Security Now! #932 - 07-25-23

Satellite Insecurity, Part 2

This week on Security Now!

What did Apple recently say to the UK? What's Google's "Web Environment Integrity" and why's it so controversial? Who's the latest to express unhappiness over Google Analytics? What happy news did the UK deliver about IoT security that the U.S. not done so far? Might you be qualified to join the U.S.'s forthcoming Expeditionary Cyber Force? What's the latest on ransomware attack payouts and also on the Massive MOVEit maelstrom? And who's the most recent major player to announce the adoption of Passkeys? Once we all have the answers to those questions, we've going to spend some time with our faithful listeners, then wrap up this Part 2 of our look at the current and quite distressing state of satellite insecurity.

Why "Reading the Manual" is always a good idea...

Security News

R.I.P. Kevin

First, a bit of sad news. The wider world received the news at the end of last week, that the famous and long-since reformed hacker Kevin Mitnick had quietly passed away the previous Sunday, on July 16th, just three weeks shy of his 60th birthday. After more than a year of struggle, Kevin lost his battle with pancreatic cancer.

And, Leo, back in the TechTV ScreenSavers days, you had Kevin on your show a number of times, often with Steve Wozniak, who was also one of Kevin's close friends.

Apple says: "Thanks, but we'd rather leave."

Last Thursday BBC News carried a story under the headline *"Apple slams UK surveillance-bill proposals"* but the first line of their piece was a showstopper. It read: *"Apple says it will remove services such as FaceTime and iMessage from the UK rather than weaken security if new proposals are made law and acted upon."*

As we know, since we've been tracking this super-engaging struggle between the commercial forces of absolute privacy and those in governments who are wishing to make privacy conditional, the UK is seeking to update the Investigatory Powers Act (IPA) 2016. It wants to require messaging services to clear their security features with the Home Office before releasing them to customers. The act lets the Home Office demand that security features are disabled, without telling the public. And under this forthcoming update, this would have to be immediate.

WhatsApp, Signal and all the others have previously expressed their strongest possible opposition to this, with Signal making what has been up until now the strongest public statement, stating that they will simply "walk" from the UK. Apple has been in opposition to this, too, but until now hasn't drawn any such sharp line in the sand. That's what just happened.

The UK government has opened an eight-week consultation on the proposed amendments to the IPA. The government is claiming that they are *"not seeking to create new powers"* but only to make the Act more relevant to the current technology. Uh huh.

Apple has submitted its formal 9-page response to the now-open consultation. Apple formally opposes:

- Having to tell the Home Office of any changes to product security features before they are released.
- The requirement for non-UK-based companies to comply with changes that would affect their product globally - such as providing a backdoor to end-to-end encryption.
- Having to take action immediately if a notice to disable or block a feature is received from the Home Office, rather than waiting until after the demand has been reviewed or appealed.

Apple says:

- It would not make changes to security features specifically for one country that would weaken a product for all users.

- Some changes would require issuing a software update so could not be made secretly.
- The proposals "constitute a serious and direct threat to data security and information privacy" that would affect people outside the UK.

Remember that what the various governments are asking for is **not** simply the ability for these various encrypted services to respond to court ordered surveillance – that's an entirely different "ask". What the governments are seeking now is universal surveillance of all communications of all kinds for all of their citizens. And that **is** new. That's not an update of anything that exists today.

The BBC quoted a cyber-security expert, Professor Alan Woodward, from Surrey University, who said that technology companies were *"unlikely to accept the proposals"*. He said: *"There is a degree of arrogance and ignorance from the government if they believe some of the larger tech companies will comply with the new requirements without a major fight."* I think that Signal and Apple have been quite clear that they have no interest in, or need, to fight. In order to avoid breaking any newly enacted legislation, they will simply pull their services from those regions which enact laws that seek to violate the privacy of their users. Period. Nothing to fight about here. Then we'll see what the voters in those areas think. And we'll also see how the bureaucrats, law enforcement and intelligence services like not having any secure messaging services available to them in support their own needs for privacy. What's good for the goose...

The Home Office told the BBC that the Investigatory Powers Act was designed to *"protect the public from criminals, child sex abusers and terrorists"*. That's obviously an honorable goal. But the price for doing so is too high.

Web Environment Integrity

Four Google engineers have put forth a proposal that immediately generated a huge backlash across the web developer community. Despite the fact, and in some cases perhaps due to the fact, that this proposal was dropped on Github as one of the engineer's personal projects, not from Google, many Google skeptics see this as Google's backdoored means of sliding this quietly into the stream. But as I said, it quickly hit everyone's radar.

While the developers termed this proposal *"Web Environment Integrity"* the industry quickly slapped it with the term *"Web DRM"* and noted that it would instantly provide a means for websites to refuse to offer their content to any browser running an ad blocker or to disable ad blockers remotely. And given that Google's revenue stream is largely advertising, the fact that this new web standard was proposed *"off the books"* by four web developers who all just happen to be employed by Google... well... one could be forgiven for questioning or at least wondering about true motives.

And essentially it does, indeed, amount to Web DRM – a means for enforcing the display of exactly what any website wishes to display by empowering websites to selectively remove all user freedom at their web client end to alter the website's display in any way.

This is not to say that there could not also be significant upside user benefits. For example, allowing a banking website to rigorously control what (if any) 3rd-party extensions are enabled

when a user visits, essentially locking the web browser client in order to enhance the visit's security, would be a good thing. But it's equally obvious how taking this control away from users could be abused by allowing any website to decide, on behalf of their visitors, what browser environments are acceptable.

