Security Now! #1043 - 09-16-25

Memory Integrity Enforcement

This week on Security Now!

• Are Bitcoin ATMs anything more than scamming terminals? • Ransomware hits the Uvalde school district and Jaguar. • Did “Scattered LapSus Hunters” just throw in the towel? • Germany, for one, to vote "no" on Chat Control. • Russia's new MAX messenger has startup troubles. • Samsung follows Apple's WhatsApp patch chain. • Shocker! UK school hacks are mostly by students! • HackerOne was hacked. • Connected washing machines in Amsterdam hacked. • DDoS breaks another record. • Bluesky to implement conditional age verification. • Enforcement actions for Global Privacy Control. • Might Apple have finally beaten vulnerabilities?

What, exactly, is the plan here?

Security News

Athena Bitcoin ATM Sued

The District of Columbia's Office of the Attorney General has filed a well-deserved lawsuit against the largest crypto ATM operator in the US: "Athena Bitcoin". The lawsuit alleges, with ample evidence, that the company knew its Bitcoin ATMs were being used to collect funds from victims of illegal scam operations, but rather than stopping the transfers, it charged large hidden fees then refused to provide refunds to victims.

The concept of a Bitcoin ATM is very cool, serving as a real-world interface to a purely ephemeral digital currency. But we've learned that the number one enabling factor for ransomware was the emergence of cryptocurrency. One of the principal lessons to be learned from the Internet is that – sadly – anytime there's the freedom of anonymity there's its abuse. So it should come as no surprise that scammers were quick to jump onto Bitcoin ATMs as the means for suckering the uninformed into all manner of online scams. We've previously touched on the problem of ATM abuse, now thanks to this lawsuit we have a window into how bad it is:

What's somewhat surprising is that these Bitcoin ATMs see such low levels of NON-fraudulent use. Believe it or not, only 7% of Athena's Bitcoin ATM transactions were legitimate. Officials say that 93% of all deposits made across the seven Bitcoin ATMs Athena operates in DC were the result of scams. Scammers would trick victims into going to an ATM to transfer funds into their bitcoin accounts.

DC's Attorney General alleges that Athena knew that allowing users to deposit funds into accounts they don't own would be abused for scams, but did nothing to stop scams beyond displaying an (obviously) ineffective warning on the ATM screens. DC's Attorney General Brian L. Schwalb claims Athena, instead, applied large fees to every transaction. The fees were not visible to customers and reached up to 26%, almost 100 times the fees practiced by its competitors, which go from 0.24% to 3%. As a consequence, scammed individuals were victimized twice, first by the scammers and then by Athena through the undisclosed fees.

The median loss per victim was $8,000, and the victims' median age was 71, with scammers specifically going after the less technical elderly DC population. AG Schwalb brought the lawsuit as a means of forcing Athena into compliance with anti-fraud measures and to secure financial restitution for victims, as well as to pay financial penalties for the District.

Schwalb said: "Athena knows that its machines are being used primarily by scammers, yet chooses to look the other way so that it can continue to pocket sizable hidden transaction fees. Today, we're suing to get District residents their hard-earned money back and put a stop to this illegal, predatory conduct before it harms anyone else."

Speaking of ransomware...

The Uvalde school district

The Uvalde school district is shut down this week following a ransomware attack. If that name sounds familiar that's because three years ago, in 2022, an 18-year old former student fatally shot 19 students and 2 teachers, injuring 17 others. But I doubt that this ransomware attack on the district had anything to do with that. As we know, such attacks are almost always the result of targets of opportunity – Uvalde's cybersecurity was likely wanting, and was not adequately protected from someone clicking on a link they shouldn't have. The incident impacted phones, security cameras, visitor management, and thermostatic controls. Consequently, classes will be out for a week while the district gets back on its feet.

I deliberately wrote: *"Uvalde's cybersecurity was likely wanting, and was not adequately protected from someone clicking on a link they shouldn't have."* I've mentioned this thought before and it's something everyone is going to be hearing from me going forward. The evidence clearly shows and I firmly believe that the new goal for any enterprise's internal security must be to harden itself against random people inside the organization clicking on links they should not.

Today's podcast topic is about the tremendous lengths Apple has been forced to go to, to harden their system against the inevitable bugs in software. For a long time the focus was on eliminating those bugs. But we've learned that's apparently never going to happen. So Apple has committed massive resources to being able to immediately terminate any process where misbehavior is detected.

