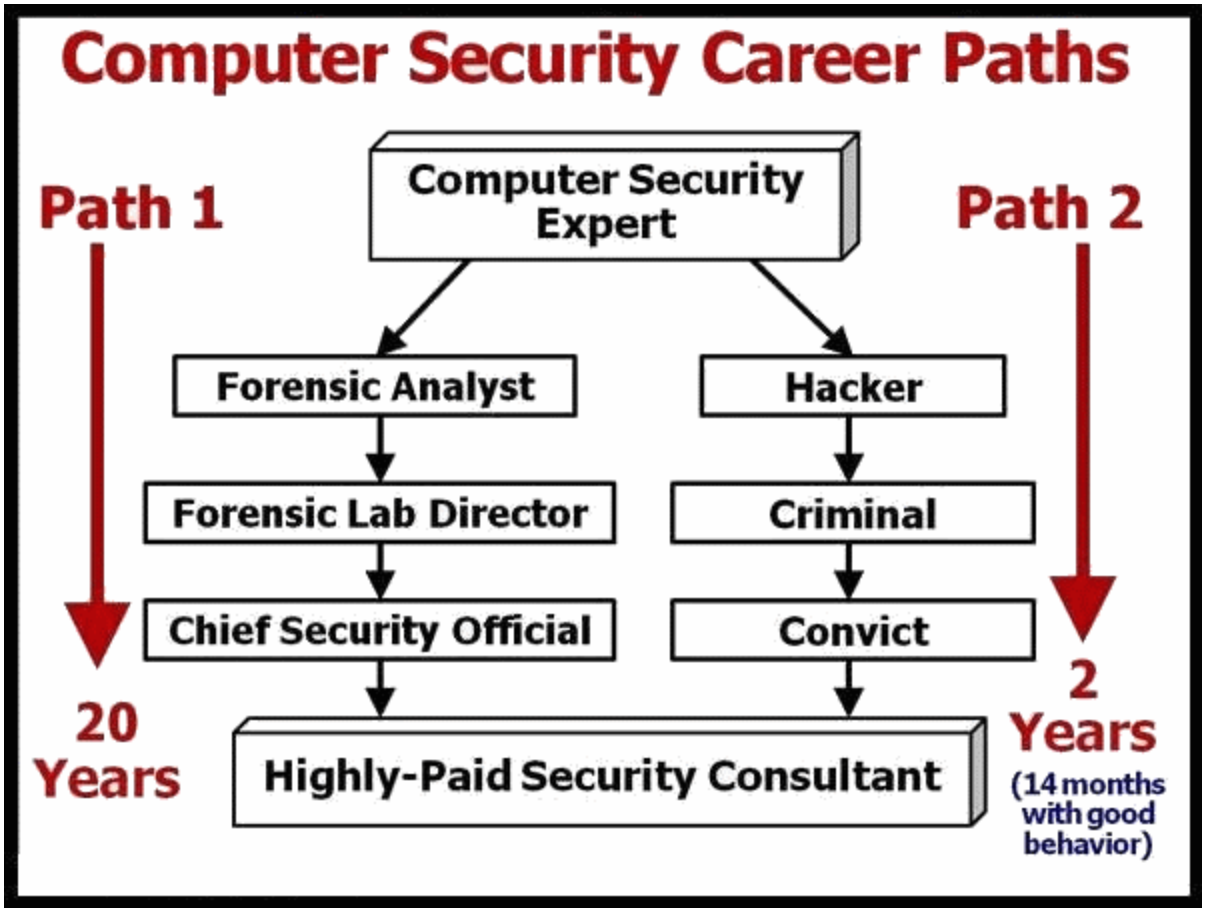
# Security Now! #881 - 07-26-22

## The MV720

### This week on Security Now!

This week we start off by updating our follow-up to this month's patch Tuesday. Things were more interesting than they originally seemed. Then we keep up with the evolving state of Microsoft Office's VBA macro foreign document execution. We also have a fabulous bit of news about some default security policy changes for Windows 11 announced by Microsoft. Then, with August rapidly approaching, we have a few calendar notes to mention, I have a welcome and long-awaited bit of SpinRite news to share, we have a bit of miscellany and some brief bits of listener feedback to cover. Then we take a deep dive into the poor-by-design security of a very popular and frightening widely used aftermarket GPS tracking device. You don't want one of these anywhere near you or your enterprise. Yet 1.5 million are.

### LOVE this...



# Security News

**Patch Tuesday Redux Redux**

Last week, which was the week following July's Patch Tuesday I congratulated Microsoft for their having patched some 84 known flaws without simultaneously crippling Windows. But since then it has come to light that I may have been somewhat premature in my praise.

Published under "Issue details for July 2022" is the topic: "Printing to USB-connected printers might fail" with a status of "Confirmed". The affected (or afflicted) platforms include both client (Windows 10, version 20H2, 21H1, and 21H2) and server version 20H2. It reads:

> *Microsoft has received reports of issues affecting some printing devices following installation of Windows updates released June 28 ( KB5014666) and later. Symptoms observed may include:*
>
> *Windows might show duplicate copies of printers installed on a device, commonly with a similar name and the suffix "Copy1". Applications that refer to the printer by a specific name cannot print. Normal printer usage might be interrupted for either scenario, resulting in failure of printing operations.*

https://docs.microsoft.com/en-us/windows/release-health/status-windows-10-21h2#printing-to-usb-connected-printers-might-fail

Windows printing, like Windows LAN Manager networking, has pretty much always been a mess. As we know, last year's Windows' Printer Spooling security debacle dogged Microsoft for more than half a year. Some of Windows' architecture has not aged well through the decades. And it's understandably difficult to ever make the decision to scrap something that mostly works, in favor of a major redesign which is certain to break a large number of things that are currently working well... especially when there is so much hidden dependency upon the existing system.
It really is the case that at this point Microsoft can barely change anything without breaking everything. I think that future history will show that they had painted themselves into a corner from which there was no escape. We keep seeing that Microsoft's customers just want them to leave everything alone, working as it now does. It could not be any more clear that Windows is not actually getting any better. It's now clearly getting worse. But that doesn't work with Microsoft's need to appear to always be moving forward... even though no one wants them to.