The engineer/authors start off their description of Web Environment Integrity by explaining:

> *Users often depend on websites trusting the client environment they run in. This trust may assume that the client environment is honest about certain aspects of itself, keeps user data and intellectual property secure, and is transparent about whether or not a human is using it. This trust is the backbone of the open internet, critical for the safety of user data and for the sustainability of the website's business.*
>
> *Some examples of scenarios where users depend on client trust include:*
>
> - *Users like visiting websites that are expensive to create and maintain, but they often want or need to do it without paying directly. These websites fund themselves with ads, but the advertisers can only afford to pay for humans to see the ads, rather than robots. This creates a need for human users to prove to websites that they're human, sometimes through tasks like challenges or logins.*
> - *Users want to know they are interacting with real people on social websites but bad actors often want to promote posts with fake engagement (for example, to promote products, or make a news story seem more important). Websites can only show users what content is popular with real people if websites are able to know the difference between a trusted and untrusted environment.*
> - *Users playing a game on a website want to know whether other players are using software that enforces the game's rules.*
> - *Users sometimes get tricked into installing malicious software that imitates software like their banking apps, to steal from those users. The bank's internet interface could protect those users if it could establish that the requests it's getting actually come from the bank's or other trustworthy software.*

Whether or not this proposal ever advances past the controversy created by its appearance, it does point to a tension that appears to be developing: Should websites be able to reach across the Internet and exert full control over the experiences of their viewers? When we run a native app on our local computer we have limited control over what it does and how it works. We can launch it and terminate it. But that's about it. It's not difficult to imagine that many websites would like to enforce that same level of control.

https://github.com/RupertBenWiser/Web-Environment-Integrity/blob/main/explainer.md

I put a link in the show notes for anyone who might be interested in digging deeper into this specific proposal.

Web Analytics under the spotlight

We've noted a number of times that various EU countries have been complaining, and have even taken to suing organizations within their borders, who are continuing to use Google Analytics which, they state, potentially transfers private identifiable data outside of their borders. But now this concern has come home to roost with a letter the Federal Trade Commission (our FTC) and the U.S. Department of Health and Human Services (HHS) have sent to 130 hospital systems and telehealth providers warning them about their obligations to protect their client's personal health information. Listen to this:

> The Office for Civil Rights (OCR) at the U.S. Department of Health and Human Services (HHS) and the Federal Trade Commission (FTC) are writing to draw your attention to serious privacy and security risks related to the use of online tracking technologies that may be present on your website or mobile application (app) and impermissibly disclosing consumers' sensitive personal health information to third parties.
>
> Recent research, news reports, FTC enforcement actions, and an OCR bulletin have highlighted risks and concerns about the use of technologies, such as the Meta/Facebook pixel and Google Analytics, that can track a user's online activities. These tracking technologies gather identifiable information about users as they interact with a website or mobile app, often in ways which are not avoidable by and largely unknown to users.
>
> Impermissible disclosures of an individual's personal health information to third parties may result in a wide range of harms to an individual or others. Such disclosures can reveal sensitive information including health conditions, diagnoses, medications, medical treatments, frequency of visits to health care professionals, where an individual seeks medical treatment, and more. In addition, impermissible disclosures of personal health information may result in identity theft, financial loss, discrimination, stigma, mental anguish, or other serious negative consequences to the reputation, health, or physical safety of the individual or to others. Health Insurance Portability and Accountability Act of 1996 (HIPAA)
>
> If you are a covered entity or business associate under HIPAA, you must comply with the HIPAA Privacy, Security, and Breach Notification Rules, with regard to protected health information (PHI) that is transmitted or maintained in electronic or any other form or medium.
>
> The HIPAA Rules apply when the information that a regulated entity collects through tracking technologies or discloses to third parties (e.g., tracking technology vendors) includes PHI. HIPAA regulated entities are not permitted to use tracking technologies in a manner that would result in impermissible disclosures of PHI to third parties or any other violations of the HIPAA Rules. OCR's December 2022 bulletin about the use of online tracking technologies by HIPAA regulated entities provides a general overview of how the HIPAA Rules apply. This bulletin discusses what tracking technologies are and reminds regulated entities of their obligations to comply with the HIPAA Rules when using tracking technologies.
>
> To the extent you are using the tracking technologies described in this letter on your website or app, we strongly encourage you to review the laws cited in this letter and take actions to protect the privacy and security of individuals' health information.

While this is not the same "thou shall not use" commandment that EU countries are issuing, Google has been Analyticizing for the past 17 years, since 2005 and only now does it appear that people are beginning to say “hey, hold on there a second…”

