Security Now! #933 - 08-01-23

TETRA:BURST

This week on Security Now!

It turns out that Advanced Persistent Threats have been leveraging satellite communications for many years. We start by looking at that. Then we'll find out what the next iOS release will be doing to further thwart device tracking. What new feature is Android 6+ releasing? What's the latest on the forthcoming 7th branch of the U.S. military? Why has Russia suddenly criminalized open source contribution? And what do we learn from VirusTotal's 2023 "malware-we've-seen" update? Then, after we share some of the terrific podcast-relevant feedback received from our amazing listeners following last week's second satellite insecurity podcast, we're going to examine one of the revelations to be detailed during next week's Blackhat hacking conference in Las Vegas.

Never play with super glue...

Security News

Satellite Turla: APT Command and Control in the Sky

Before we wander away, at least for the time being, from the topic of satellite security, I wanted to talk about another aspect of the use of satellites by bad guys, which is the deliberate routing of Internet connections through space. This is done as a means of thwarting the persistent efforts by law enforcement to track down, shutdown and sometimes take over the command and control servers being used by the major advanced persistent threat groups. Since it's another thing we've never explicitly covered, I thought that now, while we're looking skyward, would be a good time to add this to the growing list of things we **have** covered.

Back in September of 2015, Kaspersky published an informative research piece titled: "Satellite Turla: APT Command and Control in the Sky" Kaspersky stopped short of explaining the network packet flow in detail, but they provided enough for us to fill in the rest of the technology. So first, after I skipped over some of the warm-up introduction, which would be redundant for the listeners to this podcast, Kaspersky explained:

When you are an APT group, you need to deal with many different problems. One of them, and perhaps the biggest, is the constant seizure and takedown of domains and servers used for command-and-control (C&C). These servers are constantly appropriated by law enforcement or shut down by ISPs. Sometimes they can be used to trace the attackers back to their physical locations.

Some of the most advanced threat actors or users of commercial hacking tools have found a solution to the takedown problem — the use of satellite-based Internet links. In the past, we've seen three different actors using such links to mask their operations. The most interesting and unusual of them is the Turla group.

Also known as Snake or Uroburos, names which come from its top class rootkit, the Turla cyber-espionage group has been active for more than 8 years [and that was 8 years back in 2015]. Several papers have been published about the group's operations, but until recently little information was available about the more unusual aspects of their operations, such as the first stages of infection through watering-hole attacks.

What makes the Turla group special is not just the complexity of its tools, which include the Uroboros rootkit, aka "Snake", as well as mechanisms designed to bypass air gaps through multi-stage proxy networks inside LANs, but the exquisite satellite-based C&C mechanism used in the latter stages of the attack.

In this blog, we hope to shed more light on the satellite-based C&C mechanisms that APT groups, including the Turla/Snake group, use to control their most important victims. As the use of these mechanisms becomes more popular, it's important for system administrators to deploy the correct defense strategies to mitigate such attacks. For IOCs, see the appendix.

Although relatively rare, since 2007 several elite APT groups have been using — and abusing — satellite links to manage their operations — most often, their C&C infrastructure. Turla is one of them. Using this approach offers some advantages, such as making it hard to identify

the operators behind the attack, but it also poses some risks to the attackers.

On the one hand, it's valuable because the true location and hardware of the C&C server cannot be easily determined or physically seized. Satellite-based Internet receivers can be located anywhere within the area covered by a satellite, and this is generally quite large. The method used by the Turla group to hijack the downstream links is highly anonymous and does not require a valid satellite Internet subscription.

On the other hand, the disadvantage comes from the fact that satellite-based Internet is slow and can be unstable.

In the beginning, it was unclear to us and other researchers whether some of the links observed were commercial Internet connections via satellite, purchased by the attackers, or if the attackers had breached the ISPs and performed Man-in-the-Middle (MitM) attacks at the router level to hijack the stream. We have analyzed these mechanisms and come to the astonishing conclusion that the method used by the Turla group is incredibly simple and straightforward, as well as highly anonymous and very cheap to operate and manage.

Purchasing satellite-based Internet links is one of the options APT groups can choose to secure their C&C traffic. However, full duplex satellite links can be very expensive: a simple, duplex, 1Mbit up/down satellite link may cost up to $7000 per week. For longer term contracts this cost may decrease considerably, but the bandwidth still remains very expensive. [Again, this was back in 2015 so things may have changed since.]

Another way of getting a C&C server into a satellite's IP range is to hijack the network traffic between the victim and the satellite operator and to inject packets along the way. This requires either exploitation of the satellite provider itself, or of another ISP on the way.

These kinds of hijacking attacks have been observed in the past and were documented by Renesys (now part of Dyn) in a blogpost dated November 2013.

According to Renesys: "Various providers' BGP routes were hijacked, and as a result a portion of their Internet traffic was misdirected to flow through Belarusian and Icelandic ISPs. We have BGP routing data that show the second-by-second evolution of 21 Belarusian events in February and May 2013, and 17 Icelandic events in July- August 2013."

In a more recent blogpost from 2015, these researchers point out that: "For security analysts reviewing alert logs, it is important to appreciate that the IP addresses identified as the source of incidents can and are regularly spoofed. For example, an attack that appeared to come from a Comcast IP located in New Jersey may really have been from a hijacker located in Eastern Europe, briefly commandeering Comcast IP space. It is interesting to note that all six cases discussed above were conducted from either Europe or Russia."

Obviously, such incredibly apparent and large-scale attacks have little chance of surviving for long periods of time, which is one of the key requirements for running an APT operation. It is therefore not very feasible to perform the attack through MitM traffic hijacking, unless the

