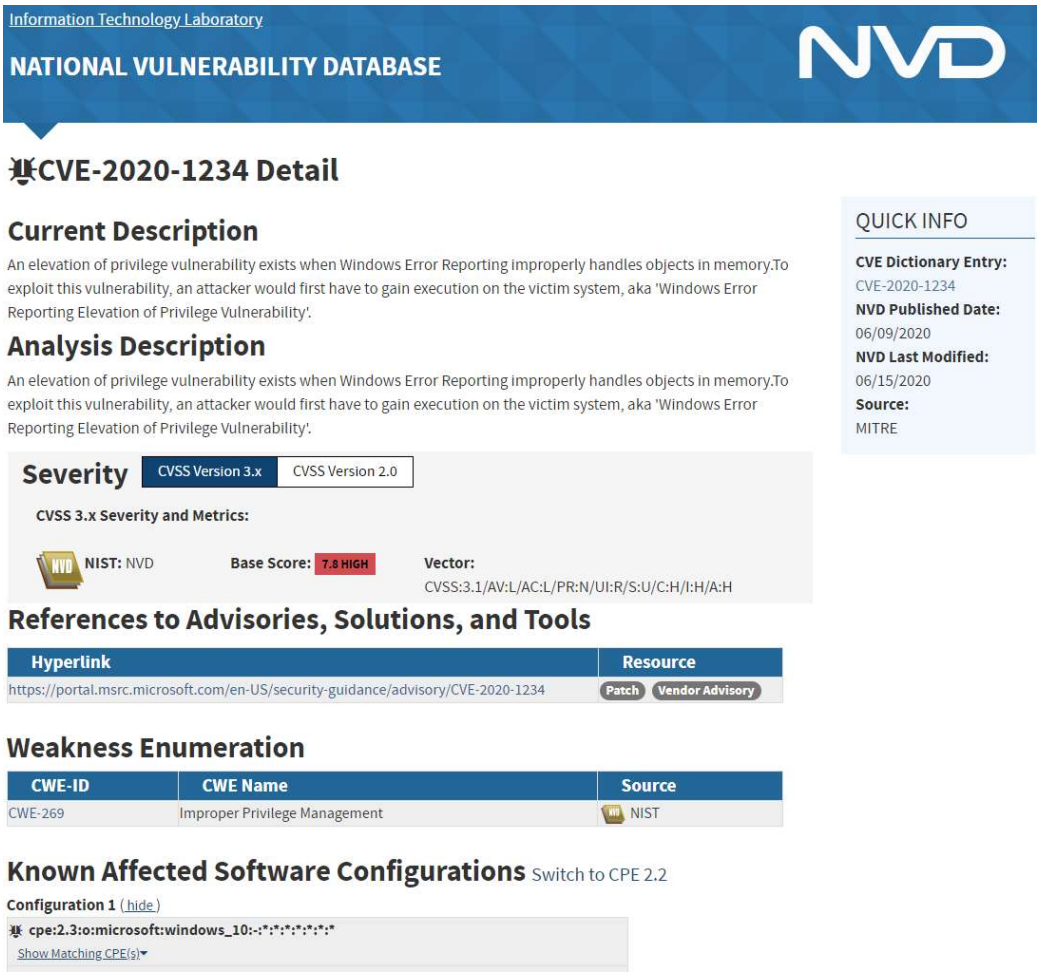


Figure 3: Example of a CVE format in the NVD

The "National Vulnerability Database" (https://nvd.nist.gov/) contains all published CVE's starting from its inception in 1999.

CVE's are the fundament under the vulnerability administration. The CVE system is used by government, companies, researchers and institutions.

Date	Title	CVE	Description	Products and versions affected	More information
2021/12/14	IGSS (Interactive Graphical SCADA System)	CVE-2021-22823, CVE-2021-22824	Multiple Vulnerabilties	IGSS Data Collector (dc.exe) V15.0.0.21320 and prior	SEVD-2021-348-01

Figure 4: Two vulnerabilities in the same Schneider Electric product, hence two CVE's are assigned.