More progress on the IoT security front

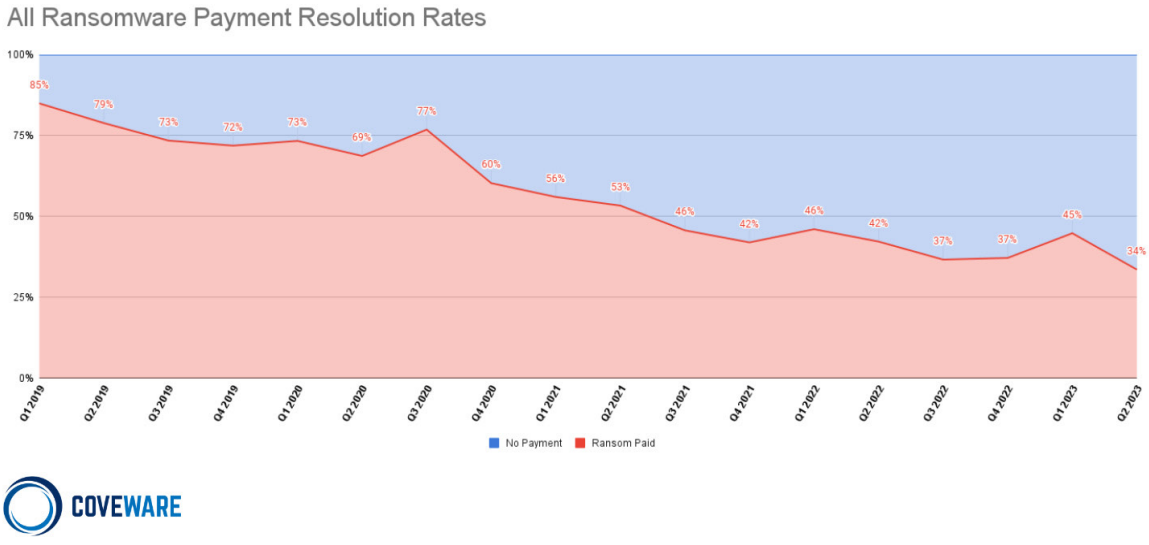
The European Union has just approved a draft version of what they are calling their “Cyber Resilience Act.” It's a set of new cybersecurity-related rules for IoT devices. The act passed the EU's Industry, Research, and Energy Committee with 61 votes in favor, one against, and 10 abstentions. Under the new regulations, vendors must ensure their products meet a certain set of criteria before being sold in the Eurozone. Products will have to come with automatic security updates as the default option, must ensure data confidentiality using encryption, and vendors must inform authorities of any attacks. And the new rules are expected to enter into effect by next year.

This is great news for the consumer since products sold globally, including in the Eurozone, would need to be in compliance. So, for example, U.S. consumers would reap the benefits. And in this case the EU is ahead of the US since all we've managed to get done here so far is to design an attractive shield emblem that will be placed on any devices that are compliant with a set of standards that don't yet exist. But hey, at least we have a pretty shield.

The “Expeditionary cyber force”

We know you listen to this podcast, so you already have some qualifications. Do you like to travel? See faraway places and wonder what the people there are saying? Enjoy wearing ridiculous cammo when sitting in front of a computer? Well, you may be just what the U.S. is looking for! Lieutenant General Timothy Haugh, the nominee to become the next head of the NSA and CyberCom, has pledged to create “expeditionary cyber forces” that can be deployed into far off lands to reach important tactical targets in forward locations. So get ready to pack up your laptop and head out!

Ransomware payouts being made much less often



COVEWARE

A chart in the show notes depicts the more or less steady drop in the percentage of ransomware attacks which result in ransom payments. When Coveware began tracking ransom payment rates at the start of 2019, 85% of ransomware attacks resulted in payments. Today, that number has hit an all time low of just 34%.

Coveware's report which was published Friday was titled: *"Ransom Monetization Rates Fall to Record Low Despite Jump In Average Ransom Payment"*. So the news is not all good, but it's great that today only one out of every three attacks results in payment. That sure beats 85% four years ago. The first three sentences of their report read:

> *In the second quarter of 2023, the percentage of ransomware attacks that resulted in the victim paying, fell to a record low of 34%. The trend represents the compounding effects that we have noted previously of companies continuing to invest in security, continuity assets, and incident response training. Despite these encouraging statistics, ransomware threat actors and the entire cyber extortion economy, continue to evolve their attack and extortion tactics.*

MOVEit Update

Emsisoft reports that the total number of confirmed victims of the Progress Software MOVEit Transfer SQL injection attacks has now surpassed 380. And Coveware expects the Russian Cl0p gang behind the attacks to receive between $75 and $100 million dollars in total.

TikTok + Passkeys

So who's the latest major player to be adding support for Passkeys? Would you believe TikTok?! Yep. Last week TikTok posted:

> *"We will begin rolling out passkeys for iOS in certain regions, starting with Asia, Africa, Australia, and South America beginning this month, and anticipate expanding in geographies and operating systems over time."*

And TikTok has also become a member of the FIDO Alliance.

Closing the Loop

Allan E / @resmandinga

> *I agree that I would never save 2FA seeds in my password manager. But it may be the least bad option for protecting shared business accounts on social media accounts in some cases.*

I didn't intend to suggest that there was no justifiable use case for having a password manager store TOTP secrets. The question was just a perfect opportunity to highlight and talk about a clear example of the trade off between user convenience and security.

Steven Haver / @TeachFromLove

Re: Bitwarden TOTP. It's actually a huge increase in security for people who otherwise can't be bothered to turn on 2FA. It's also extremely useful with shared logins that are shared with multiple people via a Bitwarden Organization. But for more tech savvy people, I understand why you would want the greater compartmentalization. I run a hybrid approach where my "less important" TOTPs are in Bitwarden, the more important ones are in OTP Auth, and the most important ones are Webauthn on my security keys. In a way, TOTP seems like a dying format for those who already use a security key, as FIDO/U2F/Webauthn become available on more on more sites. Whereas once there were 40 secrets in my OTP Auth, I'm down to just a handful these days. Thanks for a great podcast (and super excited to take 6.1 for a spin soon). Steven

Of course I agree with everything that Steven has said. And his hybrid approach makes sense. I suspect that for most people, just using Bitwarden will be the way to go. But, again, my point was to use this more as an example of the tradeoff between convenience and security.