Similarly, we've talked many times about the need to train employees not to click on that link in the email that appears to be from their mother, or on that link that says they only have two days remaining before their account with their bank will be closed. That's analogous to telling every coder of every piece of software in an iPhone that they may never make another mistake. You can ask, but you're not going to get it. My point is that regardless of how much training they receive, employees ARE going to click on those malicious links. It's inevitable. So, similar to what Apple has finally been forced to do, the only sane recourse is for enterprises to get very very serious about hardening their internal security against anyone who might click on anything that they receive over the Internet – whatever it takes. If that means implementing new VLAN network segmentation and giving up the massive convenience of having everyone being able to participate as equal peers on the same network, then so be it. All of these recent massive ShinyHunter and Salesforce compromises are showing us that the calls are now coming from inside the house. The bad guys have clearly located our Achilles heel, and it is us.

My message to our listeners who are in charge of such things is that if results are what matter, rather than feel-good but ultimately failure-prone measures, it's no longer sufficient to rely upon adequate training of every single last employee, including the bosses. We've tried that. It doesn't work. The only thing that will work is arranging to make clicking on malicious links safe. That's the next frontier. We need to figure out how to do that.

An automotive supply chain in crisis

And speaking of clicking on a bad link... I wanted to touch on just one more recent ransomware attack because its consequences were somewhat unique and interesting. More than two weeks ago, Jaguar Land Rover's automotive production lines ground to a halt due to a ransomware attack. And today, all production remains halted. The company expects that at least three of its production lines may be able to resume operation later this week. But here's the interesting bit: According to the BBC, several of Jaguar's smaller suppliers are now facing bankruptcies due to the prolonged production stoppage. The loss to Jaguar themselves is estimated to be between 50 and 100 million pounds since the attack. But the ripple effects of the incident are revealing it to be perhaps one of the most significant – as in, the worst – cyberattacks in the country's history. It's even expected to affect Britain's national economic growth stats!

The end of "ShinyHunters/ScatteredSpider/LAPSUS$" ??

It's impossible for us to know what's actually going on. But that hybrid group that was calling itself "Scattered LapSus Hunters" composed of individuals from ShinyHunters, ScatteredSpider and LAPSUS$ posted a rambling goodbye note referring to their attack on Jaguar and four moderate intrusions into Google. They're the one who we noted last week were demanding the firing of two of Google's Thread Intelligence Group "or else."

As is the case with so many of these sorts of things, we'll almost certainly never know what really happened here. But we've been covering the consequences of their actions which, while not qualifying as a reign of terror, did at least certainly put this group squarely on the map. It might just be that they ran dry of targets of opportunity which they had previously acquired. Or perhaps some counter cyber-intelligence managed to penetrate their ranks to convince them to stand down. Whatever the case, I wanted to keep our listeners current with the news that they have formally said goodbye. We'll see what happens next.

Germany opposed breaking encryption for the EU's "Chat Control"

Many of the governments within the European Union have by no means given up on legislation to obtain some sort of access or control of privately encrypted interpersonal messaging among its member citizens. But there is some disunion, evidenced in news from last Wednesday, posted by the German government which indicated that they, Germany, will have none of that. Period. They wrote:

> *September 10th, 2025 — Berlin. From the Digital Affairs & State Modernization Committee:*
>
> *The Digital Affairs Committee met Wednesday afternoon to discuss the status of the CSAM regulation, known publicly under the term "chat control." Its purpose is to combat sexual violence against children and adolescents online. For over three years, various proposals have been under discussion at the EU level to require providers of messaging and hosting services to detect material related to online sexual child abuse. An agreement has not yet been reached.*
>
> *As a representative of the Federal Interior Ministry reported to the Members of Parliament, the Danish Presidency of the Council, in office since early July, is treating the matter as a high priority. A unified legal basis across the EU is urgently needed, given that the current situation is worrying. It is clear that private, confidential communication must remain private. At the same time, there is an obligation to take action against child abuse online.*
>
> *A representative from the Federal Ministry of Justice pointed out that the matter involves very severe intrusions into privacy, leaving open the question of how deep those intrusions are. He also pointed to the strict limits that have already been made clear in EU Court of Justice case law on data retention, and emphasized that a regulation is needed which will stand legal scrutiny.*

Oops. In other words, the EU already has strong existing laws that would make what "Chat Control" wants to accomplish illegal under their own law. The article finished, writing:

> *In their questions, MPs asked about the joint position of the federal government, the criticism from civil society about the regulation, and the further process in the negotiations. The representative from the Interior Ministry explained that the Danish position could not be supported 100%. For example, Germany is opposed to breaking encryption. The goal is to produce a unified compromise proposal — also to prevent an interim regulation from lapsing.*

This has all the earmarks of being a very heavy lift. This Chat Control dream faces very stiff headwinds. I don't know what it means for Germany to declare that it's a firm "no" vote, but the EU's existing personal privacy laws will need to be changed for Chat Control to be legal.