Completeness
There are more vulnerabilities than administered in the CVE catalogue. Estimates are that the real number of vulnerabilities is five? times higher. Most of the unknown vulnerabilities are only known to hackers, sold on the dark web or sold to companies or foreign governments[5].

Another reason for some vulnerabilities having no CVE-ID is that the company whose products are mentioned did not bother to have a CVE-ID assigned[6]. The vulnerability is then only known under a company-specific designation.

Chinese researchers and companies may not publish about vulnerabilities without first having informed their government first (https://www.scmp.com/tech/big-tech/article/3160670/apache-log4j-bug-chinas-industry-ministry-pulls-support-alibaba-cloud). For stately purposes, it is expected that the existence of some vulnerabilities must be kept secret.

What is the "CVSS" ?

Not every vulnerability is the same. Some can be exploited from the local network only, others also from internet. Some are easy, others require much knowledge from the hacker. Some can be exploited only with help from the (unsuspecting) user, other do not require any action by a user.

To judge how dangerous a vulnerability is, the "CVSS – Common Vulnerability Scoring System" has been set up. Every vulnerability can be characterized, and be given a score in the range 0..10 reflecting its severity (0 = harmless, 10 = very dangerous).

The CVSS system has been developed by FIRST in cooperation with experts from the cybersecurity community It has grown over the years, currently CVSS 3.1 is in use. The website https://www.first.org/cvss/ gives all necessary background information.

On https://www.first.org/cvss/calculator/3.1, a CVSS score can be calculated for a vulnerability, by filling in the "base metrics" parameters. There are 8 of them:

- Attack Vector (AV)
- Attack Complexity (AC)
- Privileges Required (PR)

[5] As extensively described in Nicole Perlroth's prize-winning book "This is where the world ends – the cyberweapons's arms race".
[6] It is not that companies may be completely unaware of the CVE system, as they sometimes do use the CVE system for other vulnerabilities, but strangely not for all.

- User Interaction (UI)
- Scope (S)
- Confidentiality (C)
- Integrity (I)
- Availability (A)

Each base metric has several possible values, for example for "Privileges Required":

- None: the attacker needs no authorization to carry out an attack
- Low: the attacker must have been authorized with basic capabilities to carry out an attack
- High: the attacker must have been authorized for significant control to carry out an attack

Which value is chosen depends on the vulnerability itself. Usually the discoverer of a vulnerability (often the vendor of a device, but also security researchers) has the knowledge to choose the appropriate value for each metric.

Each base metric has a certain weighing factor[7]. The final outcome is a value in the range 0.0 .. 10.0, where 10 = very dangerous. There are 5 severity levels:

- None: 0.0
- Low: 0.1 – 3.9
- Medium: 4.0 – 6.9
- High: 7.0 – 8.9
- Critical: 9.0 – 10.0

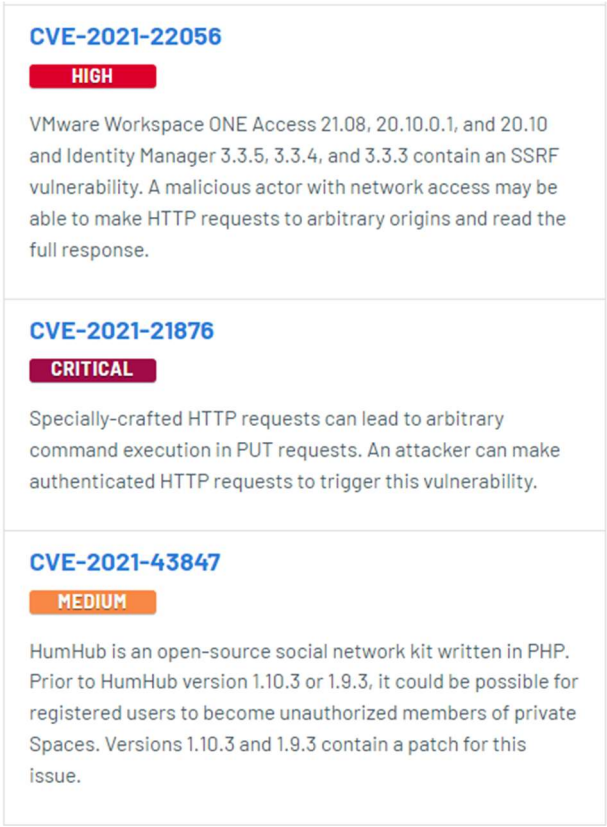


Figure 5: Severity levels usage on a website (source: Tenable)

[7] The algorithm is too complicated to explain here, but a full explanation is available on https://www.first.org/cvss/v3.1/specification-document

A website with a calculator is available on https://www.first.org/cvss/calculator/3.1. By filling in a choice for each base metric, the CVSS score (top right) appears.