Anyway, if by some chance your printing to USB stopped working earlier this month and you haven't yet decided to tackle it, the trouble appears to surround the spontaneous creation of a duplicate printer instance where **it** somehow obtains the proper configuration while upsetting the configuration of the original instance. In their "workarounds" explanation, Microsoft writes:

> *Open the Settings app, navigate to "Bluetooth & devices", and select "Printers & scanners"*
>
> *If there appears to be a duplicate installation of an existing printer, such as with suffix "Copy1", confirm if printing works for this printer. This printer should operate as expected.*
>
> *If there is a need to use the original printer installation and not the duplicate [the one which*

*now works], right-click the duplicate printer, select "Printer properties" and select the "Ports" tab. Observe the port in use.*

*Open "Printer properties" on the original printer and select the "Ports" tab. From the list displayed, select the port option in use by the duplicate printer. If this original printer [then] works normally, the duplicate printer copy can be removed.*

Reading between the lines of this workaround, it sounds as though whatever-it-was Microsoft was attempting to do, intended to create a new instance of a USB printer, copy the original instance's settings into the new instance. Then remove the original instance and give the new instance the name of the original. It sounds like for some users, that process got part of the way along and then died, and did not back itself out and revert to the original configuration which was working before all of this began.

**Windows 11 Start button failure**

That USB printer problem affected Windows 10. But Windows 11 did not escape unscathed. It can be disconcerting when Windows' Start button no longer works. Microsoft wrote:

*After installing KB5014668 or later updates, we have received reports that a small number of devices might be unable to open the Start menu. On affected devices, clicking or selecting the Start button, or using the Windows key on your keyboard might have no effect.*

Fortunately, Microsoft has the answer. Updates have become so unpredictably faulty that Microsoft created the "KIR" facility. If you can't fix the updates, at least you can get rid of them. KIR stands for Known Issue Rollback. It provides Windows with a sometimes needed capability to revert buggy fixes which have been delivered through Windows Update. After a rollback is pushed via KIR, all consumer and non-managed business devices will likely receive the fix within a day. The trouble is, it's difficult to get anything done without any Start button or even a keyboard trigger.

**The continuing saga of Windows VBA macros**

As yet another example of one of Microsoft's very poor early design decisions not aging well, and their refusal for many years not to simply do the right thing, we have the continuing saga and drama of Windows VBA macros.

Last Wednesday night, Microsoft confirmed that it is resuming the roll out of their plan which they first announced earlier this year, in February. That announcement back then was greeted with great relief by everyone who understood what it would mean for the security of Microsoft's much-abused Office documents: After years of head-in-the-sand policy, Microsoft would be blocking the execution of remotely-received VBA macros by default across most Office apps.

Predictably, this would also break some things. Which, of course, explains Microsoft's reticence to do the right thing sooner. We've never really talked about the pushback against this change, but I came across some interesting bits which address that. Even though Microsoft declined to

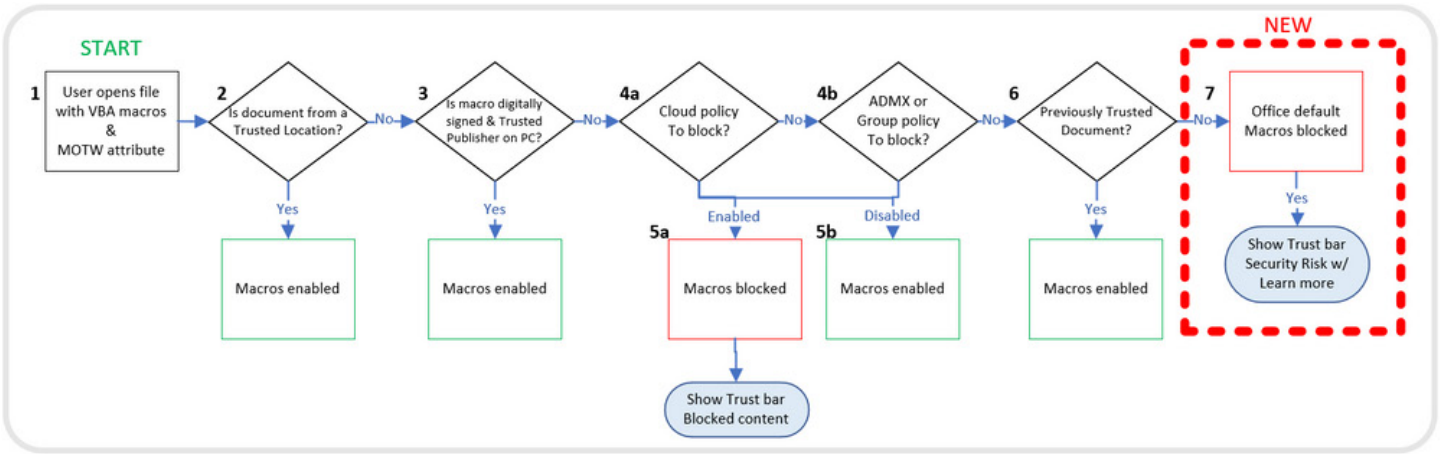
provide information about why the effort had been paused, several experts said customers complained about the new feature.