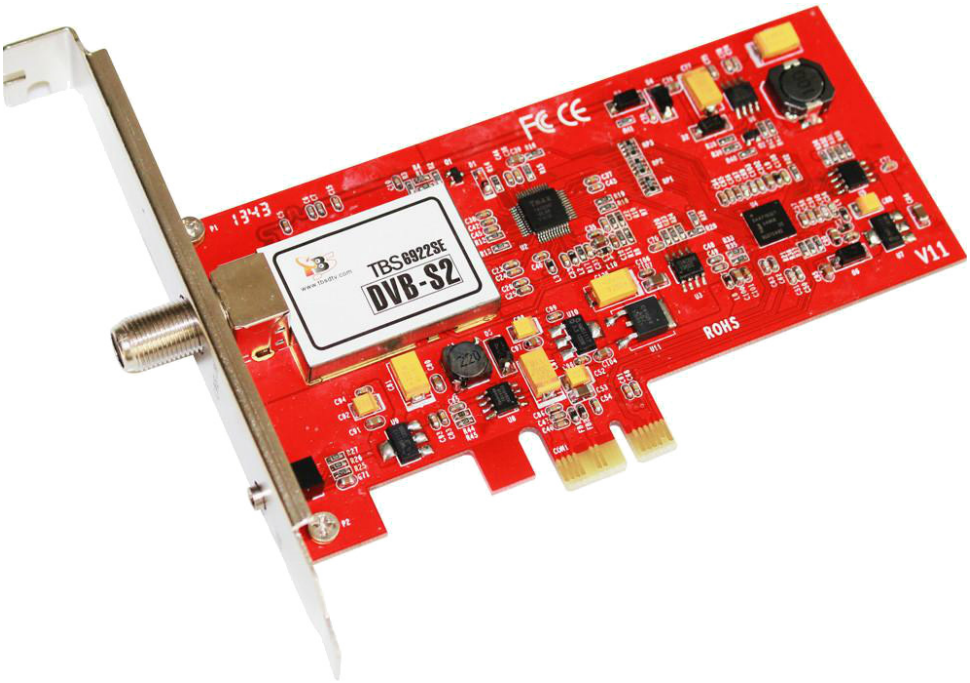
attackers have direct control over some high-traffic network points, such as backbone routers or fiber optics. There are signs that such attacks are becoming more common, but there is a much simpler way to hijack satellite-based Internet traffic.

Enter: Satellite link (DVB-S) hijacking

The hijacking of satellite DVB-S links has been described a few times in the past and a presentation on hijacking satellite DVB links was delivered at BlackHat 2010 by an S21Sec researcher.

To hijack satellite DVB-S links, one needs the following:

- *A satellite dish – the size depends on geographical position and satellite*
- *A low-noise block downconverter (LNB)*
- *A dedicated DVB-S tuner (PCIe card)*
- *A PC, preferably running Linux*



While the dish and the LNB are more-or-less standard, the card is perhaps the most important component. Currently, the best DVB-S cards are made by a company called TBS Technologies. The TBS-6922SE is perhaps the best entry-level card for the task. (~$99 USD)

The TBS card is particularly well-suited to this task because it has dedicated Linux kernel drivers and supports a function known as a brute-force scan which allows wide-frequency ranges to be tested for interesting signals. Of course, other PCI or PCIe cards might work as well, while, in general the USB-based cards are relatively poor and should be avoided.

Unlike full duplex satellite-based Internet, the downstream-only Internet links are used to accelerate Internet downloads and are very cheap and easy to deploy. They are also inherently insecure and use no encryption to obfuscate the traffic. This creates the possibility for abuse.

Kaspersky's article didn't go into any more detail about how this works. They switched to providing tables of IP ranges that had been observed in the past and noted the satellite Internet providers that were using those ranges.

But fortunately we have all the information we need to understand the advantage this gives anyone who is attempting to hide their command and control server.

The key is that these Internet communication satellites have extremely broad coverage areas, coupled with the fact that, just like the Internet, the IP packet traffic being carried is not, itself, encrypted. As we know, TCP and UDP are not themselves encrypted protocols. They're just carriers of data that today is typically encrypted. But the Internet's packets themselves are not.

So, imagine that some nasty advanced persistent threat malware has been surreptitiously placed into a high-value computer, and that more than anything the bad guys do not want their command and control infrastructure – which this malware will be reaching out to for instructions – to be discovered, commandeered and shutdown. Presumably this APT group has many such infestations which are all reusing the same infrastructure. So the loss of that command and control server would cripple their entire network.

Okay. So they have their APT malware periodically send a UDP packet to the IP of a previously chosen customer – probably a big stable customer – of a given satellite-based Internet provider.

Having the malware send an outbound UDP packet has the effect of opening up return paths through any NAT routing and firewalls that would otherwise prevent unsolicited traffic from entering the enterprise's network and reaching the malware-laden machine.

So this UDP packet is sent out to a previously selected customer of a satellite ISP. So it will be received first by that ISP. But unlike other ISPs, the received packet is beamed up-stream directly at a chosen communications satellite. This causes it to then be broadcast out across the entire coverage area of the satellite. Somewhere, down on the ground, is that subscriber, but also somewhere else – anywhere else within the satellite's large coverage area – the malicious command and control server is silently lurking with its own satellite dish passively aimed up at the IPS's broadcasting satellite. It patiently listens for any UDP packets addressed to that IP. Since the subscriber will likely have a NAT router or firewall that will simply ignore any unsolicited nonsense (and since they will have been pre-selected for that to be true) their receipt of that incoming packet will be ignored. But it will be what the malicious command and control receiving base station has been waiting for. Upon receiving that UDP packet, the base station can reply by sending its own UDP reply packet via the terrestrial ground Internet since there's no need for it to be returned to space. The reply packet, carrying the spoofing the IP and port of the original packet's intended receiver, allows the command and control system to send whatever commands it wishes back to the querying machine.

And the traffic doesn't need to be only UDP. Nothing prevents the listening C&C base station from establishing a full 3-way handshake and bringing up an encrypted TCP connection.

The key to the hack is that it's the world's largest airgap. The outbound traffic is being sprayed over a huge geographic area to be picked up by a totally passive satellite dish receiver which can

be anywhere. And the command and control system's IP address being used is someone else's, so it's also an air gapped man-in-the-middle traffic interception attack. You have to give the bad guys credit. It's a slick hack.

iOS 17 to further crack down on device fingerprinting

Apple just updated its developer program to further crack down on developers who are abusing some of its API features to collect data on user devices as an underhanded means of tracking them online. Apple says that even if a user has given an app permission to track their activity, fingerprinting the underlying device is still not allowed. So, with the release of iOS 17 and macOS Sonoma this fall, developers who want to access certain API features — which could be and have been used to enable persistent device-level tracking — will have to provide a valid reason to do so. Apps that don't provide a good reason won't be accepted on the App Store starting next spring.

Android to start warning of "unknown trackers"

And speaking of tracking, but this time tracking that's more deliberate, we recently took a deep dive into AirTags tracking technology, prompted by the announcement that Apple and Google were going to arrive at a common specification. Since Bluetooth-based tracking is inherently crowd sourced, it is in both party's interests to have a single common standard so that both Apple and Android handsets can provide tracking location feedback for the other's ecosystem. Given that the "joint" specification was indistinguishable from that Apple had already been doing for several years, what appeared to have actually happened was simply that Apple opened their specification for Google's welcome adoption. And that's good for everyone.

So, last Thursday's news is that Google will soon be adding "Unknown Tracking Alerts" to Android. They said :"Unknown tracker alerts, which we announced at I/O 2023, are beginning to roll out to Android 6.0+ users this month." and "Unknown tracker alerts currently work with Apple AirTags. We'll continue to work with tag manufacturers to expand this important protection to other tracking tags over time through our joint industry specification."