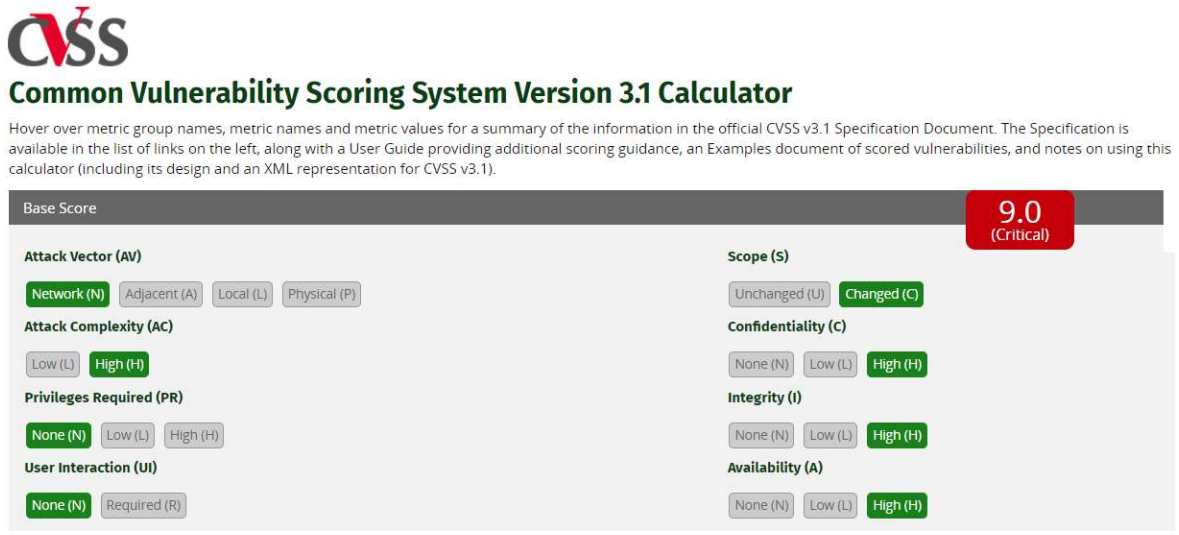


Figure 6: The CVSS calculator

Notice how the CVSS score changes when you modify a base metric's value. This may sometimes look strange, why does the score increase in value when you change (for example) the User Interaction value from "Required" to "None" ? This is because the score is calculated as seen from the viewpoint of an attacker: a vulnerability which requires the user to (unwillingly) assist is less dangerous from a vulnerability where the user is not needed.

Usually, the base metrics used in the calculation of the score are provided in shorthand notation. This is called the "CVSS Vector". For example, the CVSS vector as show in figure 6 would be: "CVSS3.1/AV:N/AC:H/PR:N/UI:N/S:C/C:H/I:H/A:H". You recognize the 8 base metric parameters between / characters, and then the value for each of them following the colon.

CVSS scores are always listed in the CVE catalogue. An example is shown in figure 7. For backwards compatibility, it usually also shows the CVSS version 2 score. It is advised not to use CVSS2 unless strictly necessary, for example when no CVSS version 3 score is provided (usually with CVE's dating before 2016).