Michael Tal, technical director of Votiro, a company which specializes in malicious content filtering in the cloud, told The Record that he works closely with partners in the banking and financial sectors and explained that macros play *"an integral part of our client's business workflows."* He said that the initial block caused a *"massive hindrance on business productivity."* Yeah, right, something changed, just as Microsoft warned everyone it was going to back in February when they said to get prepared. It turns out, Microsoft wasn't fooling.

Micheal Tal explained: *"Macros are a powerful tool in the financial sector, as they are used to create robust financial modeling, calculate loan interest, automate repetitive, labor-intensive tasks – they are recorded sets of actions which can be run to save time and labor. It is also used to simplify budget forecasting and makes a difference in a day-to-day workload of any entity who's using it as it speeds up the process to generate a task after finalizing the creation of the macro and setting the variables."*

Tal added that while he understood Microsoft's desire to combat malware like Emotet, Trickbot, Qbot and Dridex, they should have come up with a more creative approach to deal with legitimate business use cases for macros and allow for continuity without compromising security.

What? To all of this whining I say "Oh, boo hoo!" My god, it's not as if macros have been stripped out of the Office tools and are gone. They simply no longer run without provocation. You just need to click a button to explicitly permit their use, and even that can be done, as we've seen in that crazy flowchart, through enterprise-wide group policy settings:




Decision box 4a in the flowchart is "Cloud policy to block?" set it to "no" then decision box 4b "ADMX or Group policy to block?" can be used to enable macros at that point... so behavior doesn't change. You don't get the enhanced protection from foreign content, but neither are your delicate banking and financial users upset by the need to press an extra button.

If nothing else, the fact that this change DID create such a kerfuffle serves to conclusively demonstrate just how necessary it was, and is, to disable macros from running from external documents by default!

How can any moron think that it's a good idea to allow macros to run, unbidden, in a document received through eMail. Sure, legitimate documents do that. So simply sign them and they still will. Those macros which the banking and financial sectors are so upset about not running all by themselves could just as easily be malicious and could be the way ransomware slips into their networks. They can't have it both ways.

The way they have been historically operating has been insecure. Microsoft should have simply ignored them and made them tighten up their security. We have seen over and over how difficult it is to make these changes. They have to be forced.

This is such an obviously good thing for security, that it shouldn't have been any surprise that the security industry was upset when, two weeks ago, Microsoft announced their decision to "temporarily" roll back this welcome and so-long overdue change. Then finally, last Wednesday evening, Microsoft updated their webpage about the feature, which applies to Access, Excel, PowerPoint, Visio and Word, saying:

*"We're resuming the rollout of this change in Current Channel. Based on our review of customer feedback, we've made updates to both our end user and our IT admin documentation to make clearer what options you have for different scenarios."*

Hallelujah!!

And as if Microsoft needed any additional reason or evidence, the "Follina" vulnerability from May served as another perfect example of how Office macros gave threat actors a way to exploit user files.

**Windows 11 now blocks RDP brute-force attacks by default**

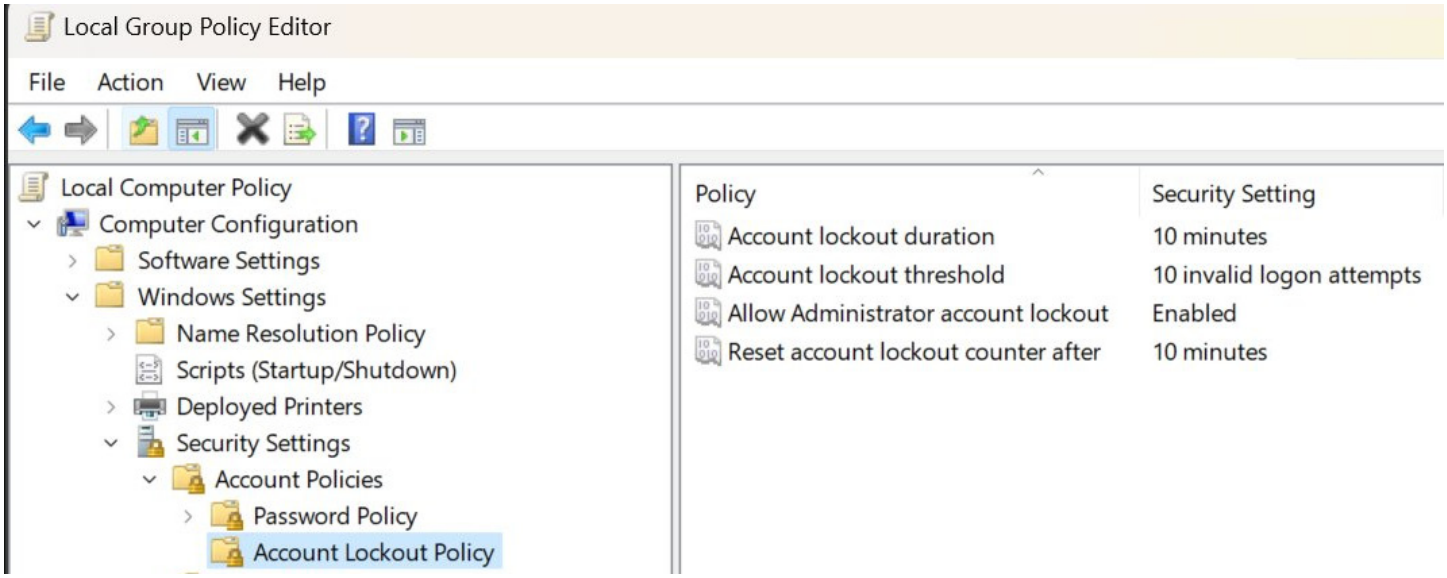
And speaking of "Hallelujah!!" ... last Thursday, Microsoft's VP for Enterprise and OS Security, David Weston, tweeted:

> ***@windowsinsider*** *Win11 builds now have a DEFAULT account lockout policy to mitigate RDP and other brute force password vectors. This technique is very commonly used in Human Operated Ransomware and other attacks - this control will make brute forcing much harder which is awesome!*

Everyone listening to this podcast is acutely aware of the importance of default settings. There may as well not even be any settings since the default is almost universally what it left to apply.

The other thing everyone listening to this podcast knows is that the inherent insecurity created by Microsoft's remote desktop protocol being placed out onto the public Internet without any sort of brute force credential-stuffing protection in place — by default — has been responsible for untold numbers of remote network intrusion, pain and loss for their customers.

The FBI said RDP is responsible for roughly 70% to 80% of all network breaches leading to ransomware attacks. **RDP.** So, David's news was incredibly welcome.  But, my jubilation was somewhat tempered when I saw the Local Group Policy Editor settings that he was announcing...




Windows 11 now has its failed login attempts account lockout triggering after 10 invalid logon attempts. But the lockout duration is only 10 minutes and the failed attempts counter also resets every 10 minutes. Could someone explain to me how any **legitimate** user of remote desktop protocol, whose RDP client has probably memorized and stored their logon authentication anyway — making it automatic for the user — is going to be inconvenienced by a lockout which doesn't engage until they have somehow failed to properly authenticate themselves 10 times?

Since, as we've seen, Microsoft clearly goes to great lengths to never inconvenience any user in the name of security, they must believe, as I do, that no one but an attacker would never trip the "10 strikes before you're out" rule. So why, then, give an attacker a clean slate 10 minutes after those 10 failed attempts? Let's hope and pray that the error message returned by the RDP endpoint is the same for a failed logon attempt as for a block by policy.

In any event, I get it. Baby steps. At least we have the concept now in place of default mitigation against brute forcing unmonitored authentication endpoints. That's progress.

## Calendar Notes

**This is our final podcast of July as we head into August.** And August traditionally contains a few interesting security events: We have the two traditional major hacking conferences, Black Hat and DefCon.

This year, running from August 6th through the 11th, the 25th Black Hat USA will be holding a hybrid conference allowing the cybersecurity community to choose how they wish to participate. The conference will open with four days of Virtual Trainings conducted in real-time online, with all instructors accessible throughout each class. Then the final two-day main conference will be held both virtually online and live in-person in Las Vegas at the Mandalay Bay Hotel and Casino.

Then, as always immediately following Black Hat, is DEFCON, August 11th through the 14th. I

had to smile and shake my head as I was checking-in on DEFCON's description of itself and its proceedings. Since I think everyone will get a kick out of what they wrote, I'll share what they wrote:

> Started in 1992 by the Dark Tangent, DEFCON is the world's longest running and largest underground hacking conference.
>
> Hackers, corporate IT professionals, and three letter government agencies all converge on Las Vegas every summer to absorb cutting edge hacking research from the most brilliant minds in the world and test their skills in contests of hacking might.
>
> DefCon comes right after Black Hat, a conference and trade show for cybersecurity professionals. While Black Hat feels more like a traditional Vegas trade show, DEFCON is anything but!
>
> How is it different from other conventions?
>
> Well first, DEFCON is run by volunteers and has no corporate sponsorship.  Second, there is no online registration, so even the organizers really don't know who is attending.
>
> When you arrive, everything is paid for with CASH! They don't take Credit Cards — most of these people attending really don't want a record of them attending. Everyone from your average everyday hacker to criminals and agents from government agencies like the FBI, CIA and National Security Agency will attend.
>
> When you enter, you pay $280 cash and they hand you a generic badge — NO ID is required for admittance.

**Security Now's 17th birthday...**
And the final event of note, which occurs every August 19th, is the anniversary of the first episode of this podcast. This coming August 19th finishes out our 17th year and we'll begin into year 18.

## SpinRite

Last Friday afternoon I posted to the grc.spinrite.dev newsgroup under the subject "It's Alive!" As we know, I essentially had to take SpinRite completely offline and down to perform the degree of surgery that was needed, not only to completely strip out all of SpinRite's traditional dependence upon the BIOS, but to also, as I've explained during this journey, to completely re-architect SpinRite around a data-recovery-centric device-independent mass storage device abstraction so that not only can SATA and IDE drives connected with AHCI and PCI BusMastering adapters now be communicated with at their lowest possible hardware levels, but also so that the next step in SpinRite's evolution, which will add similar direct access for USB and NVMe devices, and whatever else might show up in the future, will be able to have support plugged-in without needing any similar reworking. I've done all of that work upfront.