Sakis Kasampalis / @SKasampalis

Hey Steve. How exactly is Threads blocking Europeans using VPNs? I thought that the idea of a VPN was that they cannot tell where you are located. Are they blacklisting IPs of the popular third-party VPNs? What about self-hosted ones?

So that question has multiple parts. The first part is that there are two ends to every connection and every end inherently knows the address and the rough location of the other end. So when someone in the EU connects to Meta directly, Meta gets their IP address and can choose to refuse it.

What clearest way to visualize that a VPN does is to see it as two connections: The user's connection to the VPN service and the VPN service's connection to the destination. So when a customer is connecting through a VPN, Meta doesn't see the customer's IP and their rough location. Meta is being connected to by the VPN, so that's the only IP and location that Meta sees. This bring us to the second part of Sakis' question, which was "*Are they blacklisting IPs of the popular third-party VPNs?*" And the answer to that is "Yes, that's definitely one way to do what they are doing." It might also be that in the interest of preserving their user's privacy, VPNs might also be deliberately stripping out some user tagging information that a user's web browser would normally provide. So Meta might be detecting the presence of a "middleman" in the connection through other means. But either way, Meta can simply decide not to honor "indirect connections" through VPNs specifically because they can be used to mask the user's true location.

And finally, as for self-hosted VPNs, the question is where the VPN's traffic would emerge onto the Internet? Self-hosting sort of suggests that the endpoint is still located locally. But then its IP would be geolocated and blocked. So it would be necessary to self-host a VPN in such a way that the VPN's traffic emerged onto the Internet from a non-blocked region. That might be doable by spinning up an AWS or Azure instance, but seems like a lot of trouble to go through just to obtain foreign access to Threads, whose popularity, by the way, appears to have collapsed overnight.

Leo: On Sunday's TWiT show you and your two guests talked about the collapse of Threads traffic. One of the guests noted how easy Meta had made it for Instagram users to join Threads. Even I joined Threads because I have a stagnant Instagram account and I wanted to grab my handle just in case Threads might amount to something someday. But I wanted to note that Threads' apparent overnight success was always entirely illusory, because when it's made that easy to join, joining doesn't actually mean anything. It's reminiscent of the news website paywall model. Originally, all sites were free and ad supported. Then some of them thought "Hey, look at all the traffic we have! Let's charge a little bit of money for people coming!" And mostly people said "Wait, what? You actually want money? I think I'll find the same news elsewhere, thanks very much."

It's going to be very interesting to see how Meta's Threads does in the longer term. And that's really the only metric that matters. I can't think of any better competitor to Twitter than Meta. And Elon's conduct with Twitter really does suggest that Twitter needs some competition to get it to see reality. And I agree with you and your Sunday guests: Elon should sell it and move on. He's really not the proper custodian for what could still be the world's preeminent instant messaging platform.

Matthew N. Dudek / @mndudek

Hi Steve, I'm looking into getting some wireless keyboards for the office, and was concerned about the security of the connection between the keyboard and the dongle (not bluetooth, one like the Logitech K400+). Have you found any info on this, and if MITM attacks are a problem for these kinds of devices? What about the security of bluetooth keyboards? Are they any better?

Many years ago we talked about the early widely available wireless keyboards which claimed to be offering "encryption", but that encryption turned out to amount to XORing the byte that the keyboard sent with a static value. At best we'd call that obfuscation, since passively recording the use of the keyboard and performing an analysis of the frequency of the characters seen, would quickly reveal the fixed XOR mask. At that point everything typed could be unscrambled and anything desired could be injected.

The keyboard in question uses Logitech's own "Unifying Receiver" technology. It's not horrible security inasmuch as it uses AES encryption in CTR mode. Unfortunately, they tried to do it on the cheap and a security review of the technology four years ago resulted in CVE-2019-13053. And **that** CVE was the result of an incomplete fix for CVE-2016-10761 three years before that. Logitech has publicly stated that they feel it's good enough and that they will not be changing anything. And, of course, at this late date, changing anything would be quite "disunifying".

From a quick look at the current state of Logitech's technology it appears that allowing an attacker to press a few keys on the keyboard while sniffing its transmission is all that's needed. Also, the protocol leaks metadata for things like turning the NUM LOCK and CAPS LOCK lights and other functions. This allows for entirely passive attacks. For AES in CTR mode to be used securely, the counter's values can never be reused under the same initialization vector (IV). But

enforcing that guarantee is difficult for any bare bones protocol, which is what Logitech created for their mice, keyboards, pointers and other peripherals.

The solution is simple: Where true security is important, just use the full Bluetooth protocol, though such a keyboard might be more expensive. All of my keyboards are wired, but my wife uses a Logitech MX Keys keyboard. It's a lovely low-profile keyboard which uses a full Bluetooth low-energy link. Once it was paired to her Windows 10 machine she's never had a problem with it.

Glenn Lau / @glennnlau

Is it possible to Spinrite a phone (iOS/ Android) to speed up the phone?

Unfortunately, I'm pretty certain that would not work. While it would be possible to plug the phone into a PC to view it as a drive, only the user-facing storage would be seen, not the underlying hidden and protected kernel and apps, which is really what you'd want to be rewriting.