The 7th branch of the US military

The National Defense Authorization Act which successfully passed through the U.S. Senate last week included a provision requiring the National Academy of Public Administration to conduct an assessment on the feasibility of establishing a new, formal, 7th branch of the U.S. military: The US Cyber Force. So, this does appear to be happening. Since many of our listeners have explained that wearing ridiculous camouflage clothing indoors is a bizarre requirement of the U.S. military (my word "bizarre" not theirs), perhaps, at least, the Cyber Force's cammo could have a cool cyber theme? ... like the green falling and fading symbols from The Matrix? Or maybe do the whole thing just in 1's and 0's? Anyway, I hope someone gives this need as much serious thought as we have here!

The other wrinkle is that both the Army and Air Force have recently created their own new specialized cyber teams to support their traditional "kinetic teams" with cyber tasks related to intelligence gathering, electronic warfare, and sensors. This makes sense, since those cyber

teams supporting traditional kinetic warfare are probably highly targeted and specialized for their specific tasks, whereas the military's new 7th branch would be far more wide ranging and not at all focused upon specific current Army and Air Force military operations.

But through all this it's quite obvious that “cyber” has well and truly arrived, both on the front lines, and in dimly lit dens filled with monitors and empty caffeinated beverage cans.

Russia criminalizes open source project contribution

Meanwhile, Russia continues to separate itself from the West. The Russian Parliament just passed three bills which, once signed by Putin will ban Russian citizens from participating in the activities of foreign non-profit organizations that have not specifically registered with the Russian government — and none have. Commentary about this over on opennet.ru notes that an unintended side effect will be that Russians using open source software would be prevented from contributing in any way to those projects, even from submitting bug reports. As we know, today's open source software includes Linux, Firefox and most major database systems and programming languages. I read the entire piece after having Google translate it into English, and it only talked about the unintended consequences. I was unable to determine that the intended consequences of the three pending bills would be. Why would Russia think this was a good idea?

VirusTotal's 2023 report

VirusTotal is out with their look at 2023 to date. This is always interesting since it highlights the broad trends which unpin the specifics that we looks at every week. So, the main takeways from this most recent update are that:

Email attachments continue to be the most popular way to spread malware.

Traditional file types (Excel, RTF, CAB and compressed formats) are becoming less popular. Although the use of PDFs slowly decreased for the last few months in June 2023, the biggest peak in PDF usage was observed during 2023 over last two years.

OneNote and JavaScript (distributed by HTML) are the most rapidly growing formats for malicious attachments in 2023 with OneNote emerging in 2023 as a reliable alternative for attackers to the traditional use of macros in other Office products.

Malicious OneNote files usually embed an additional malicious file (vba, html+jscript, powershell, or any combination of them) and, as happens with malicious Office attachments, the attempt is then made to convince the victim to allow execution.

Payloads vary from malware family to family, but many of them access external URLs to download a DLL file camouflaged as a PNG file. This very old trick is used to bypass simple firewall rules or just to appear less suspicious to the eye.

The most usual kill chain where OneNote format is involved is as follows:

1. The victim receives an email with a OneNote attachment. The mail body encourages the victim to click on a button to see a hidden/distorted image or document.

2. This button executes a script (VB script, HTA, powershell, etc,) that will launch a payload, either embedded into the same script or downloaded from an external resource.

3. The external payload might be yet another OneNote file, an image file renamed as a ".bat" file, a DLL file that's loaded into memory or even a Windows executable.

So, we have inherently dangerous capabilities mixed with social engineering attacks. And only one mistake made by one curious or inattentive person within a major organization is all that's required to invite the malware in.

ISO image files for malware spreading are a flexible alternative for both widespread and targeted attacks. And their distribution as heavily compressed attachments makes them difficult to scan by some security solutions. ISO files are being disguised as legitimate installation packages for a variety of software, including Windows, Telegram, AnyDesk, and malicious CryptoNotepad, among others.

Our data shows that there was an increase in the number of malicious files attached to emails between March and April of 2023. In terms of suspicious attachments, for the past two years, we have observed spikes in the number of suspicious PDF files linked to malicious campaigns. These files can be used for a variety of purposes, such as exploiting vulnerabilities (less usual) or phishing (most of the time).

And during 2023, we saw a significant increase in the use of JavaScript distributed alongside HTML, in sophisticated phishing attacks designed to steal victims' credentials. Excel, RTF, CAB, compressed formats, and Word all seem to be declining in popularity as malicious attachments compared to OneNote and JavaScript.

Closing the Loop

Jeff Parrish / @kb9gxk

Thank you for another great episode. I am IT for a healthcare facility and this episode made me review the HTML of our EHR provider. I have now contacted them about the Google Analytics tracking they have on their site after we are logged in.

Robert C. Covington / @_covington

Long time listener. I oversee Cyber Security for a large children's hospital system. Your podcast transcripts are frequently on my screen during team briefings. Regarding web site tracking and the recent OCR notice referenced in episode 932, there is a side consequence I have not heard mentioned. Cyber insurance companies are now declining to cover any legal actions arising out of web site tracking and collection of PHI (Personal Health Information). This is sending many healthcare orgs scrambling to get tracking tools off of their web sites.

Keep up your excellent work!
Robert Covington
PS you fell into the classic trap on 932. It is HIPAA, not HIPPA ;-)

Jon Dagle / @jondagle

Hi Steve, Thanks for the shout-out on the 25 July episode. I am the "neat guy" who you saw on TWIS talking about orbital debris. (I'm fairly sure you weren't referring to Geof and I'm sure you weren't referring to Rod! haha) Thank you for your kind mention. I've been a SN listener since Episode 1 – proud SpinRite owner (and somewhere I have a certificate for a TWIT Brick). Pretty sure I've not missed a single episode (at least not a whole one). At the beginning I was in the USAF. I stuck around for hobbyist purposes and with a plan to go into cybersecurity, but I made a detour into space policy.

Orbital debris is a clear and present concern, if not actual danger. The space advocacy organization where I work considers this one of a handful of high priorities. While there are a number of public sources for tracking objects in orbit, they don't all agree. According to orbit.ing-now.com a relatively approachable source. A high-level summary is available here: https://nanoavionics.com/blog/how-many-satellites-are-in-space/

There are about 7700-8400 active human made satellites in orbit around our planet. The vast majority (90%) are in low-Earth orbit, <1000km. About 1/3 of these have been added in the past few years, mainly by SpaceX Starlink. About 7% of the total are in geostationary orbit, with the remainder in medium-Earth orbit (very few). Almost 2300 "inactive" satellites. Thanks for the shout-out and the "brush with greatness." ;)

Jonathan / Washington, DC / Policy Chair, National Space Society

It was definitely you, Jon, whom I was referring to last week, having seen that piece of This Week in Space. So thanks for the additional info! And we have another John, John Sutherland, who knows more than he's able to share...