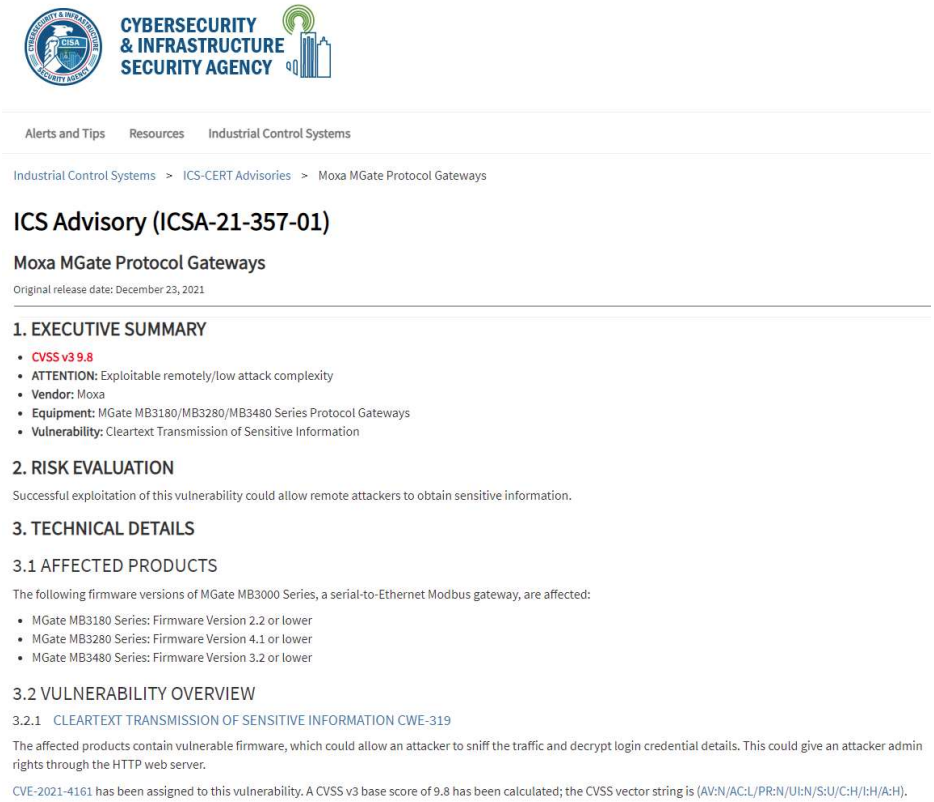
Figure 7: The CVSS3 base score for a vulnerability, and the vector (bottom right).

Industrial vulnerabilities

ICS-CERT catalogue

The CVE catalogue lists vulnerabilities in all sorts of products, consumer, office and industrial. Most of the consumer-product related vulnerabilities are not applicable for industrial applications. For the latter, a special catalogue is available called "ICS-CERT" (www.cisa.gov/uscert/ics/advisories).

It only lists industrial products vulnerabilities, a subset of all CVE's. For many working with industrial systems ICS-CERT is a major source of information.


CYBERSECURITY & INFRASTRUCTURE SECURITY AGENCY

Alerts and Tips Resources Industrial Control Systems

Industrial Control Systems > ICS-CERT Advisories > Moxa MGate Protocol Gateways

ICS Advisory (ICSA-21-357-01)

Moxa MGate Protocol Gateways

Original release date: December 23, 2021

1. EXECUTIVE SUMMARY

- CVSS v3 9.8
- ATTENTION: Exploitable remotely/low attack complexity
- Vendor: Moxa
- Equipment: MGate MB3180/MB3280/MB3480 Series Protocol Gateways
- Vulnerability: Cleartext Transmission of Sensitive Information

2. RISK EVALUATION

Successful exploitation of this vulnerability could allow remote attackers to obtain sensitive information.

3. TECHNICAL DETAILS

3.1 AFFECTED PRODUCTS

The following firmware versions of MGate MB3000 Series, a serial-to-Ethernet Modbus gateway, are affected:

- MGate MB3180 Series: Firmware Version 2.2 or lower
- MGate MB3280 Series: Firmware Version 4.1 or lower
- MGate MB3480 Series: Firmware Version 3.2 or lower

3.2 VULNERABILITY OVERVIEW

3.2.1 CLEARTEXT TRANSMISSION OF SENSITIVE INFORMATION CWE-319

The affected products contain vulnerable firmware, which could allow an attacker to sniff the traffic and decrypt login credential details. This could give an attacker admin rights through the HTTP web server.