Jorge Moran / @0xjams

Hi Steve, I'm a big fan. I've been listening to Security Now for years, I was wondering. A couple of weeks ago when you talked about your SyncThing setup you said you don't like containers. Is it just because of the added complexity or do you have more reasons?

It's only personal preference. I totally get it that there's a place for containers like Docker. I agree that they are a terrific solution for many applications. But for myself, I've often seen how quickly things can get out of control when the approach I would characterize as "just throw some more code at it" is taken. So, if I needed to run SyncThing on Synology and the only way to do that was to be containerized, then that's what I'd do. But it just feels much better to be running SyncThing as a native Synology build.

This reminds me of another aspect of a story I've shared before, of how when I attended that DigiCert customer advisory meeting in Utah nearly six years ago, I casually mentioned the rack of equipment I had at the Level 3 data center. All of the guys around the table looked at me like I had two heads; so I said "What??" and one of them, who was clearly speaking for all, since the rest of them were nodding their heads, said: "Steve, no one does hardware anymore."

I took that to mean that they had all moved all of their infrastructure to the cloud and were now paying Amazon or Microsoft or whomever for virtually hosting their entire infrastructures. But I also noted that everyone but I worked for a major corporation and that none of them, but I, were paying the bills for their infrastructures. It's true that I do occasionally need to drive over to Level 3 to exchange a dead SSD or a spinning drive that's died in a RAID. But in return for that, which I also enjoy since I get to touch hardware, my infrastructure costs are fixed and very low. I own all of the hardware. So I'm renting space, cooling, bandwidth and power... and these days that doesn't cost very much because Level 3 actively wants to keep me from virtualizing my infrastructure with AWS or Azure. (I don't tell them this, but they have nothing to worry

about!)

When Jorge ended his Tweet asking: "*Is it just because of the added complexity or do you have more reasons?*" My first thought was, "Hey... I even avoid compilers whenever possible!"

brianweeden / @brianweeden

> *Steve, loved the show this week on satellites. I work in the space sector on this issue. As you're prepping next week, I can offer up an open source report that my org puts out which includes an entire chapter on cyber attacks on satellites: https://swfound.org/counterspace. Looking forward to next week's part 2!*

I followed the link that Brian provided and since it's exactly on point for today, I'll share the report's introductory paragraph, which introduces the term "counterspace", and reads:

> *Space security has become an increasingly salient policy issue. Over the last several years, there has been growing concern from multiple governments over the reliance on vulnerable space capabilities for national security, and the corresponding proliferation of offensive counterspace capabilities that could be used to disrupt, deny, degrade, or destroy space systems. This in turn has led to increased rhetoric from some countries about the need to prepare for future conflicts on Earth to extend into space, and calls from some corners to increase the development of offensive counterspace capabilities and put in place more aggressive policies and postures.*
>
> *We feel strongly that a more open and public debate on these issues is urgently needed. Space is not the sole domain of militaries and intelligence services. Our global society and economy is increasingly dependent on space capabilities, and a future conflict in space could have massive, long-term negative repercussions that are felt here on Earth. Even testing of these capabilities could have long-lasting negative repercussions for the space environment, and all who operate there. The public should be as aware of the developing threats and risks of different policy options as would be the case for other national security issues in the air, land, and sea domains.*
>
> *The 2023 edition of the report assesses the current and near-term future capabilities for each country, along with their potential military utility. Countries covered in this report are divided up into those who have conducted debris-causing anti-satellite tests (the United States, Russia, China, India) and those who are developing counterspace technologies (Australia, France, Japan, Iran, North Korea, South Korea, and the United Kingdom). It covers events and activities through February 2023.*

SpinRite

No big SpinRite news this week. At the start of the work to update SpinRite, I created that new USB drive setup capability since I knew that was going to be needed. The interim spin-off from that work was GRC's InitDisk utility. So I'm currently working to merge that InitDisk technology into SpinRite's Windows app. Once that appears to be finished I'll ask the GRC newsgroup gang to test it and find anything I've overlooked, then the DOS SpinRite executable and the Windows setup utility will be merged and we'll have the first fully functional SpinRite v6.1.

Satellite Insecurity, Part 2

During last week's part 1 coverage of our 2-part look at Satellite Insecurity, we laid the groundwork to develop an appreciation for just how dependent upon that unseen orbiting infrastructure above us we have gradually become. I'll confess that I rolled my eyes when our previous US president, Donald Trump, announced the creation of "Space Force", a new branch of the US military intended to focus upon what happens above our heads. My eye rolling was mostly due to a lack of appreciation for what is now an obvious need. Satellites are uniquely vulnerable to many forms of attack – and both physical and cyber attacks are actually happening! Last week we learned that ground-based missiles are capable of destroying satellites from the ground and that space borne robot satellites capable of both repairing friendly satellites and deliberately damaging hostile satellites are not science fiction. They exist, too.

So it was against this backdrop that All of this was triggered by the recent publication of a research paper which demonstrated that those satellites orbiting above are also disturbingly vulnerable to ground-based cyber attack, which is our focus today. The short news blurb which initially caught my eye said:

> *Satellite security decades behind: A team of academics from Germany has analyzed the firmware of three low-earth orbit satellite models and found satellite security practices* ***lagging by decades*** *compared to modern laptops and mobile devices. Researchers found the firmware to be prone to several types of vulnerabilities, lacking basic protection features such as encryption and authentication. The researchers claim they devised attacks that could hijack satellite systems, cut satellites off from their ground stations, move satellites to new areas, and even crash them into the ground or into other space objects.*

As I mentioned last week, the researchers assembled their research into a paper titled: "Space Odyssey: An Experimental Software Security Analysis of Satellites." This research was delivered during the recent 44th IEEE Symposium on Security and Privacy held two months ago in May. Where it was awarded a Distinguished Paper Award for the conference.