John Sutherland / @John0rion (sent in two Twitter DMs…)

I wanted to offer a bit of knowledge I had about US military satellites. I was active duty and what is now Space Force for 11 years, and I'm currently a contractor still supporting space. I "flew" SAT-COM for 4 years, then taught for 7 years. I taught both classified and unclassified classes so I am very familiar with where the line is for classified. So I can go right up to that line.

Having just finished the second part of the Satellite Insecurity, I can share that, luckily, most of the problems you talked about are not as true for US DoD satellites. The preconceptions that attackers would not have the equipment was never the case, China and Russia have always had similar ground station capabilities as we have. The oldest satellites I have worked with were developed in the late 80s and they were highly encrypted and rolled keys constantly. For communication satellites the data is just routed so encryption is as good as it could be on Earth and not subject to the satellites' age. Controlling the satellites i.e. moving them, changing configuration, is done with separate antennas that are monitored and any communication with them is watched real time. If someone did break this encryption it would quickly be found out.

As for physical attacks, the arms of attacking satellites is only a start. When we table topped attacks and planned responses, TTP (tactics techniques and procedures), we looked at Jamming, ASats, mechanical arms, and lasers. Jamming being the most common and ones we have actually seen happen. Most jammers are big ground-based semi-trucks or ships that just try to over power the uplink. We have many mitigations to this and I taught a class on RF Attack and Defense as part of operators' advanced training. ASats as you talked about with blowing up satellites from the ground are extremely unlikely at this point. We are much more concerned with small satellites with explosives. The idea being that an adversary would place and leave something small on a foreign satellite that could be triggered on demand at any time in the future. I cannot talk to the mechanical arms as the line beyond which I cannot talk is around this. But it's safe to say this has been looked at and is in some level of development by both sides. Lasers are not a threat to all types of satellites but China and Russia have used lasers to blind sensors of low flying "spy" satellites. This is hard to guard against, but we do equip satellites with shutters now, and for satellites lacking shutters, we only need to spin them around.

There's more that can't be talked about but. Bith your level of technical knowledge and a little imagination you could get close to guessing what is going on. I can tell you I've never been surprised when I got a security briefing.

Mikael Falkvidd / @mfalkvidd
Mikael is on the board of OWASP Gothenburg, Sweden who invited me to present SQRL to their group. It turns out that Mikael knows more than a little bit about satellite software…

Regarding authenticated telecommands to satellites: What satellite programmers are most afraid of is bit flips (caused by single-event upset (SEU) which happen due to radiation in space). Imagine that a SEU flips a bit in the key used to authenticate the telecommand. Authentication would fail. And guessing which bit (s) flipped can take a lot of time. There are of course mitigations, for example using error correction codes or storing the key in multiple places. But complexity is the ememy of reliability, and resources (compute, flash, ram) on board satellites have been very scarce historically. And people want reliable satellites, so they are hesitant to introduce new features. "Flight-proven" is the mantra, so the old ways live on. The risk of losing the satellite because of a SEU has been deemed higher than the risk that the satellite is hacked. Not an excuse today, but that's how the industry is. (I have written software for two satellites) SEUs are also (one of) the reasons telecommands exist to write to any memory location. Nasa used this feature to restore a bit flip on Voyager 2 in 2010 (33 years after its launch).

Mikael also provided a link to a summary from JPL, the Jet propulsion Laboratory in Pasadena, which documented events surrounding exactly this happening back in May of 2010. Somewhat astonishingly, Voyager 2 remains alive and functioning to this day, though something happened with it just last week, which I'll get to in a second. We last checked-in on Voyager 2 nearly five years ago when, on Nov. 5, 2018, it became only the second spacecraft to exit our solar system's heliosphere. And we considered whether this event might break the simulation that some people appear to be convinced we are all "living" within. But, so far, the simulation appears to be holding.

But first let's turn the calendar back 13 years to May 6th of 2010, when JPL wrote:

Engineers have shifted NASA's Voyager 2 spacecraft into a mode that transmits only spacecraft health and status data while they diagnose an unexpected change in the pattern of returning data. Preliminary engineering data received on May 1 show the spacecraft is basically healthy, and that the source of the issue is the flight data system, which is responsible for formatting the data to send back to Earth. The change in the data return pattern has prevented mission managers from decoding science data.

The first changes in the return of data packets from Voyager 2, which is near the edge of our solar system, appeared on April 22. Mission team members have been working to troubleshoot and resume the regular flow of science data. Because of a planned roll maneuver and moratorium on sending commands, engineers got their first chance to send commands to the spacecraft on April 30. It takes nearly 13 hours for signals to reach the spacecraft and nearly 13 hours for signals to come down to NASA's Deep Space Network on Earth.

Voyager 2 launched on August 20, 1977, about two weeks before its twin spacecraft, Voyager 1. The two spacecraft are the most distant human-made objects, out at the edge of the heliosphere, the bubble the sun creates around the solar system. Mission managers expect Voyager 1 to leave our solar system and enter interstellar space in the next five years or so, with Voyager 2 on track to enter interstellar space shortly afterward. Voyager 1 is in good

health and performing normally.

Ed Stone, Voyager project scientist at the California Institute of Technology in Pasadena said: "Voyager 2's initial mission was a four-year journey to Saturn, but it is still returning data 33 years later. It has already given us remarkable views of Uranus and Neptune, planets we had never seen close-up before. We will know soon what it will take for it to continue its epic journey of discovery."

The original goals for the two Voyager spacecraft were to explore Jupiter and Saturn.

As part of a mission extension, Voyager 2 also flew by Uranus in 1986 and Neptune in 1989, taking advantage of a once-in-176-year alignment to take a grand tour of the outer planets. Among its many findings, Voyager 2 discovered Neptune's Great Dark Spot and 450-meter-per-second (1,000-mph) winds. It also detected geysers erupting from the pinkish-hued nitrogen ice that forms the polar cap of Neptune's moon Triton. Working in concert with Voyager 1, it also helped discover actively erupting volcanoes on Jupiter's moon Io, and waves and kinks in Saturn's icy rings created by tugs of nearby moons.

Voyager 2 is about 13.8 billion kilometers, or 8.6 billion miles, from Earth. Voyager 1 is about 16.9 billion kilometers (10.5 billion miles) away from Earth.