I'm mentioning this because I can finally report that SpinRite is beginning to come back to life. Humpty Dumpty is getting its pieces reassembled. It's starting to run again. But it is by no means ready yet. I don't want to give anyone that impression. I still have lots of work left to do because the surgery SpinRite needed broke virtually every assumption that it was originally built upon — assumptions which were made back in 1987 when we had 4.77 Megahertz Intel 8088 processors with a maximum of 640 Kbytes of RAM and a 20 megabyte hard drive was a luxury.

```
;+-------------------------------------------------------------------------+
;|   file: sr.asm            by: Steven M. Gibson         created: 03/30/87    |
;+-------------------------------------------------------------------------+
```

Essentially, everything has changed since then, yet those assumptions had been allowed to remain in place, even through SpinRite 6, though they were becoming quite old and creaky. So as everyone knows, 6.1 is not a patch to SpinRite 6. Even though it's a minor version bump and therefore a free upgrade for everyone who owns version 6, I'm making this multi-year investment in SpinRite's future today because I've seen the future and to my utter amazement, SpinRite is still in it. Mostly, though, I am so looking forward to writing that code. v6.1 will be the new foundation for that.

## Miscellany

**pfSense and TailScale**

Everyone knows that pfSense is my preferred Internet firewall router solution. It's open source and has a fully capable free community supported release. It's rock solid, runs on most hardware — inducing little fanless consumer routers such as my favorite little NetGate SG-1100. And it has a comfortable web UI.

Among pfSense's many features is a modular package management system which makes managing the router a "manpage-free" pleasure. It's just point and click. So the news I wanted to share is that pfSense will soon (with its forthcoming v22.05) be receiving built-in drop-menu selectable support for the TailScale VPN mesh overlay network. This is terrific, since TailScale, which uses WireGuard as it's VPN transport layer, offers a startlingly simple configuration and operation experience. It offers automatic key rotation, NAT traversal, and single sign-on with two-factor authentication.

Before long, bringing up TailScale on pfSense endpoints will be as easy as using it once it's there.

# Closing The Loop

**"Dangerously Close to Hijinks" / @christyramsey**

*Thanks for all you do. Would you share the software solution you use for grc.sc? Do you recommend it?*

*Thanks to you and Elaine I found the reference for URL shortener YOURLS in #858*

https://yourls.org/ YOURLS : **Y**our **O**wn **UR**L **L**ink **S**hortener

Requirements:

- PHP 7.4 or above
- At least MYSQL 5
- A web server with mod_rewrite enabled
- Note: YOURLS can also run on Nginx, Cherokee and more (and on IIS)
- HTTPS support
- PHP CURL extension installed if you plan on playing with the API

**biswb / @biswbmatt**

*In regards to episode 880: "IPV6 is the technology of the future... and it always will be."*

# The MV720

The MV720 is a tiny cube measuring about an inch by an inch by an inch. And if someone were to tuck you your car's hood without your knowledge, plugging it into your car's wiring harness, you would be hard pressed to know that anything was out of place. In fact, if your car's authorized service people were working under the hood, they, too, would likely pass it off as just some "supposed to be there" relay. And that would be by design, since the manufacturers of this sneaky little cellular radio equipped GPS satellite monitoring and vehicle control overriding demon boast at the top of its web page, below the title "Easy to Hide" that "MV720 looks like a relay bit is actually a locator:


The question is, who put it there, and why? Since this thing only costs $20, it could be anyone who has reason, legal or otherwise, to want to monitor and track the vehicle's location and speed while having the option to remotely shut down the motor's flow of oil, causing the vehicle to gradually slow down to a point where the engine can be shut off and disabled... all remotely.

So, again, who put it there, and why?

These things exist. And while I'd be happy to be talking about them if only for the sake of noting their existence, they wouldn't normally rise to the level of being a headline topic of this podcast. So our long time listeners can probably see where this is likely headed.

What do we know about this thing's manufacturer? MiCODUS is a Shenzhen, China-based manufacturer and supplier of automotive electronics and accessories. The company's main products are asset, personal, and vehicle GPS trackers. MiCODUS devices are available for purchase via Amazon, Aliexpress, Ebay, Alibaba, and other major online retailers. (And, sure

enough, I found one on Amazon for $26.) In addition to GPS devices, the company provides a cloud-based platform (web, iOS, and Android) for remote management, fleet and asset tracking, and vertical-specific applications. MiCODUS states it provides a "secure, open and scalable platform that plays an essential role in the optimization of resource utilization by enabling visibility and simplifying management."

The security vulnerability research firm, BitSight took a close look at this little device's security. BitSight chose the MiCOCUS's MV720, because it's the company's least expensive model with fuel cut-off capability. As we'll see in a minute, it's a cellular-enabled GPS tracker which uses a SIM card to transmit status and location updates to supporting servers and to receive SMS commands from its user. (And, unfortunately, also from pretty much anyone.)

And I'm sure no one is going to be surprised by what they found. They found six vulnerabilities of severity up to CVSS 9.8. If there were only a couple of these little one inch cubes wandering around the planet somewhere, hopefully in China, that would not represent a clear and present danger. But one and a half **million** of these little demons are currently present in vehicles located throughout 169 countries. They're present in the vehicles used by several Fortune 50 firms in the U.S., by European governments and by state government agencies in the U.S. A South American military agency uses them, as does a nuclear plant operator. Given the tracking power and the ability to remotely cut off a vehicle's fuel supply, multiple security vulnerabilities become worrisome.

Now, we all know that mistakes happen and that anyone can make them. What matters is how a company owns up to and addresses them once they've been made aware of them. And this is where things go from worrisome to worse.

- On September 9, 2021, BitSight initiated contact via the only email available on the MiCODUS website (sales@micodus.com). MiCODUS replied, asking for additional information to pass on to the MiCODUS sales department. BitSight requested a security or engineering contact. MiCODUS did not respond to that request.

- BitSight contacted MiCODUS on October 1, 2021, again requesting to speak with a security or engineering contact. This request was refused.