Russia’s new MAX messenger not launching so well.
It turns out that even when there are many Western models to follow, launching a new secure messaging service from scratch is not a slam dunk. The news out of Russia is that hackers immediately began selling hacked accounts for Russia’s MAX messenger for prices up to $250, or access to accounts can also be rented by the hour. Working to combat this abuse, Russian officials say they've already blocked more than 67,000 accounts for suspicious activity, such as spam and sharing malicious files. Looks like the Kremlin and Roskomnadzor are going to have their hands full dealing with the consequences of their own messaging service. Couldn’t happen to a nicer bunch.

Samsung echoes Apple with a patch
Samsung recently patched a zero-day (CVE-2025-21043) which they rated as CRITICAL in the Android OS version that ships with its devices. The vulnerability was discovered in Android’s libimagecodec.quram.so file. I didn’t dig in to see whether it may have been similar to that Apple also recently patched. But, like the recently patched Apple vulnerability, this one also formed part of an exploit chain that targeted WhatsApp users. In both the Android and Apple cases, the exploit chain was used to deploy spyware on targeted devices.

Most UK school hacks caused by students
While I was assembling today’s show notes I was reminded that there’s all the difference in the world between a casual mistake made by an employee who clicks on a malicious link they receive and an employee on the inside who wishes to maliciously attack their own employer.

An article from the UK’s privacy watchdog is what reminded me of the difference. They found and reported that UK students are increasingly behind the hacks of their own schools. The UK Information Commissioner's Office says it studied 215 insider-caused breaches within the UK educational sector between 2022 and mid-2024 and found that students were behind 57% of all the intrusions. And where a stolen password was used to breach a school system, students were involved in almost all cases (97%). The underlying motives were cited as being dares, notoriety, financial gain, revenge, and rivalries. In other words, “because it’s possible to do it” sorts of hijinks.

Breaches were blamed on staff leaving devices unattended, students being allowed to use staff devices, incorrect permissions on school resources, and, in some, though rare 5% of cases, on students using sophisticated techniques to bypass security and network controls.

After researching those 215 insider student-caused breaches, the Information Commissioner's Office reached a pair of conclusions. The first one was that an early familiarization with hacking might lead kids down the wrong path and serve as a gateway to a life of cybercrime. I remember being that age and I was notorious for all manner of hijinks – the adventure of the portable dog killer to name one. But I think it would be a stretch to imagine that some high schooler’s success with guessing a teacher’s password (or looking underneath the keyboard for it) would lead to a life of cybercrime. After all, everyone is an insider within their own family’s home where plenty of tantalizing hacking opportunities may exist. So one’s school is just another of many. The second conclusion the ICO reached was that the responsibility for much of their students’ hacking successes lay at the feet of the school’s administrators who repeatedly failed to properly and adequately secure their networks. (And writing one’s password on a Post-It note under the keyboard is never a good idea!) In conclusion the ICO urged schools to “remove the temptation from their students” by taking steps to improve their cybersecurity and data protection practices.

HackerOne Hacked

It’s never a good sign when a security-aware Bug Bounty company such as HackerOne themselves gets hacked. But this wasn’t on them. The blast radius of the recent Salesloft Drift supply chain attack has been wide and deep and HackerOne caught up in it. They first posted this back on August 28th:

> *Recently, hundreds of companies have been responding to an attack that resulted in unauthorized access to Salesforce records connected to the Drift (from Salesloft) application, a situation detailed in reports from Mandiant and others.*
>
> *As part of our commitment to transparency, trust, and our company’s value of “Default to Disclosure”, we are writing to confirm that HackerOne is among the companies impacted by this incident. Our security team received notice of the potential compromise from Salesforce on Friday, August 22, and this was confirmed by Salesloft on August 23. HackerOne’s security team immediately initiated incident response procedures, working in partnership with Salesforce and Salesloft, to assess the scope and impact of this incident.*
>
> *HackerOne’s investigation is ongoing, but we can confirm that a subset of records in our Salesforce instance was accessed via a compromise of the Drift application. Due to HackerOne’s strict policies and controls governing data segmentation, we have no reason to suspect that the incident impacted or exposed any customer vulnerability data.*
>
> *We are continuing to conduct forensics on the records that were accessed and will communicate directly with any impacted customers, as appropriate.*