The Voyagers were built by JPL, which continues to operate both spacecraft. Caltech manages JPL for NASA.

Okay. So May 6th, 2010 and something is broken and has gone wrong with Voyager 2 such that the spacecraft's science data is no longer being properly formatted. Eleven days later on May 17th, 2010, we learn what went wrong:

Engineers at NASA's Jet Propulsion Laboratory said Monday, May 17 that one flip of a bit in the memory of an onboard computer appears to have caused the change in the science data pattern returning from Voyager 2. A value in a single memory location was changed from a 0 to a 1. On May 12, engineers received a full memory readout from the flight data system computer, which formats the data to send back to Earth. They isolated the one bit in the memory that had changed, and they recreated the effect on a computer at JPL. They found the effect agrees with data coming down from the spacecraft. They are planning to reset the bit to its normal state on Wednesday, May 19.

And then, three days later on May 20th we have the report of the conclusion of this high stakes drama:

Engineers have successfully corrected the memory on NASA's Voyager 2 spacecraft by resetting a computer bit that had flipped. Reset commands were beamed up to the spacecraft yesterday, Wed., May 19, and engineering data received today confirm that the reset was successful. The Voyager team will continue monitoring the engineering data, and if the bit remains properly reset, commands to switch to the science data mode will be beamed up to Voyager 2 on Sat., May 22. Receipt of science data would then resume on Sun., May 23.

And all of that did happen on schedule. But I also noted that something had also happened just last week. NASA's blog posting last Friday, July 28th read:

A series of planned commands sent to NASA's Voyager 2 spacecraft on July 21St inadvertently caused the antenna to point 2 degrees away from Earth. As a result, Voyager 2 is currently unable to receive commands or transmit data back to Earth.

Voyager 2 is currently located almost 12.4 billion miles (19.9 billion kilometers) from Earth and this change has interrupted communication between Voyager 2 and the ground antennas of the Deep Space Network (DSN). Data being sent by the spacecraft is no longer reaching the Deep Space Network, and the spacecraft is not receiving commands from ground controllers.

Voyager 2 is programmed to reset its orientation multiple times each year to keep its antenna pointing at Earth; the next reset will occur on Oct. 15, which should enable communication to resume. The mission team expects Voyager 2 to remain on its planned trajectory during the quiet period.

Voyager 1, which is almost 15 billion miles (24 billion kilometers) from Earth, continues to operate normally.

And, finally, a couple of interesting tidbits about the Voyager probes:

Uplink communications is via S-band (16-bits/sec command rate) while an X-band transmitter provides downlink telemetry at 160 bits/sec normally and 1.4 kbps for playback of high-rate plasma wave data. All data are transmitted from and received at the spacecraft via the 3.7 meter high-gain antenna (HGA).

Electrical power is supplied by three Radioisotope Thermoelectric Generators (RTGs). The current power levels are about 249 watts for each spacecraft. As the electrical power decreases, power loads on the spacecraft must be turned off in order to avoid having demand exceed supply. As loads are turned off, some spacecraft capabilities are eliminated.

NASA maintains an extremely cool read-time Voyager status page which continuously shows the location of both spacecraft and other interesting tidbits such as which science modules are presently turned on and off given the amount of available power which is slowly dwindling from each craft's three Radioisotope Thermoelectric Generators. The page is so cool that I created a GRC.SC shortcut to make it easy to find: https://grc.sc/voyager https://voyager.jpl.nasa.gov/mission/status/ Or you can just Google "Voyager Mission Status."

So, a big thanks to our satellite-informed listeners for their very interesting feedback!

Jon David Schober / @jondavidschober

Hey Steve, On SN932, I heard you talking about how you are keeping the rack of servers at Level 3, and not moving to "the cloud". In case you wanted some interesting reading, here is a blog post from David Hansson, founder of 37Signals and Basecamp and creator of Ruby on Rails. He discusses how they regret moving their business to AWS, how expensive everything was, and how much better life is being back on their own hardware

So, first of all, Jon, thanks very much for the pointer. Since this topic is quite near and dear to my heart, and since I think it might also be extremely interesting to a large number of our listeners, I want to share the blog post that Jon pointed to. As Jon said, this was written by David Hansson and posted last October 19th, 2022 titled: ***"Why we're leaving the cloud"***:

Basecamp has had one foot in the cloud for well over a decade, and HEY has been running there exclusively since it was launched two years ago. We've run extensively in both Amazon's cloud and Google's cloud. We've run on bare virtual machines, we've run on Kubernetes. We've seen all the cloud has to offer, and tried most of it. It's finally time to conclude: ***Renting computers is (mostly) a bad deal for medium-sized companies like ours with stable growth. The savings promised in reduced complexity never materialized. So we're making our plans to leave.***

The cloud excels at two ends of the spectrum, where only one end was ever relevant for us. The first end is when your application is so simple and low traffic that you really do save on complexity by starting with fully managed services. This is the shining path that Heroku forged, and the one that has since been paved by Render and others. It remains a fabulous way to get started when you have no customers, and it'll carry you quite far even once you start having some. (Then you'll later be faced with a Good Problem once the bills grow into the stratosphere as usage picks up, but that's a reasonable trade-off.)

The second is when your load is highly irregular. When you have wild swings or towering peaks in usage. When the baseline is a sliver of your largest needs. Or when you have no idea whether you need ten servers or a hundred. There's nothing like the cloud when that happens, like we learned when launching HEY, and suddenly 300,000 users signed up to try our service in three weeks instead of our forecast of 30,000 in six months.

But neither of those two conditions apply to us today. They never did for Basecamp. Yet by continuing to operate in the cloud, we're paying an at times almost absurd premium for the possibility that it could. It's like paying a quarter of your house's value for earthquake insurance when you don't live anywhere near a fault line. Yeah, sure, if somehow a quake two states over opens the earth so wide it cracks your foundation, you might be happy to have it, but it doesn't feel proportional, does it?

Let's take HEY as an example. We're paying over half a million dollars per year for database (RDS) and search (ES) services from Amazon. Yes, when you're processing email for many tens of thousands of customers, there's a lot of data to analyze and store, but this still strikes me as rather absurd. Do you know how many insanely beefy servers you could purchase on a budget of half a million dollars per year?

Now the argument always goes: Sure, but you have to manage these machines! The cloud is so much simpler! The savings will all be there in labor costs! Except no. Anyone who thinks running a major service like HEY or Basecamp in the cloud is "simple" has clearly never tried. Some things are simpler, others more complex, but on the whole, I've yet to hear of organizations at our scale being able to materially shrink their operations team, just because they moved to the cloud.