- Then MiCODUS contacted BitSight nine days later, on October 10, 2021 claiming to be "working on the issues," despite BitSight not yet sharing any technical information with the vendor.

- On November 23, 2021, BitSight made another attempt to contact the vendor. MiCODUS did not respond.

- On January 14, 2022, BitSight shared its research and findings with CISA to further its efforts. BitSight requested CISA engage with the vendor and share information.

- On May 1, 2022, CISA attempted to contact the vendor to share information. CISA established a connection with the vendor and shared the original research and findings. However, CISA has not heard from the vendor since it shared the research.

- On July 19, 2022, after reasonably exhausting all options to reach MiCODUS and given the lack of engagement from the vendor, BitSight and CISA determined that these vulnerabilities warrant public disclosure. So, CISA and BitSight decided to publish the research.

They wrote: *"Our joint action ensures that organizations have the information they need to proactively protect themselves."*

https://www.bitsight.com/sites/default/files/2022-07/MiCODUS-GPS-Report-Final.pdf

So, what do we and the entire rest of the world now know about one and a half million insecure Chinese vehicle tracking devices operating throughout 169 different countries? Here's how BitSight described their overall findings:

*BitSight discovered six severe vulnerabilities in the MiCODUS MV720 GPS tracker, a popular automotive tracking device designed for vehicle fleet management and theft protection for consumers and organizations. The MV720 is a hardwired GPS tracker, allowing for external, physical control of the device. In addition to GPS tracking, the MV720 offers anti-theft, fuel cut off, remote control, and geofencing capabilities.*

*The exploitation of these vulnerabilities could have disastrous and even life-threatening implications. For example, an attacker could exploit some of the vulnerabilities to cut fuel to an entire fleet of commercial or emergency vehicles. Or, the attacker could leverage GPS information to monitor and abruptly stop vehicles on dangerous highways. Attackers could choose to surreptitiously track individuals or demand ransom payments to return disabled vehicles to working condition. There are many possible scenarios which could result in loss of life, property damage, privacy intrusions, and threaten national security.*

*BitSight's research was conducted with the sole purpose of assessing the security of the MV720 GPS tracker and to determine whether an attacker could access a user's GPS position. Although the results surpassed the proposed initial goal, this report does not represent a full security audit of the MiCODUS ecosystem. However, we believe other models may be vulnerable due to security flaws in the MiCODUS architecture. MiCODUS states there are 1.5 million of their GPS tracking devices in use today by individual consumers and organizations.*

*Organizations and individuals using MV720 devices in their vehicles are at risk. Leveraging our proprietary data sets, BitSight discovered MiCODUS devices used in 169 countries by organizations including government agencies, military, and law enforcement, as well as businesses spanning a variety of sectors and industries including aerospace, energy, engineering, manufacturing, shipping, and more.*

*Given the impact and severity of the vulnerabilities found, it is highly recommended that users immediately stop using or disable any MiCODUS MV720 GPS trackers until a fix is made available.*

Through packet and traffic analysis observed between the website, Android application, GPS trackers, and servers, BitSight determined that the MiCODUS architecture is organized as follows:

All services appear to be hosted by a single server (www.micodus.net/47.254.77.28). It provides a website via HTTPS port 443, an unencrypted API server to support mobile apps via HTTP port 80 (app.micodus.net/47.254.77.28) and a GPS tracker custom protocol server running on port 7700 (d.micodus.net/47.254.77.28). Although the website that's used to access MiCODUS GPS trackers via a browser uses HTTPS, the mobile app uses unencrypted and unauthenticated plain HTTP. GPS trackers communicate with the backend server via a custom protocol on TCP port 7700. This protocol does not appear to be encrypted. Users can directly control and access the GPS tracker via standard SMS text commands. The full command list for model MV720 is:

https://www.micodus.com/uploadfiles/files/eb/eb0440d0bc-mv720-full-sms-commands-list.pdf

Apparently without any logon or authentication, you can send the tracker the simple SMS command "where" and it will reply with a Google Maps link centered on the vehicle's present location. Sending "555" turns off the vehicle's fuel and sending "666" resumes the fuel.

BitSight imagined a couple of classic attacks which are enabled by the vulnerabilities they discovered:

**Man-in-the-Middle Attack:** An attacker performing a man-in-the-middle attack could intercept and change requests between the mobile application and supporting servers, taking advantage of unencrypted HTTP communications. This would give the threat actor complete control of the GPS tracker; access to location information, routes, geofences, and tracking in real-time; as well as the ability to cut off fuel, disarm alarms, and more.

**Authentication Bypass Attack:** A flawed authentication mechanism in the mobile application could allow an attacker to access any device via a hardcoded key. Using this key, an attacker could send messages to the GPS tracker as if they were coming via the SMS channel which should only accept commands from the GPS owner's mobile number. Again, this would give an attacker complete control of the device; access to location information, routes, geofences, and tracking in real-time; and the ability to cut off fuel, disarm alarms, and more.

**Persistent Invisible Monitoring Attack:** It is possible to remotely reprogram the GPS tracker to use a custom IP address as its API server. This would give an attacker the ability to monitor and control all communications to and from the GPS tracker. The attacker could completely control the GPS tracker, with all the implications listed above, *including the reporting of incorrect locations to the GPS server.*

The ability to remotely reprogram these devices to use a persistent custom Internet IP as its API server strikes me as one of those "wouldn't that be cool" or "we can do this so we should" sorts of things that engineers toss in just because they can, without there being any possible need or justification for the feature, and at a significant and serious cost in security. It's so dumb for that to be in there.