CVE-2021-4161 has been assigned to this vulnerability. A CVSS v3 base score of 9.8 has been calculated; the CVSS vector string is (AV:N/AC:L/PR:N/UI:N/S:U/C:H/I:H/A:H).

Figure 8: Example of an ICS-CERT advisory

Each ICS-CERT advisory has the same format (see figure 8). The ID used is "ICSA-<year>-<day of year>-<sequencenumber>", so advisory "ICSA-21-357-02" was the second advisory issued on the 357th day of 2021. When it is necessary to update the advisory with new information, a letter is added (i.e. ICSA-21-357-01A, -01B, etc.). An example of an advisory[8] is shown above.

In the advisory, note the reference to any applicable CVE('s), the CVSS score, and the vendor advisory (if any). In principle, the information in an ICS-CERT advisory is the same as to be found in a CVE (with a few exceptions).

[8] ICS-CERT also publishes advisories on vulnerabilities in medical products. The ID uses is slightly different, it starts with "ICSMA".

Disadvantages of ICS CERT
A disadvantage of ICS CERT is that not all companies whose products may be found in an industrial environment are listed. For example, the products of Cisco are not listed on ICS-CERT, even though Cisco is often to be found in industrial systems.

Also, not all industrial vendors appear in the ICS-CERT catalogue. Even vendors who report to ICS-CERT may not always report all their vulnerabilities, or only with months delay. According to a report by Kaspersky[9], 23% of industrial control systems vulnerabilities are not in the ICS-CERT catalogue.

ICS-CERT advisories are not machine-readable, i.e. the information in advisories cannot be automatically processed and ingested by other software, for example intrusion detection systems or asset management systems.

VDE-CERT

A counterpart of ICS-CERT is the German "VDE-CERT", set up by the VDE[10]. Many small and large German industrial companies report their cybersecurity advisories to the VDE. Despite VDE being German, the advisories are published in English only (https://cert.vde.com/en/).