So, everything that we’d want to see. A straightforward reporting of the event with a promise to follow-up when anything more is learned. And that follow-up was posted last Thursday. They wrote:

> *HackerOne continues to investigate the recent Salesloft Drift incident, and we are posting here to update you on the status of our investigation as well as provide additional information we are able to share at this time.*
>
> *Based on the information we have to date, a subset of HackerOne's Salesforce data was accessed via the Drift application on August 13th and August 18th. Both the dates and the indicators of compromise are consistent with what Salesloft has reported, which can be found at trust.salesloft.com.*
>
> *We can confirm that all Salesforce Drift connectors are currently offline, and, as a precaution, we have rotated all relevant API and service credentials. Due to HackerOne's strict policies and controls governing data segmentation, we have no reason to suspect that the incident impacted or exposed any customer vulnerability data. Nor have we found any indication of lateral movement.*
>
> *We understand that you may still have questions about this incident, and we appreciate your patience as we continue our investigation. HackerOne has engaged a third-party forensics firm to ascertain what records were accessed, and we will communicate directly with impacted customers, as appropriate.*

I usually try to find some lesson for us to take away from the incidents we cover. But I don’t know of one here. Today’s modern model of outsourcing services and interconnecting separate enterprise's automated systems with persistent authentication bring an inherently risk which we are currently seeing play out. One of the recent trends I'm sure everyone listening has encountered is the increasing (and at least for me) annoying use of automated conversational AI chat windows that increasingly appear, typically in the lower right corner of a website. I have yet to find engaging with one of those annoyances to be fruitful. If you've encountered one of those it may have been courtesy of Salesloft Drift since that's what their technology does that's been the root cause of all of this pain.

Salesloft Drift describes themselves as: <quote> "A conversational-AI / chat / lead qualification component of the Salesloft platform. It's built on or integrates the Drift chat/AI agent that engages website visitors in real time, qualifies leads, routes them to the sales team (via workflows like Rhythm), and helps convert them into pipeline."

I don't want to be converted "into pipeline" (whatever the heck that is) all I want to know is whatever happened to the end table we ordered? But that information is not available through the chatty chat bot.

In order to integrate with its client enterprise customers, this Salesloft Drift AI system needs to have access into its customer's networks. Consequently, when Salesloft Drift is hacked, all of its many customers' networks then suffer their own respective breaches as the hackers of the company to which they have outsourced this service obtain the credentials that allow access into every one of those enterprises' internal networks.

It's an inherently unstable solution with an astonishing blast radius. But they, you get to annoy every one of your visitors by asking them, unprompted, what they need and whether there's anything that they want to ask – while not having any answers. This is what we call progress.

Washing Machines Hacked

It was a little over a year ago in episode #975, May of 2024 that we last talked about students hacking their university-provided washing machines to obtain free laundry services. Now, a university campus in Amsterdam has shut down its laundry room after its five smart washing machines were hacked in July. Students were able to wash their clothes for free for months but that will be ending shortly. Those five Internet-connected smart machines are being replaced with "dumb" washing machines that accept old fashioned coins. Who even has coins anymore? Seems like the students are going to get what they deserve, needing to somehow now find coins.

As I mentioned when we talked about this before, UC Berkeley also provided coin-op washing machines in pre-Internet 1973, when I happened to be there. And, really, what did they expect? The machines had been placed in Ehrman Hall, which was the engineering dorm, where I was. It turned out that the coin-op box had been added as an after-thought and the removal of a sheet metal screw from the back of the box created a hole through which a properly shaped length of coat-hanger wire could be threaded. And with a bit of fishing around inside the box, the lever that was normally actuated by the insertion of a quarter into the front could be tricked into believing that had just happened. So let's just say that I never needed to bring laundry home on the weekends for my mom to wash.