Okay. So what are these six vulnerabilities that BitSight found and that CISA agreed were worthy of CVS designation and in one case a CVSS of 9.8?

Let's start with that CVSS of 9.8.

The API server uses a single, global, master password for all devices. This is the password used by the user's mobile apps to query and perform actions and execute remote commands. This allows an attacker to log into the API web server, impersonate any user, and directly send SMS commands to their GPS tracker as if they were coming from the GPS owner's mobile number. Using the master API password, a remote, unauthenticated attacker can:

- Gain complete control of any GPS tracker;
- Access location information, routes, geofences, track locations in real-time;
- Cut off fuel to vehicles; and/or
- Disarm alarms and other features.

It's difficult to label this one a mistake or oversight. They must be well aware that they is not per-account API authentication. All authentication is shared globally. And what's worse, it's not even necessary to reverse-engineer one of their API endpoint apps or rig up some sort of fancy DNS spoofing and TLS-intercepting man-in–the-middle proxy with a fraudulent cert or CA. Since the App's API endpoint is simple HTTP to port 80, any passive web traffic sniffer can be used to capture their application's interaction with the MiCODUS API endpoint server. It's not surprising that CISA scored this one as a 9.8.

Oh, I forgot to mention, that master password, which is stored in all apps and which works universally, is: "7DU2DJFDR8321" And as they wrote: *"Using Key=7DU2DJFDR8321 on any endpoint call, for any user, works."*

**Vulnerability #2:**
The second major problem, also a CVSS 9.8, is another broken authentication. They write:

> The API server provides a way to directly send SMS commands to the GPS tracking device as if those messages were coming from the administrator's mobile device. 123456 is the default GPS tracker password, which should be changed. However, some commands work even without a password.
>
> The web interface and mobile app also require a password when directly contacting the tracker via SMS. However, it shares the same default password issues as the GPS tracker. Even if the user changes the password, the device is not secure. Some SMS-like command messages sent directly from the API server do not need the device password to function, leaving the device exposed to attackers.
>
> One potential attack can be perpetrated by abusing the **adminip** command [this is the feature that I noted earlier should never have been included in the API] which defines the API endpoint on the GPS tracker. This enables an attacker to achieve a persistent man-in-the-middle position, controlling all traffic between the GPS tracker and the original server, and gaining total control over the GPS tracker.

In their full disclosure, they provide a working proof of concept where they redirect the API server from one of the GPS trackers to their own server, which is able to do whatever it wishes with it. It could forward it, or modify it to, for example, show that the GRC tracker is in a different location than it actually is.

**Vulnerability #3:**
And the third vulnerability, coming in with a respectable CVSS score of 8.1, we have the always popular default password of "123456". They wrote:

> *As noted above, all devices ship preconfigured with the default password 123456, as does the mobile interface. There is no mandatory rule to change the password nor is there any claiming process. The setup itself does not require a password change to use the device. We observed that many users have never changed their password. Since there is no proper claiming procedure, users are not forced or encouraged to change their passwords, the server does not seem to have any password brute force or rate limiting in place, and because the Device ID is easily predictable, attackers can easily access random GPS trackers.*
>
> *Although CISA did not assign a unique CVE to this issue we identified, we nevertheless believe it represents a severe vulnerability. Assigning a default password to a service or device that is readily reachable via the Internet, with no mechanism to force the user to change it, has proven to be a crucial security mistake and a consistent item on the OWASP Top 10 list.*

It's 2022. Once upon a time, maybe back in the 1990's, someone might have been able to make the argument that forcing people to come up with a unique high-quality password might put someone off. Okay, even back then probably not. But in 2022, you cannot use the Internet without at least having the concept of unique strong passwords. Even if you insist upon not using a different one for every website. I can see not caring about the password for some random login once website. But the password for a GPS tracking widget that's empowered to cut off the gas supply to the vehicle? That's probably worth assigning a unique password.

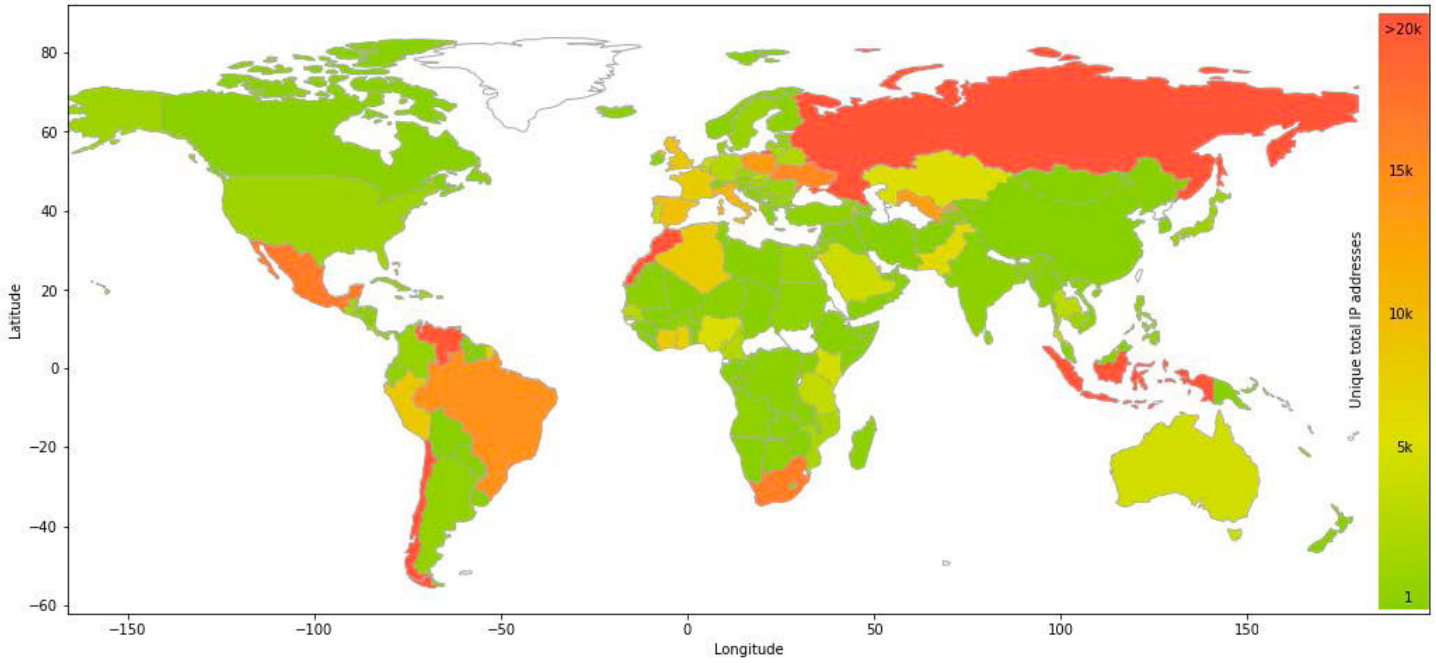
Doing a bit of reverse engineering, the API queries and replies they examined used the field "Name" as the Device ID, and the MV720 device models follow the pattern: 72011XXXXXX, with the value represented by the X's occurring in sequential order. So these researchers performed a random scan of 1,000 responding Device IDs following the pattern they had identified. Of those 1,000. 945 (or 94.5%) were, not surprisingly, using the default password: 123456.