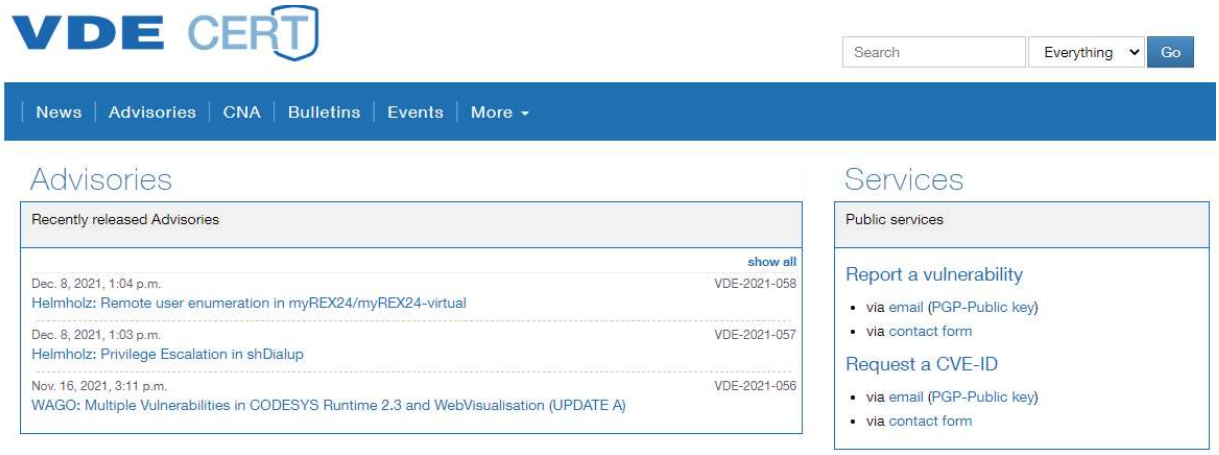


Figure 9: Excerpt from the VDE-CERT website

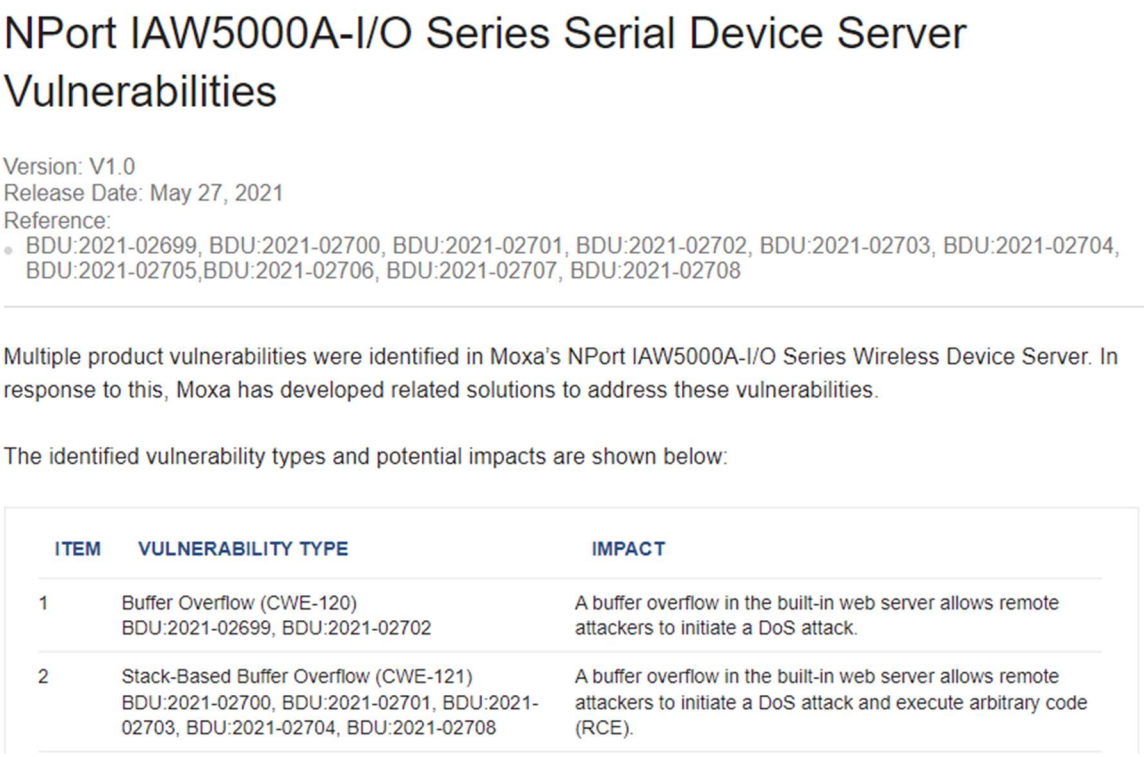
Noticeably absent from VDE CERT is Siemens. VDE CERT is not only for German companies, but is available to support any European vendor. VDE CERT may pass on their advisories to ICS-CERT.

FSTEC

This is the Russian equivalent of the (American) NVD. The "Federal Service for Technical and Export Control" has its own catalogue of vulnerabilities. In most cases these are 1:1 copies of those in the NVD, but not all. For example, those vulnerabilities found (and reported) by Russian researchers or companies.

[9] See https://ics.kaspersky.com/media/ics-conference-2019/04-Artem-Zinenko-Nedostatki-publichnyh-baz-uyazvimostey.pdf. A weakness of such statistics is that it is unclear what an "industrial control system" is.
[10] Verband der Elektrotechnik Elektronik und Informationstechnik = Association for Electrical, Electronic & Information Technologies.

Vulnerabilities have an ID in the format "BDU:<year>-<number>", for example "BDU:2021-06423". It is not the same designation as the CVE ID; for example the aforementioned BDU ID has CVE-2021-37996".



NPort IAW5000A-I/O Series Serial Device Server Vulnerabilities

Version: V1.0
Release Date: May 27, 2021
Reference:
- BDU:2021-02699, BDU:2021-02700, BDU:2021-02701, BDU:2021-02702, BDU:2021-02703, BDU:2021-02704, BDU:2021-02705,BDU:2021-02706, BDU:2021-02707, BDU:2021-02708

Multiple product vulnerabilities were identified in Moxa's NPort IAW5000A-I/O Series Wireless Device Server. In response to this, Moxa has developed related solutions to address these vulnerabilities.