Here's what the team described of their finding in their paper's Abstract:

> *Satellites are an essential aspect of our modern society and have contributed significantly to the way we live today, most notable through modern telecommunications, global positioning, and Earth observation. In recent years, and especially in the wake of the New Space Era, the number of satellite deployments has seen explosive growth. Despite its critical importance, little academic research has been conducted on satellite security and, in particular, on the security of onboard firmware. This lack likely stems from outdated assumptions on achieving security by obscurity, effectively preventing meaningful research on satellite firmware.*
>
> *In this paper, we first provide a taxonomy of threats against satellite firmware. We then conduct an experimental security analysis of three real-world satellite firmware images. We base our analysis on a set of real-world attacker models and find* ***several security-critical vulnerabilities in all analyzed firmware images.*** *The results of our experimental security assessment show that modern in-orbit satellites suffer from different software security*

vulnerabilities and often a lack of proper access protection mechanisms. They also underline the need to overcome prevailing but obsolete assumptions. To substantiate our observations, we also performed a survey of 19 professional satellite developers to obtain a comprehensive picture of the satellite security landscape.

In other words, after this team of six researchers had uncovered what they thought they had uncovered, they wondered whether things could possibly be that bad, so they surveyed 19 satellite developers to confirm that they were seeing things correctly.

The researchers begin by explaining a bit of the history of the industry. I want to share this since it will be so entirely believable and even understandable – though also so obviously wrong – to this podcast's audience. They write:

Satellites are sophisticated technical devices that are placed in outer space for research purposes or to provide terrestrial applications with services that leverage the coverage of the Earth's surface [from a distance]. While the first satellite, Sputnik, dates back to 1957, we are in the midst of a renaissance of spaceflight referred to as the New Space Era. Especially in recent years, we have observed an enormous growth in the number of earth-orbiting satellites. According to the United Nations Office for Outer Space Affairs (UNOOSA), the number of satellites has nearly doubled from 4, 867 in 2019 to 9, 350 in 2022. The majority of these satellites form mega-constellations like Starlink, which plans to launch more than 40,000 satellites in the coming years.

Small satellites are at the heart of this New Space Era as their size and the widespread use of Commercial off-the-shelf (COTS) components makes them affordable even for small institutions. Furthermore, they cover a broad spectrum of use cases ranging from commercial applications (like Earth observation, machine-to-machine communication, and Internet services) to research applications, such as technology testing, weather and earthquake forecasting, and even interplanetary missions.

Although their applications vary widely, small satellites commonly consist of radio equipment and microcontroller boards. Hence—in the broadest sense—they are computer systems connected to a ground station on Earth and, sometimes, even to other satellites. Because they rely on wireless connections for command and control and use microcontrollers, they are potentially as vulnerable to attacks as any other connected IT platform on Earth.

This issue has not been very relevant in the past, since access to ground stations was expensive and limited to large satellite operators. However, the situation changed fundamentally in recent years. Nowadays, ground stations are even affordable for private individuals and with the emergence of Ground Station as a Service (GSaaS) models, such as those offered by Amazon Web Services and Microsoft Azure, the entry barrier becomes even lower. We have seen in the mobile network security domain how the providers' assumption that the radio equipment required for attacks would be too costly and out of reach for attackers was ultimately disproved by technological advances. [Right! The Pineapple and that thing Leo has in his pocket!] *So, affordable ground stations create a novel attack surface, where adversaries can communicate with satellites and take advantage of software vulnerabilities. If they successfully compromise the satellite's firmware, they can access the satellite and potentially take complete control of the system.*

Despite warnings being made early, little has been done to address this problem for several

reasons. While the lack of security standards for satellites and the complex supply chain complicate the situation, the main reason is the inaccessibility of satellite firmware.

Historically, satellite developers have relied on [oh yes!] *security by obscurity. The developers of the Iridium network even mentioned that their system would be too complex for attackers. But attackers have nevertheless successfully decrypted the communication of the network.*

The inaccessibility of satellites in orbit makes dumping of the firmware by researchers very challenging (if not impossible), impeding progress in this area. Hence, the developers of satellite firmware act as gatekeepers and do not provide researchers with research subjects. Previous commentators have acknowledge that the topic is still understudied and conclude that collaboration between satellite development and the security field is required. Additionally, well-known topics like the security of satellite communication, the security of satellite-based Internet services, and threat scenarios for satellites have recently gained increasing attention. However, discussions around individual satellites typically lack technical details of satellite and real-world foundations due to the inaccessibility of satellite software.

Okay. So we have a situation were the physical isolation that's inherent in anything launched into orbit has supported a laxity of security rigor. And it also really sounds as though the developers of these systems have not been following along with the startling advances being made in the capabilities of the underground hacking community. As we've seen time and time again, if money can be made through some hack or attack, it's going to happen and those attacks are only going to be improving over time. It's a very good thing that Bitcoin was not a satellite-based cryptocurrency, or there wouldn't be any satellites left in orbit today!