Another record breaking DDoS attack

I'm just going to start this next piece by reading what was posted, then I'll share my sadness:

> *London, UK – Tuesday, September 9th 2025 – FastNetMon today announced that it detected a record-scale distributed denial-of-service (DDoS) attack targeting the website of a leading DDoS scrubbing vendor in Western Europe.*
>
> *The attack reached 1.5 billion packets per second (1.5 Gpps) — one of the largest packet-rate floods publicly disclosed. The malicious traffic was primarily a UDP flood launched from compromised customer-premises equipment (CPE), including IoT devices and routers, across more than 11,000 unique networks worldwide.*
>
> *The disclosure comes only days after Cloudflare reported mitigating an 11.5 Tbps DDoS attack, showing how attackers are pushing both packet and bandwidth volumes to unprecedented levels.*
>
> *Pavel Odintsov, Founder of FastNetMon said: "This event is part of a dangerous trend. When tens of thousands of customer-premises equipment devices can be hijacked and used in coordinated packet floods of this magnitude, the risks for network operators grows exponentially. The industry* ***must*** *act to implement detection logic at the ISP level to stop outgoing attacks before they scale."*
>
> *FastNetMon Advanced platform is designed to handle attacks of this size. Using highly optimised C++ algorithms for real-time network visibility, FastNetMon enabled its customer to automatically detect the flood within seconds — preventing disruption to the target service.*

Okay, I'm not sure what "highly optimized C++ algorithms" have to do with anything, and, unfortunately, this Pavel guy is dreaming. We've been talking about the problem of DDoS flooding throughout the entire 20 years of this podcast and during that time, while attacks have grown astronomically in scale they have also become less possible to prevent.

Back in the early days, spoofing source IP addresses was the order of the day. We argued at the time, correctly, that no ISP should emit any packets from their networks that contained a fraudulent source IP. "Egress filtering" could be employed to nip those attacks in the bud before the traffic was given the chance to aggregate into an overwhelming flood. And that was true.

But the only reason devices back then were spoofing their source IP addresses was to hide their true IP from their victims. Once you have tens of thousands of individually compromised home routers and IoT devices, hiding is no longer necessary. Who cares if the identity of these devices is known? They're scattered across the globe in faraway countries behind ISPs that will never answer the phone. As a consequence, source IP spoofing as a requirement for packet and bandwidth flooding is far less important today than it once was.

The other factor is that it's trivial for a CDN like Cloudflare to drop all incoming readily spoofable UDP traffic. It's a web hosting provider so it used to need TCP over port 80 and 443. And as we noted recently, even port 80 is falling by the wayside. So now the name of the game is connection flooding, and connection flooding needs TCP with round-trip packets. This prohibits the use of any spoofing. But again, who cares when today's massive bot networks have tens of thousands of individually throwaway agents?

One of the earliest things we talked about on this podcast during our "How the Internet Works" series was the brilliant genius invention of the idea of opportunistic packet routing. By completely dropping the idea that every communication packet needed to get through the network with 100% reliability, the designers of the Internet invented an incredibly elegant solution for the ages.

There's just one problem: To this day – and forevermore – that incredibly elegant system is utterly and completely vulnerable to packet generation abuse and there is no way to fix that. None. This astonishing global network is there so that anyone anywhere can send a packet to anyone else anywhere else. Unfortunately, there's nothing to prevent bad guys with thousands of remotely scattered devices under their control all sending as much packet traffic as they can to anyone they choose.

The result of this is that frequently-targeted companies are choosing to hide behind the growing number of companies who are able to provide comprehensive DDoS protection thanks to having many points of Internet presence, their own massive network bandwidth, and automation to block incoming attack traffic. It's not an ideal solution but I suppose it's the price we pay for a system that otherwise works so incredibly well.

Bluesky to implement "conditional" age verification for South Dakota and Wyoming

As age verification requirements continue to evolve, we have an update last Wednesday from Bluesky. Recall that last time we talked about Bluesky they were going completely dark in Mississippi due to Mississippi's "all or nothing" age verification law. After the first two paragraphs that don't say anything, Bluesky wrote:

> *In the UK, we complied with a new law that requires platforms to restrict children from accessing adult content. In Mississippi, the law requires us to restrict access to the site for every unverified user. To implement this change, we would have had to invest substantial resources in a solution that we believe limits free speech and disproportionately harms smaller platforms. We chose not to offer our service there at this time while legal challenges continue.*
>
> *South Dakota and Wyoming have also passed online safety laws that impose requirements on services like ours. These are very similar to the requirements of the UK Online Safety Act. So, as we did in the UK, we'll enable Kids Web Services' (KWS) age verification solution for users in these states. Through KWS, Bluesky users in South Dakota and Wyoming can choose from multiple methods to verify their age. We believe this approach currently strikes the right balance. Bluesky will remain available to users in these states, and we will not need to restrict the app for everyone.*
>
> *We're committed to keeping our community informed as we navigate these new regulations. As more states and countries adopt similar requirements, we will update this blog post accordingly.*

So just to be clear, the difference between Mississippi, South Dakota and Wyoming is that the more sane laws passed in South Dakota and Wyoming only require age verification before their citizens are allowed to access adult content as opposed to all social media content. That's what's similar to what was done in the UK.