The identified vulnerability types and potential impacts are shown below:

ITEM	VULNERABILITY TYPE	IMPACT
1	Buffer Overflow (CWE-120) BDU:2021-02699, BDU:2021-02702	A buffer overflow in the built-in web server allows remote attackers to initiate a DoS attack.
2	Stack-Based Buffer Overflow (CWE-121) BDU:2021-02700, BDU:2021-02701, BDU:2021-02703, BDU:2021-02704, BDU:2021-02708	A buffer overflow in the built-in web server allows remote attackers to initiate a DoS attack and execute arbitrary code (RCE).

Figure 10: Vulnerabilities reported by Moxa, as found by Russian researchers, with a BDU ID and not a CVE ID.

Although the website (https://bdu.fstec.ru/vul) is in Russian, a huge part can be translated automatically by Google Translate pretty well. Although FSTEC won't be a major source of information about industrial vulnerabilities because it has only some 10% of the number of advisories in the NVD, sometimes it has its use.

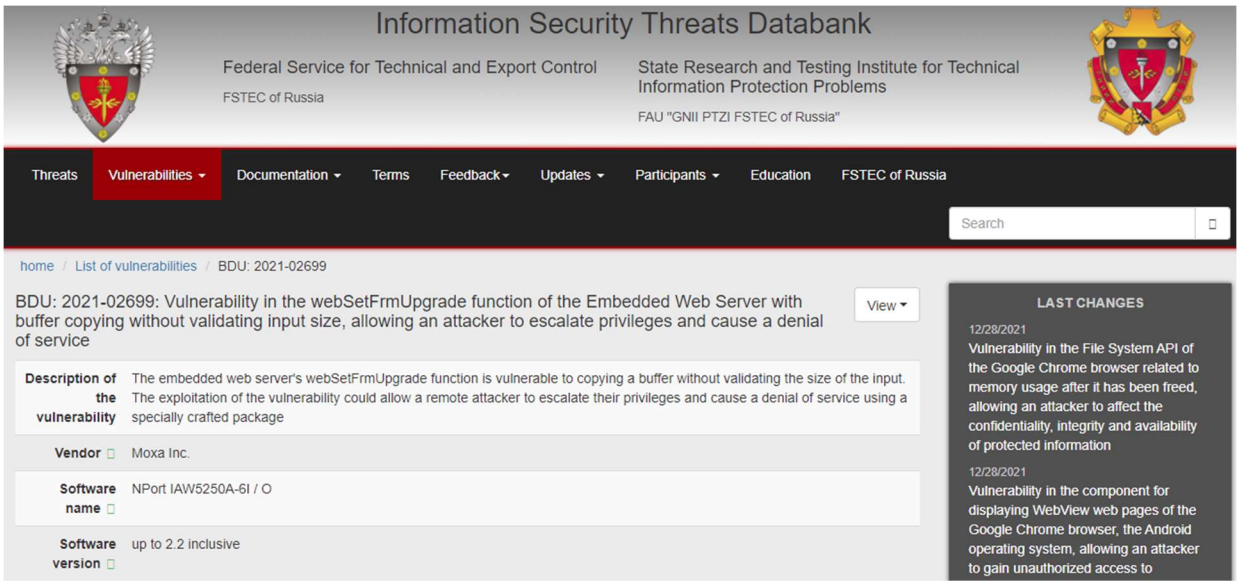


Figure 11: a page from the FSTEC website (after translation by Google).

Vendor advisories

Primary sources of information about industrial vulnerabilities are the vendors of the products. They have the best knowledge of their own products: hardware, software, and the product knowledge. A vulnerability usually (but not always) has a CVE ID assigned, and likely also an ICS-CERT ID. Upon reading the vendor advisory, the CVE and ICS-CERT advisory you may see the same information reported in three different formats.

Each vendor has his own way of publication, some publish on their website, others only to registered customers, and many vendors publish nothing.

Also, there is no standard format used in advisories. Some refer to a CVE, others not. Some list a CVSS score, others not. This makes automatic processing of vendor advisories difficult[11] (especially when they are provided in PDF format).