It was a wonderful marketing coup, though. Sold with analogies like "well you don't run your own powerplant either, do you?" or "are infrastructure services really your core competency?". Then lathered up with a thick coat of NEW-NEW-NEW paint, and The Cloud has beamed so brightly only the luddites would consider running their own servers in its shadow.

Meanwhile Amazon in particular is printing profits renting out servers at obscene margins. AWS' profit margin is almost 30% ($18.5b in profits on $62.2B in revenue), despite huge investments in future capacity and new services. This margin is bound to soar now that "the firm said it plans to extend the useful life of its servers from four years to five, and its networking equipment from five years to six in the future".

Which is fine! Of course it's expensive to rent your computers from someone else. But it's never presented in those terms. The cloud is sold as computing on demand, which sounds futuristic and cool, and very much not like something as mundane as "renting computers", even though that's mostly what it is.

But this isn't just about cost. It's also about what kind of internet we want to operate in the future. It strikes me as downright tragic that this decentralized wonder of the world is now largely operating on computers owned by a handful of mega corporations. If one of the primary AWS regions go down, seemingly half the internet is offline along with it. This is not what DARPA designed!

Thus I consider it a duty that we at 37signals do our part to swim against the stream. We have a business model that's incredibly compatible with owning hardware and writing it off over many years. Growth trajectories that are mostly predictable. Expert staff who might as well employ their talents operating our own machines as those belonging to Amazon or Google. And I think there are plenty of other companies in similar boats.

But before we can more broadly set sail back towards lower-cost and decentralized shores, we need to turn the rudder of our collective conversation away from the cloud-serving marketing nonsense about running your own power plant. Up until very recently, everyone ran their own servers, and much of the progress in tooling that enabled the cloud is available for your own machines as well. Don't let the entrenched cloud interests dazzle you into believing that running your own setup is too complicated. Everyone and their dog did it to get the internet off the ground, and it's only gotten easier since.

It's time to part the clouds and let the internet shine through.

While I can appreciate that it may not be a simple matter to leave the cloud, I wanted to point out, if it’s not already too late, that it might be a mistake to move from one’s own hardware into the cloud. As I noted last week, traditional data center facilities costs are not being inflated because the presence of the cloud is threatening existing old school data centers, which actively want to reduce their customer’s incentive to shutdown and move away to the cloud. there. My Level 3 data center has some massively built-out commitments of hardware by huge customers which occupy a great deal of floor space. But there are also a great many empty 19-inch racks like mine. They don’t want to lose customers like me and I don’t want to leave them.

Steven Perry 🏳️‍🌈 / @GallifreyRebel

Hi Steve, I was listening to yesterday's episode of Security Now (932) and wondered if anyone had ever shared with you and Leo a little bit of trivia about the show Lost in Space. Everyone knows and uses the catchphrase “Danger, Will Robinson” but did you know that it was only ever said once in the entire run of the show?! It was season 3, episode 11 when it happened.

It was never said again. But that is the phrase we all know and love about the show. Thought I'd pass it along. Have a good day!

That surprised me, but after a bit of confirmatory poking around the Internet, that appears to be the case. The Robot was often waving its arms around and saying "Danger! Danger!" so perhaps that helped to make that particular catchphrase stick.

I suppose, if this podcast has a similar mantra, it would be: "What could possibly go wrong?!?!"

John Carling / @JRCsystems

Hey Steve… I have a request... maybe you've already made something like this for yourself, similar to Never10 ?... But here goes… I'm using Windows 11 Pro. I swear, every time Microsoft updates Edge, it asks if it should be the default browser. AND, without asking, it overrides Adobe Acrobat as the default PDF viewer. How do we make that stop? Built-in browser/OS settings don't seem to work. Every time it updates, it kills previous settings. Is there a registry hack that could be auto-applied on every reboot to prevent Edge from stealing PDF viewing away from Acrobat, and stop itself from asking if it should be the new default browser.

So, I don't yet have that problem since no machine of mine is running Windows 11, and from everything I've heard about Windows 11, no machine of mine ever will. As we old timers know, Microsoft has this odd tendency to alternate good versions of Windows with really bad versions. So if they haven't finally completely lost the formula for Windows, perhaps Windows 12 will be useful.

But as for Edge's settings being changed, I hear TWiT's official Windows expert, Paul Thurrott, constantly complaining about the same thing, though I don't know how much attention Paul has spent attempting to override that behavior. Part of what Paul needs to do is leave things alone specifically so that he's able to observe what happens to normal Windows users.

One thought I had was that Windows does tend to store all such settings in the Registry, and coders often fail to explicitly check for access rights and read-only'ness. So if you can locate the setting that associates Edge's assigned PDF reader, or the system's PDF reader, it might be possible to set the Registry value to "read only" to have that stick the next time Edge updates.

I wanted to share John's question with our listeners in case anyone has explored how to lock down these things and has found an answer, since it's a common problem and since Microsoft appears to be growing increasingly aggressive about forcing their users away from often preferred non-Microsoft alternatives, I'll be glad to share any solution that our listeners might have discovered.

TETRA:BURST

By far the news that was most forwarded to me this past week was that the encrypted security of a globally used "secure" radio communications system whose security has been trusted and relied upon worldwide, turns out not to be as secure as everyone hoped and was led to believe. And moreover, the system's insecurity was well known and kept secret by those whose commercial interests depended upon the system being trusted – when it was not trustworthy.

Wired did a beautiful job of describing the situation in their story last week titled: "Code Kept Secret for Years Reveals Its Flaw—a Backdoor" and "A secret encryption cipher baked into radio systems used by critical infrastructure workers, police, and others around the world is finally seeing sunlight. Researchers say it isn't pretty."

I'm going to share Wired's coverage of this while liberally interjecting my own commentary. Here's what Wired described...

For more than 25 years, a technology used for critical data and voice radio communications around the world has been shrouded in secrecy to prevent anyone from closely scrutinizing its security properties for vulnerabilities. But now it's finally getting a public airing thanks to a small group of researchers in the Netherlands who got their hands on its viscera and found serious flaws, including a deliberate backdoor.

The backdoor, known for years by vendors that sold the technology but not necessarily by customers, exists in an encryption algorithm baked into radios sold for commercial use in critical infrastructure. It's used to transmit encrypted data and commands in pipelines, railways, the electric grid, mass transit, and freight trains. It would allow someone to snoop on communications to learn how a system works, then potentially send commands to the radios that could trigger blackouts, halt gas pipeline flows, or reroute trains.