But following the tragic Mississippi suicide of the young man who was catfished on Instagram, the state of Mississippi has effectively declared war on all social media regardless of its content. While 1st amendment lawsuits are flying, Bluesky decided to just back out of Mississippi until the dust settles.

And speaking of the UK

And speaking of the UK, the UK really appears to be on the war path following their July 25th passage of the new age-check requirements under their Online Safety Act. Only a week after its passage they announced that they had launched investigations into the compliance of four companies – which collectively run 34 pornography sites – to verify that they were now using "highly effective age assurance" to prevent children from accessing that content. At the time they said that these 34 new cases added to Ofcom's 11 investigations already in progress into 4chan, an online suicide forum, seven file-sharing services, and a pair of other porn publishers.

They concluded by saying that they expected to be making further enforcement announcements in the coming weeks and months... which just happened last Thursday with their apparently proud announcement that another 22 porn sites were now being investigated to verify the effectiveness of their age verification measures.

It's one thing to need to show your ID in order to pick up a medication prescription, or before purchasing alcohol. But it's obviously a far more sensitive matter to need to produce an ID in order to obtain access to online content that is, to say the least, controversial and probably extremely embarrassing. So it's hardly any surprise to learn that the traffic of the websites that are requiring such proof of age has dropped precipitously and significantly.

Global Privacy Control

We talked about GPC – the Global Privacy Control – which is a signal, reminiscent of its predecessor DNT – Do Not Track – which never got off the ground since without enforcement it means nothing. It's simply ignored. But on the enforcement front, GPC may have a brighter future. The news is that state attorneys general from California, Colorado, and Connecticut have announced a joint investigation into companies refusing to comply with Global Privacy Control. Data trackers that refuse to honor the GPC are in violation of recently passed state privacy laws. Seven other US states also require companies to honor GPC, but have not joined the enforcement action. This is great news since, as I noted, without true enforcement the law means nothing and will likely suffer the same fate as befell DNT.

Listener Feedback

Micheal Buck

> *Hi Steve, In Episode #1040 you talked about your disappointment with what you called "Synology's built-in NAS synchronizer." I'm not sure you gave your listeners a fair review of Synology's solutions. I am a Synology user and have used Synology Drive, which works like SyncThing, DropBox, and other synchronizing tools. Like you, I have several machines that I use and like to keep files synchronized between these machines. Synology Drive was easy to set up and I have been using it for years without any problems. It keeps my files synchronized between multiple Mac and Linux machines. I also use the tool that Leo mentioned, HyperBackup. Most Synology NAS machines have an external USB port. My son also has a Synology and we each purchased a large USB drive and plugged them into each others' NAS's USB port. Then we each use HyperBackup to back up our NAS machines to our own USB drives at each other's location. The data is encrypted, and we don't eat up the disk space on each other's NAS. Thanks for all you and Leo do to provide a great podcast.*
>
> *Cheers. Mike / SpinRite owner and podcast listener since episode 1 / Payson, Utah*

In case anyone else may have been confused by my disappointment with Synology's built-in inter-NAS synchronization, I wanted to take another moment to clarify. There was nothing whatsoever wrong with it. I agree with Mike that it was quick and easy to set up. And I have a strong bias toward what we would refer to a "living off the land" solutions – meaning that if Synology provides a means of keeping two of their NASes synchronized I would be strongly inclined to assume that they know best how to do it.

And, again, it worked. I would have never been unhappy with it or aware that the system, at least for me, was operating in a far from optimal way unless I had been watching the Synology drive's massive apparent full resynchronization using SoftPerfect's wonderful free NetWorx utility I've spoken of before. I have it configured to continually display the SNMP counters on my router's interface, so it's showing me not my own machine's bandwidth, but the instantaneous bandwidth usage of my entire LAN.

What I witnessed to my extreme chagrin on many occasions was my network's bandwidth being pinned for a very long period of time after only updating a few files on my NAS. And when I checked the NAS's drive lights they were all flashing away like mad.

Again, everything worked, but it was certainly not a situation that I wanted to live with. The only change I made was to shutdown Synology's native synchronizer and run SyncThing natively on both NASes with them synchronizing everything. Now, using SynchThing, when I update a few files on my local NAS after building a new instance of the DNS Benchmark, after a short delay I'll see a brief, few seconds long "blip" of outgoing bandwidth as my local SyncThing instance sends those – and only those – updated files over to the other NAS.