A big problem is that the need for security, the things that security is securing, are more often than not a complete mystery to a device's new user. If someone explained to them what the consequence would be of **not** changing the password, they almost certainly would happily change it. Default passwords should never be used anywhere. Even a bad password is better than most of them being known and identical.

The final three are also serious vulnerabilities but they are less glamorous. There's a cross-site scripting vulnerability with a CVSS of 7.5 in their web server which would allow an attacker to inject their own code into an unwitting user's browser when displaying pages from the MiCODUS site.

The last two problems are described as "Insecure Direct Object References" facilitated on the web server. Generically, Insecure Direct Object References" are a form of access control vulnerability which occurs when an application uses user-supplied input to directly access objects, without further verification. In this instance, the API implicitly trusts the Device ID being provided by the device to identify itself. This means that it's possible to change the declared Device ID, regardless of the logged-in user, to access data from **any** device in the server's database. This might make additional information available such as license plate numbers, SIM card numbers, mobile numbers, and so forth. Those last two earned scores of 7.1 and 6.5.

These little gremlins are located all over the place as shown by the heat map I've dropped into the show notes:




Fortunately, the U.S. does not appear to be a hotspot and Canada is even cooler. But Mexico has a bunch, as does South America. About the geographic distribution, BitSight wrote:

> Based on observable trends in the data, BitSight can theorize about the usage scenarios in each country. For example, Indonesia has many unique IP addresses communicating with the MiCODUS server, but mostly in the GPS tracker port. This may suggest there are a small number of users with a high number of devices, which is typical in a fleet management scenario. By comparison, Mexico has a very high number of connections to the web and mobile ports, which could indicate individuals are using the GPS tracker as an anti-theft device.
>
> These, of course, are just assumptions; without direct knowledge and engagement from the vendor, we are left to hypothesize about the exact purpose each device is fulfilling for each user. Regardless, we presume access to the web port is strongly related to the number of unique MiCODUS users. Web access IP addresses tend to be relatively fixed, hence less unique IPs, regardless of the number of connections. As a result, this measurement is likely a good indicator of distinct MiCODUS clients.

And I'll just note that the very fact BitSight was able to gain these insights is further demonstration of how broken this system's security is.

BitSight summed their findings by writing:

> Although GPS trackers have existed for many years, streamlined manufacturing of these devices has made them accessible to anyone. Having a centralized dashboard to monitor GPS trackers with the ability to enable or disable a vehicle, monitor speed, routes and leverage other features is useful to many individuals and organizations. However, such functionality can introduce serious security risks.
>
> Unfortunately, the MiCODUS MV720 lacks basic security protections needed to protect users from serious security issues. With limited testing, BitSight uncovered a multitude of flaws affecting all components of the GPS tracker ecosystem.
>
> BitSight recommends that individuals and organizations currently using MiCODUS MV720 GPS tracking devices disable these devices until a fix is made available. Organizations using any MiCODUS GPS tracker, regardless of the model, should be alerted to insecurity regarding its system architecture, which may place any device at risk.
>
> This research highlights why it is critical to consider Internet of Things (IoT) devices in cyber resilience efforts. Implementing Internet connected devices like the MiCODUS GPS trackers discussed in this report can expand an organization's attack surface and expose individual consumers to new risks. Understanding how IoT and other technologies impact risk should be considered essential.

Amen to that.

The takeaway lesson for us here is to take this as a valuable case study. We need to recognize that when we're using anything such as this, which connects to a remote Internet service of unknown reputation, we truly are placing a huge amount of trust into entities whose trust has not been earned and may not be deserved.

It's bad enough to have one's light switches and plugs connecting back to potentially hostile foreign soil. But giving remote (and in this case clearly irresponsible) entities real time knowledge of vehicle location and movement and even over the vehicle's real time fuel flow, seems reckless at best.

But, hey... it's only $26 on Amazon!