Researchers found a second vulnerability in a different part of the same radio technology that is used in more specialized systems sold exclusively to police forces, prison personnel, military, intelligence agencies, and emergency services, such as the C2000 communication system used by Dutch police, fire brigades, ambulance services, and Ministry of Defense for mission- critical voice and data communications. The flaw would let someone decrypt encrypted voice and data communications and send fraudulent messages to spread misinformation or redirect personnel and forces during critical times.

Three Dutch security analysts discovered the vulnerabilities—five in total—in a European radio standard called TETRA (Terrestrial Trunked Radio), which is used in radios made by Motorola, Damm, Hytera, and others. The standard has been used in radios since the '90s, but the flaws remained unknown because encryption algorithms used in TETRA were kept secret until now.

The technology is not widely used in the US, where other radio standards are more commonly deployed. But Caleb Mathis, a consultant with Ampere Industrial Security, conducted open source research for WIRED and uncovered contracts, press releases, and other documentation showing TETRA-based radios are used in at least two dozen critical infrastructures in the US. Because TETRA is embedded in radios supplied through resellers and system integrators like PowerTrunk, it's difficult to identify who might be using them and for what. But Mathis helped WIRED identify several electric utilities, a state border control agency, an oil refinery, chemical

plants, a major mass transit system on the East Coast, three international airports that use them for communications among security and ground crew personnel, and a US Army training base.

The researchers with Midnight Blue in the Netherlands discovered the TETRA vulnerabilities – which they're calling TETRA:Burst – in 2021 but agreed not to disclose them publicly until radio manufacturers could create patches and mitigations. Not all of the issues can be fixed with a patch, however, and it's not clear which manufacturers have prepared them for customers. Motorola–one of the largest radio vendors–did not respond to repeated inquiries from WIRED.

The Dutch National Cyber Security Centre assumed the responsibility of notifying radio vendors and computer emergency response teams around the world about the problems, and of coordinating a timeframe for when the researchers should publicly disclose the issues.

And, by the way, that will be a week from tomorrow during their Blackhat presentation titled "All cops are broadcasting: breaking TETRA after decades in the shadows."

In a brief email, NCSC spokesperson Miral Scheffer called TETRA "a crucial foundation for mission-critical communication in the Netherlands and around the world" and emphasized the need for such communications to always be reliable and secure, "especially during crisis situations." She confirmed the vulnerabilities would let an attacker in the vicinity of impacted radios "intercept, manipulate or disturb" communications and said the NCSC had informed various organizations and governments, including Germany, Denmark, Belgium, and England, advising them how to proceed. A spokesperson for DHS's CISA said they are aware of the vulnerabilities but wouldn't comment further.

The researchers say anyone using radio technologies should check with their manufacturer to determine if their devices are using TETRA and what fixes or mitigations are available.

The researchers plan to present their findings at the BlackHat security conference in Las Vegas, when they will release detailed technical analysis as well as the secret TETRA encryption algorithms that have been unavailable to the public until now. They hope others with more expertise will dig into the algorithms to see if they can find other issues.

TETRA was developed in the '90s by the European Telecommunications Standards Institute, or ETSI. The standard includes four encryption algorithms—TEA1, TEA2, TEA3, and TEA4—that can be used by radio manufacturers in different products, depending on their intended use and customer.

Okay now, hold on. The four different encryption algorithms can be used by radio manufacturers in different products depending upon their intended use and customer? If that doesn't smell fishy I don't know what does. Wired continues...

***TEA1** is for commercial uses; for radios used in critical infrastructure in Europe and the rest of the world, though, it is also designed for use by public safety agencies and military, according to an ETSI document, and the researchers found police agencies that use it.*

***TEA2** is restricted for use in Europe by police, emergency services, military, and intelligence agencies.*

What?! Why? TEA1 is for commercial uses whereas TEA2 is restricted for use in Europe by police, emergency services, military and intelligence agencies? Why?

> ***TEA3*** *is available for police and emergency services outside Europe—in countries deemed "friendly" to the EU, such as Mexico and India; those not considered friendly—such as Iran—only had the option to use TEA1.*
>
> ***TEA4****, another commercial algorithm, is hardly used, the researchers say.*
>
> *The vast majority of police forces around the world, aside from the US, use TETRA-based radio technology, the researchers found, after conducting open source research. TETRA is used by police forces in Belgium and the Scandinavian countries, East European countries like Serbia, Moldova, Bulgaria, and Macedonia, as well as in the Middle East in Iran, Iraq, Lebanon, and Syria.*
>
> *Additionally, the Ministries of Defense in Bulgaria, Kazakhstan, and Syria use it. The Polish military counterintelligence agency uses it, as do the Finnish defense forces, and Lebanon and Saudi Arabia's intelligence service, to name just a few.*
>
> *Critical infrastructure in the US and other countries use TETRA for machine-to-machine communication in SCADA and other industrial control system settings—especially in widely distributed pipelines, railways, and electric grids, where wired and cellular communications may not be available.*

And now, get a load of this blast from the past:

> *Although the standard itself is publicly available for review, the encryption algorithms are only available under a signed NDA to trusted parties, such as radio manufacturers. The vendors have to include protections in their products to make it difficult for anyone to extract the algorithms and analyze them.*

Oh, boy.

> *To obtain the algorithms, the researchers purchased an off-the-shelf Motorola MTM5400 radio and spent four months locating and extracting the algorithms from the secure enclave in the radio's firmware. They had to use a number of zero-day exploits to defeat Motorola protections, which they reported to Motorola to fix. Once they reverse-engineered the algorithms, the first vulnerability they found was the backdoor in TEA1.*

So, huge props to these guys. No one made it easy for them to obtain the information they needed. In fact, their efforts were deliberately thwarted by the use of a secure enclave and they needed to find 0-day exploits in order to crack the lid off the code.

And let's also just pause for a moment to thank our lucky stars that this reverse engineering conduct has been deemed legal. If white hat hackers could be jailed for conducting research in the interest of improving the security of the products they're examining – even when doing so is not in the interest of those who are working hard to keep those secrets – the world would be far

less secure and only bad guys would be pursuing such reverse engineering.

So here's what they found...

All four TETRA encryption algorithms use 80-bit keys, which the researchers say [and I would agree] *even more than two decades after their release, still provides sufficient security to prevent someone from cracking them. But **TEA1** has a **"feature"** that reduces its encryption key length to just 32 bits, which the researchers were able to crack in less than a minute using a standard laptop and samples of just four ciphertexts.*

Brian Murgatroyd, chair of the technical body at ETSI responsible for the TETRA standard, objects to calling this a backdoor. He says when they developed the standard, they needed an algorithm for commercial use that could meet export requirements to be used outside Europe, and that in 1995 a 32-bit key still provided security, though he acknowledges that with today's computing power that's no longer the case.