So, yes, SyncThing's native synchronization works. No question about it. And it's easy to set up and configure. But it might be worth monitoring its bandwidth usage, or even just watch its drive activity lights after you've updated a bunch of files all at once.

Greg Williams

> *Hi Steve, Just a few notes. Cloudflare already has Certificate Transparency Monitoring (although it's in preview): https://developers.cloudflare.com/ssl/edge-certificates/additional-options/certificate-transparency-monitoring/. No idea why they didn't use it themselves.*
>
> *You also mentioned the 1.1.1.1 domain. That's not a domain, it's an IP address that's not owned directly by Cloudflare, but APNIC (more info https://blog.cloudflare.com/announcing-1111/). See the Wikipedia article on https://en.wikipedia.org/wiki/1.1.1.1 #Prior_usage_of_the_IP_address for other references to the default use of 1.1.1.1 as laziness by other vendors, including Cisco. Cheers, Greg Williams, Brisbane, Australia*

Of course, Greg is 100% correct about 1.1.1.1 not being a domain. I know better. The numeral "1" is not a TLD and numeral "1" could never **be** a TLD since the RFC-specified minimum length of any TLD is two characters. So thanks for the correction, Greg. I got a kick out of Greg's reference to a Wikipedia page section titled *#Prior_usage_of_the_IP_address* which Greg said contained other references to the use of 1.1.1.1 out of laziness by other vendors, including Cisco. So my guess that there was almost certainly never any malicious intent behind this was probably correct. And these clowns were not the first to use 1.1.1.1 out of ease and laziness.

Buzz

> *I've listened to the last show, and as a UK Citizen I can confirm that Apple's ADP is still active for those users who opted in at the start. It is unavailable to any New Users. Best Regards, Buzz*

Dan Bright

> *Hi Steve, Regarding last week's talk about the availability of Apple's ADP in the UK, I have it turned on myself, and can confirm that Apple has not yet removed it from my account.*
> *Kind regards, Dan (Scotland).*

Buzz' and Dan's notes were echoed by other listeners who all confirmed that while it was still no longer possible to enable "fresh" ADP, it had never yet been forcibly removed from any UK-based Apple user who has reported in. If the effect of the still-inferred and presumed UK "notice" stands then the presumption is that Apple will eventually be required to ask all UK users to please flip the switch off... or perhaps Apple will themselves preemptively disable the feature and inform their users that "the devil made them do it."

John David Hickin

> *I'm following the proposals to solve the problem of asserting that age of X is >= Y. Zero knowledge proofs may come in handy here, but it seems to me that there is a potential use case that deserves thinking about: If different sites start to impose differing age requirements (while attracting the same visitors) then web tracking across those sites may be able to refine (upwards) the lower limit on any person's 'guessed' age. I'm not sure if it's a real issue but somebody will surely try to monetize it.*

John’s thinking is correct and clever and it reminds us that even just the assertion that we’re of at least some given age represents an explicit disclosure of information. The handwriting is certainly on the wall that the previous era of full and free unfettered access to the Internet’s content is rapidly drawing to a close thanks to recent legislation in the UK, soon to come to the EU and already within many State jurisdictions of the United States of America. Internet websites, which inherently have global reach, are being required to comply with the laws which govern their visitors which often requires that those visitors sacrifice the fully anonymous access we have enjoyed up to this point to the requirement of an acceptable form of age verification.

I haven’t noted this before, but we may see safe havens for anonymous Internet access spring up in the wake of these new legal restrictions. Websites that are compelled to obey the law might geolocate their visitors and limit their age restriction enforcement to only those countries that impose these requirements. Given that doing so is entirely possible, it would seem to follow logically from country-specific legal requirements. So, for example, anyone coming from the UK, the EU or the US would be required to provide proof of their age. But, for example, Icelandic visitors, who are outside the EU and live within a society with very liberal Internet regulations might not be required to give up any identifying information.

If that were the case, it would not be a stretch to imagine commercial VPN providers deliberately establishing points of presence in Iceland and offering customers anywhere, including the UK, EU and US, the option of having their VPN traffic routed out through Icelandic locations.

Memory Integrity Enforcement

Apple’s big September 2025 product update announcement included technical capability advance which, while garnering much less attention, was nevertheless perhaps somewhat more important in the long run for Apple’s users than their decision to create a new Cosmic Orange color for their latest iPhone. Under the covers of any iPhone 17 and its A19 chips lies an advance in hardware technology that goes further than anything Apple has previously implemented to prevent coding mistakes from being leveraged into exploitable vulnerabilities that can be used against iPhone users.