But in all seriousness, the US, China and Russia don't care about the price of Bitcoin. What they want is the ability to instantly cripple each other's above-Earth command and control infrastructure if the you-know-what suddenly hits and fan.

These researchers felt that they were able to significantly contribute to an understanding of satellite-based security in three ways. They wrote:

First, we present a taxonomy of threats against onboard satellite firmware. Such a systematic review of the attack surfaces allows us to better represent the complex nature of satellites and categorize security-relevant findings throughout the paper.

Second, we conduct an experimental and comprehensive security analysis of ***three real-world, in-orbit satellites*** *to better understand the attack surface and the current state of software security in this particular domain. We focus on Low Earth Orbit (LEO) satellites, as this orbit is the main focus of the New Space Era.*

The most prevalent satellite class is the nanosatellite, more specifically, the CubeSat which is a standard form factor of 10 cm cubes (called Units or U – that's 4 inches on a side!). These satellites typically weigh less than 1.33 kg per U and are used in many different projects. After a long period of persuasion, trust building, discussions, and contracts, we obtained access to several satellite firmware images that we were able to analyze. As a result of our security assessment, we found six different kinds of security vulnerabilities in recently launched modern spacecraft, including unprotected telecommand interfaces. All vulnerabilities have been responsibly disclosed to the vendors. Note that the entry barrier to identify these

vulnerabilities was complex, given the sensitive nature of these systems. To the best of our knowledge, our work is the first to demonstrate exploitation of satellite firmware vulnerabilities allowing attackers to gain persistent control over the satellite.

Third, we conducted a survey of 19 professional satellite engineers and developers to broaden the scope of our research. In total, the responses cover technical information on ***17 satellites****, and the participants worked on a total of* ***132 satellites****.*

Our survey reveals that protocol obscurity is as prevalent as encryption for access protection and that small development teams are rather inclined to develop full custom protocols instead of using existing ones. As one of our survey participants mentioned: "We focused on providing a functioning system instead of a secure one".

Oh, goodness.

Thankfully, satellite communications is not entirely a standards-free roll your own environment. There is a standards body known as the CCSDS for Consultative Committee for Space Data Systems (CCSDS). It's a consortium of numerous space agencies that agree on the standards which will be used for a satellite's communications. So the CCSDS provides a wealth of protocol standards for communicating with all components and parties involved in spacecraft operations. These standards cover all layers of the OSI networking model, usually offering several options per layer.

Two protocols that were examined by these researchers. The higher level protocol is SDLS for Space Data Link Security. It provides the data link layer. And the lower level network protocol is SPP, the Space Packet Protocol.

Their paper then delves into the detailed intercommunications among the various satellite components. The attacker's goals are no different in the sky than on the ground. They would love to take over the entire package if they could. But failing that, being able to tap into the communications flow might be all that's available. And if even that is out of research, then denying the services provided by the satellite to its rightful users is the final fallback.

The researchers explain that the information containment that has historically existed until recently has been crumbling with the many recent changes taking place within the satellite industry. They wrote:

For decades, the satellite community and developers have acted as gatekeepers for the topic of satellite security. By keeping the software and components of satellites under lock, they created a "barrier of obscurity" that prevented any meaningful research on this subject. Hence, external communities had no way to study satellite internals and potential security issues.

In recent years, this changed as the developments in the space domain have moved towards the use of common off the shelf components, open satellite designs, and open-source libraries. These factors have been multiplied by the explosive growth in the number of satellites and the inherent increase in the size of the community. Hence, the number of people holding knowledge about satellites has been steadily increasing. Overall, we argue that a transformation is slowly happening concerning the effectiveness of security by obscurity in

space-borne assets.

As a result, we must assume that attackers have detailed knowledge of the target satellite, including detailed documentation and access to firmware images. Further, several open-source satellites already enable attackers to study satellites. We therefore assume attackers have detailed knowledge of satellites, including their firmware, except for cryptographic secrets.

So, in other words, this is a modern security model – at least from the standpoint of these researchers. The satellite industry may not have caught up yet, but the only way for researchers to test satellite security is with an honest set of assumptions of the threat. As we know, it's always necessary to assume that one's adversary knows everything about the design of their target, because too often that's true.

Another area that they needed to address they termed the "Myth of Inaccessibility." They wrote:

Until recently, it was generally assumed that satellites always communicate with prohibitively expensive GSs. As a result, only few actors could attack a satellite (similar to the assumption for mobile cell phone networks many years ago). Unfortunately, this assumption had a major impact on the adaptation of security features in satellites. However, GS prices have dropped significantly in the past few years. Today, it is possible to create a fully functional GS for less than $10k, and there are open-source communities around developing GSs. In addition, GSaaS providers such as Amazon Web Services or Microsoft Azure rent a GS to the user, or allow GS owners to monetize unused GS capacity by temporarily renting it to end users. As a result, one does not even need to own GS equipment to interact with satellites. Additionally, transceivers for specific satellite services have become so compact and cheap that they can be found in consumer electronics, such as the iPhone 14. Furthermore, there are now many LEO satellite constellations in space with satellite-to-satellite communication capability. At the same time, there is an increasing number of smaller research LEO satellites. There are already a number of satellites with significant communication capabilities in space that are even intended to be used by third parties.

Therefore, we believe that there is a paradigm shift in the assumption that satellites are inaccessible, which is particularly pronounced for LEO satellites.