*Matthew Green, a Johns Hopkins University cryptographer and professor, calls the weakened key **"a disaster."** He said: **"I wouldn't say it's equivalent to using no encryption, but it's really, really bad."***

*Gregor Leander, a professor of computer science and cryptographer with a security research team known as CASA at Ruhr University Bochum in Germany, says it would be **"stupid"** for critical infrastructure to use **TEA1**, especially without adding end-to-end encryption on top of it. He said: **"Nobody should rely on this."***

Murgatroyd insists the most anyone can do with the backdoor is decrypt and eavesdrop on data and conversations. TETRA has strong authentication, he says, that would prevent anyone from injecting false communication.

***"That's not true,"** says Wetzels. TETRA only requires that devices authenticate themselves to the network, but data and voice communications between radios are not digitally signed or otherwise authenticated. The radios and base stations trust that any device that has the proper encryption key is authenticated, so someone who can crack the key as the researchers did, can encrypt their own messages with it and send them to base stations and other radios.*

*While the **TEA1** weakness has been withheld from the public, it's apparently widely known in the industry and governments. In a 2006 US State Department cable leaked to Wikileaks, the US embassy in Rome describes an Italian radio manufacturer asking about exporting TETRA radio systems to municipal police forces in Iran. The US pushed back on the plan, so the company representative reminded the US that encryption in the TETRA-based radio system they planned to sell to Iran is **"less than 40-bits,"** implying that the US shouldn't object to the sale because the system isn't using a strong key.*

The second major vulnerability the researchers found isn't in one of the secret algorithms, but it affects all of them. The issue lies in the standard itself and how TETRA handles time syncing and keystream generation.

When a TETRA radio contacts a base station, they initiate communication with a time sync. The network broadcasts the time, and the radio establishes that it's in sync. Then they both generate the same keystream, which is tied to that timestamp, to encrypt the subsequent communication.

Wetzels says: "The problem is that the network broadcasts the time in packets that are unauthenticated and unencrypted."

As a result, an attacker can use a simple device [probably what Leo carries around in his pocket] *to intercept and collect encrypted communication passing between a radio and base station, while noting the timestamp that initiated the communication. Then he can use a rogue base station to contact the same radio or a different one in the same network and broadcast the time that matches the time associated with the intercepted communication. The radio is dumb and believes the correct time is whatever a base station says it is. So it will generate the keystream that was used at that time to encrypt the communication the attacker collected. The attacker recovers that keystream and can use it to decrypt the communication collected earlier.*

To inject false messages, he would use his base station to tell a radio that the time is tomorrow noon and ask the radio to generate the keystream associated with that future time. Once the attacker has it, he can use the keystream to encrypt his rogue messages, and the next day at noon send them to a target radio using the correct keystream for that time.

Wetzels imagines Mexican drug cartels could use this to intercept police communications to eavesdrop on investigations and operations or deceive police with false messages sent to radios. The attacker needs to be near a target radio, but the proximity is only dependent on the strength of the rogue base station's signal and the terrain.

"You can do this within a distance of tens of meters," he says. The rogue base station would cost $5,000 or less to build.

ETSI's Murgatroyd downplays the attack saying TETRA's strong authentication requirements would prevent a non-authenticated base station from injecting messages. Wetzel disagrees, saying TETRA only requires devices to authenticate to the network, not to each other.

The researchers didn't find any weaknesses in the TEA2 algorithm used by police, military, and emergency services in Europe, but they did initially think they found another backdoor in TEA3. Given that TEA3 is the exportable version of TEA2, there was good reason to believe it might also have a backdoor to meet export requirements.

They thought they found something suspicious in a substitution box, or S-box, used in the algorithm, which contains a bad property they say would "never appear in serious cryptography." The researchers didn't have sufficient skill to examine it to determine if it was exploitable. But Leander's team did examine it and he says it's not.

Leander said: "In many ciphers, if you used such a box it would break the cipher very badly. But the way it's used in TEA3, we couldn't see that this is exploitable." This doesn't mean someone else might not find something in it he says, but he'd "be very surprised if it leads to an attack that's practical."

With regard to fixes for the other problems the researchers found, Murgatroyd says ETSI fixed the keystream/timestamp issue in a revised TETRA standard published last October, and they created three additional algorithms for vendors to use, including one that replaces TEA1.

Vendors have created firmware updates that fix the keystream/timestamp issue. But the problem with TEA1 cannot be fixed with an update. The only solution for that is to use another

algorithm—not an easy thing to switch—or to add end-to-end encryption on top of TETRA, something Wetzels says is impractical. It's very expensive since the encryption has to be applied to every device, it requires some downtime to do the upgrade—something not always feasible for critical infrastructure—and can create incompatibility issues with other components.

As for asking their vendor to switch out TEA1 for one of the new algorithms meant to replace it, Wetzels says this is problematic as well, since ETSI plans to keep those algorithms secret, like the others, asking users to trust again that the algorithms have no critical weakness.

Wetzels says "There's a very high chance that [the replacement algorithm for TEA1] will be weakened as well."

The researchers don't know if the vulnerabilities they found are being actively exploited. But they did find evidence in the Edward Snowden leaks that indicate the US National Security Agency (NSA) and UK's GCHQ intelligence agency targeted TETRA for eavesdropping in the past. One document discusses an NSA and Australian Signals Directorate project to collect Malaysian police communications during a climate change conference in Bali in 2007 and mentions that they obtained some TETRA collections on Indonesian security forces' communications.

Another Snowden leak describes GCHQ, possibly with NSA assistance, collecting TETRA communications in Argentina in 2010 when tensions rose between it and the UK over oil exploration rights in a deep-sea oil field off the coast of the Falkland Islands. It describes an operation to collect high-priority military and leadership communications of Argentina and reveals that the project resulted in successful TETRA collections.

"This doesn't indicate they exploited these vulnerabilities that we found," Wetzels says. "But it does show … that state-sponsored actors are actively looking at and collecting these TETRA networks, even in the early 2000s."

So, as I and Wired noted, in eight days all of the wraps will be coming off of this when the research team presents their work and findings during Blackhat in Las Vegas.

With TETRA we have a legacy, encrypted, radio communications system being widely used today throughout the entire world, including in the US. And it not only contained multiple readily exploitable flaws that were only fixed after security researchers cracked it open and shamed its creators with the threat of disclosure, it also contained deliberately weakened encryption which most of the world was given to use while some agencies knew of the weakness and were apparently leveraging that knowledge for eavesdropping.

And now we learn that the ETSI group who did all of this has replaced their earlier flawed work with more of the same. Keeping their encryption secret (because they believe they can) and saying "just trust us, we got it right this time." Why would anyone trust them ever again?