Researcher reports

Sometimes, researchers find a vulnerability in an industrial product, and they publish about it. When they inform the vendor (responsible disclosure), the vendor can then create a patch for the vulnerability and publish this.

But upon reading the vendor advisory, sometimes one finds much more products listed as being vulnerable than mentioned in the researcher's publication. This is because the researcher probably had only one device to test, but the vendor found out that the vulnerability affected a whole product family, likely also with multiple software-versions.

Multiple locations

Many vendors have a single place (webpage) where they publish all their advisories. But some larger companies do not do this. Some examples:

- Siemens has a single webpage for advisories for most of their products, but the medical product division (Siemens Healthineers) has its own webpage.
- General Electric has separate webpages for their PLC's, medical, power industry and grid products.

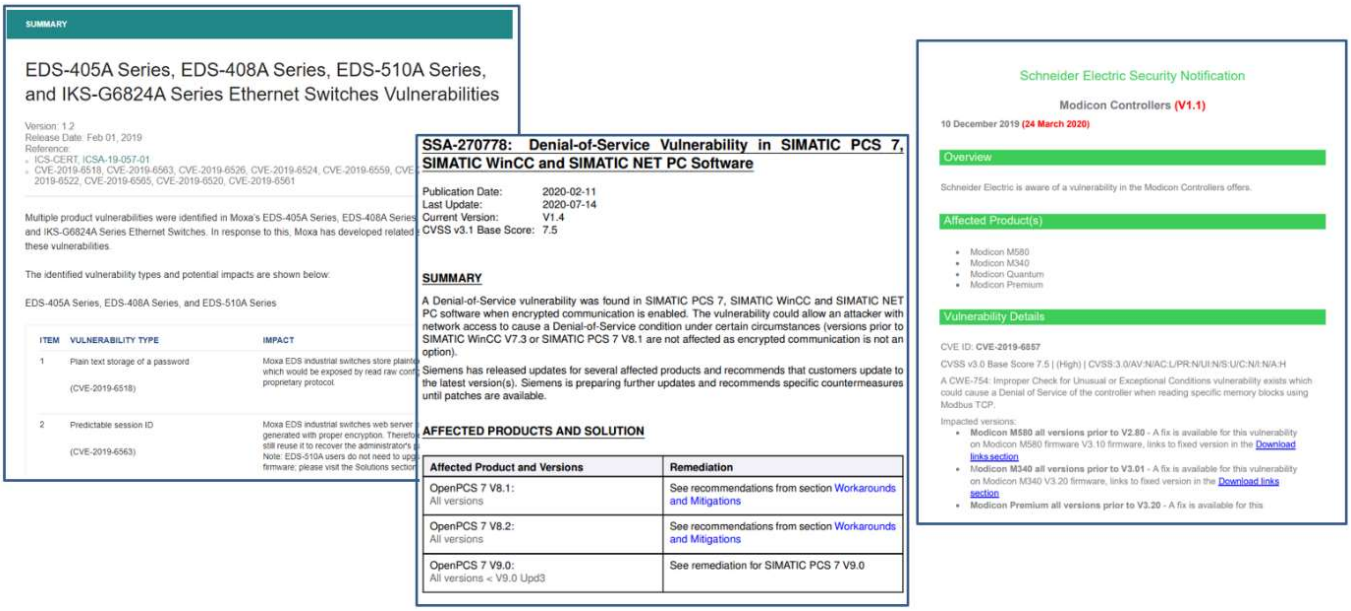


Figure 12: Three vendor advisories (Moxa, Siemens, Schneiders)

[11] An exception to this is Cisco, which has many advisories available in "CVRF" format (Common Vulnerability Reporting Format). Unfortunately, the rest of the industry hasn't adapted this format.

Social media

The last source of information about vulnerabilities is: social media. A lot of information is to be found on LinkedIn and Twitter, especially about new vulnerabilities with high impact.

More ?

The next part, Patching 102” will describe how to set up a patch process in your company.

This article is likely not complete in describing all possible patch-related knowledge. If you have any suggestions, comments or additions, please do not hesitate to contact me (email: rh[at]enodenetworks.com).