The researchers examined a trio of satellites with widely varying architectures. One used an ARM Cortex-M3, another used a much more recent AVR32, and the third used a Leon3 SPARC V8 processor. But in all three cases upon reverse engineering the satellite's current firmware using IDA Pro and Ghidra (both which we've covered here in the past), in each case they uncovered multiple remotely exploitable vulnerabilities that led to remote code execution.

In return for receiving access to the firmware images, they responsibly disclosed their discoveries of a total of 13 vulnerabilities across the three satellites they examined. The good news is that skybound firmware CAN be updated. The bad news is that, for example, in the case of the ARM Cortex-M3 processor contained in a satellite launched in 2013, the firmware update process takes anywhere from several days to a week, depending on the GS and link quality. This is due to the low bandwidth of UHF/VHF components which run at 9600 baud, and shared bandwidths.

To share a sense for the sorts of things they found 13 times, they wrote:

> *Insecure-by-design TeleCommands. Even with no access protection, a satellite should be designed so that TCs do not compromise the satellite's stability without further validation. Two deliberately present TeleCommands allow arbitrary reading and writing of memory. On the technical level, the attacker controls all parameters passed to memcpy through command arguments, such that these two TeleCommands are dangerous TCs. Anyone with a custom GS could utilize them to gain remote code execution and seize control of the satellite.*
>
> *Noteworthy, the ability to execute arbitrary code, which these provide, would allow an attacker to write firmware updates to the flash memory persistently, making the takeover irreversible. Modern operating systems such as Linux or Windows, deploy defenses to prevent trivial exploitation of such vulnerabilities, but the RTOS in this ARM Cortex-M3 based satellite does not feature any such protections. In particular, neither ASLR nor stack cookies are used.*
>
> *To prove the impact of this vulnerability, we build an exploit, send our payload over the COM interface of our rebuilt satellite in the lab, and execute arbitrary code (in our case, we play sound over the connected speaker).*

So, just to be clear, this satellite they were referring to actually had deliberate commands which were received over its communications link which allowed any of the machine's memory to be read, written, or moved around. Coming from the security aware state that we've all been living in for many years now, it's difficult to appreciate what they mean when they say that the security of many of these satellites relies upon a lack of access to satellite communicating ground stations. In other words, they're not kidding at all! Some of these satellites will actually obey – by deliberate design – remote commands to read, write and move any memory in the system with no concept of protection.

Here's another example:

> *Trusted ICP Size Field. Upon receiving an ICP packet, the packet is passed through a FreeRTOS data queue to the command scheduler, which executes the associated command using the included arguments. We observed that a function parsing the command structure does not validate the "length of arguments" field against the total length of the ICP packet or payload. Thus, any external attacker can specify a malicious length field, which indicates that the arguments would be longer than they actually are. This causes a command handler function to use more bytes from the heap memory than intended, leading to a buffer overread. Hence, an attacker can include other data in the attacker-TC, which leads to a control data leak. Again, we verify that this works on the real satellite by testing it on our recreated hardware and manage to successfully exploit this vulnerability. The leak itself is reliable and is not impacted by environmental conditions, but extracting specific secrets depends on the heap layout. This vulnerability is reminiscent of the well-known OpenSSL Heartbleed vulnerability.*

Or how about this one which describes something they found in a different satellite:

> *OPS-SAT uses a flash file system to store files, including the firmware image. Existing TCs allow to create new files and write to them, providing the capability to upload a malicious firmware image onto the satellite. To change the filesystem path pointing to the current image,*

critical commands must be enabled, which is a global Boolean flag in the satellite's settings. Crucially, changing this flag can be done via a TC that does not require additional verification. Hence, external attackers can conduct arbitrary firmware updates, which allows them to seize control over the satellite. Interestingly, similar critical functionalities are hidden behind the same flag, indicating that engineers were aware of its critical importance but decided not to implement further protection.

Or how about a problem in a widely used library:

A widely used space SDK utilizes the uffs library, which implements a low-cost flash file system. The library is used on roughly 75 spacecraft. And, according to the library's author, is also used by NASA. We identified a stack-based buffer overflow vulnerability in the file renaming procedure, where the name of the new file is copied into a buffer of static size without any size check, resulting in arbitrary code execution.

We experimentally verified that this vulnerability can be exploited to gain arbitrary code execution. In OPS-SAT this function is only exposed to an inaccessible UART debug-port, posing no security threat to OPS-SAT in its current state. Still, moving files is a reasonable file system interaction to be exposed via TC to semi-privileged attackers. Hence, any of the other roughly 75 spacecrafts implemented such functionality, they are likely vulnerable.

Anyway, everyone should have the idea by now. These guys were not kidding when they characterized the satellite industry's security as lagging behind by several decades. Thanks to an attitude of "we're not the PC industry", "we're not connected to the Internet" and "you can't talk to our birds without special equipment", the security concerns that all of us on the ground have been fighting for the past several decades doesn't appear to have sunk in at all.

Sure, there are instances of mistakes that have not been caught. But the most glaring insanity are deliberately designed commands which are insanely powerful and lacking any authentication. They implicitly assume that no bad guy will ever be able to get their hands on a radio – even now that Amazon and Microsoft will happily lend you one of theirs.

I really hope that this work, and others similar to it, have or will come to the attention of all of the relevant parties. The good news is that down here on the ground, where we have the Internet which has been connecting everyone to everyone else, an insanely well-developed security awareness has grown among all of us ground dwellers. All anyone needs to do is ask.