It’s worth remembering that if today’s incredibly complex code did not contain subtle mistakes, none of these extra-fancy prophylactic measures would be required for security. Two weeks ago, everyone needed to update and reboot their iOS and iPadOS devices after Apple discovered that a subtle flaw in the decompression code for Adobe’s DNG lossless image compression format, coupled with a registration bypass flaw in WhatsApp, was being leveraged in the wild, almost certainly by the customers of commercial spyware vendors, to install spyware into the iDevices of highly-targeted Apple users.

Were it not for the apparent impossibility of catching all mistakes before they ship, there would be no need to go to these seemingly endless lengths to protect the users of these devices from their abuse. But one of the painful lessons the industry has reluctantly acknowledged as our understanding of the nature of security has matured is that mistakes are not disappearing, and they may never, because we’re always pushing the boundaries of what’s possible for us to build. This created the concept of “Layered Security” described as “Defence in Depth”. The idea is to, wherever possible, establish multiple, often redundant, layers of protection so that the failure of any one or more layers would still leave a system’s effective security intact.

Furthering this apparently endless effort, last Tuesday, Apple’s *“SEAR”* group, where SEAR stands for *“Security Engineering and Architecture Security Research”* informed the world of their latest and greatest hardware-assisted technology that has been incorporated into the A19 processor chips being used by their iPhone 17 and other just-announced devices. Their blog posting was titled: *“Memory Integrity Enforcement: A complete vision for memory safety in Apple devices”*.

I’ll start by sharing just the first two sentences of their posting, after which we’ll need to pause to catch our breath. Apple’s team wrote:

> *Memory Integrity Enforcement (MIE) is the culmination of an unprecedented design and engineering effort, spanning half a decade* [also commonly known as 5 years]*, that combines the unique strengths of Apple silicon hardware with our advanced operating system security to provide industry-first, always-on memory safety protection across our devices — without compromising our best-in-class device performance. We believe Memory Integrity Enforcement represents the most significant upgrade to memory safety in the history of consumer operating systems.*

Okay. That certainly sets the bar high. The reason we’re here today is to gain an understanding of what Apple has done to justify this claim. Their posting then continues to remind us of the nature of the threats they face and some details of their journey up to this point. I'm going to share that, interrupting to comment or elaborate as needed. They write:

> *There has never been a successful, widespread malware attack against iPhone.*

That's true and it's worth remembering. Microsoft might argue that Windows, being a far more open platform compared to Apple's much more controlled environment, faces a much more daunting security challenge, but all of Microsoft's biggest problems were of their own making with their own code. All of those early Internet worms leveraged fundamental flaws in Microsoft's IIS web server and the many continuing problems with Microsoft's NT LAN Manager and their Remote Desktop protocol were, in every case, enabled by Microsoft's poor coding and insecure protocol designs. Apple has objectively done a far better job, and their devices are every bit as well-connected as Microsoft's. Apple continues:

> *The only system-level iOS attacks we observe in the wild come from mercenary spyware, which is vastly more complex than regular cybercriminal activity and consumer malware. Mercenary spyware is historically associated with state actors and uses exploit chains that cost millions of dollars to target a very small number of specific individuals and their devices. Although the vast majority of users will never be targeted in this way, these exploit chains demonstrate some of the most expensive, complex, and advanced attacker capabilities at any given time and are uniquely deserving of study as we work to protect iPhone users against even the most sophisticated threats. Known mercenary spyware chains used against iOS share a common denominator with those targeting Windows and Android: they exploit memory safety vulnerabilities, which are interchangeable, powerful, and exist throughout the industry.*

Right. As I noted earlier, despite all the lessons we've learned, even recently authored code, such as that Adobe DNG file decompressor, continues to exhibit exploitable vulnerabilities. Apple writes:

> *For Apple, improving memory safety is a broad effort that includes developing with safe languages and deploying mitigations at scale. We created Swift, an easy-to-use, memory-safe language, which we employ for new code and targeted component rewrites. In iOS 15, we introduced kalloc_type, a secure memory allocator for the kernel, followed in iOS 17 by its user-level counterpart, xzone malloc. These secure allocators take advantage of knowing the type — or purpose — of allocations so that memory can be organized in a way that makes exploiting most memory corruption vulnerabilities inherently difficult.*
>
> *In 2018, we were the first in the industry to deploy Pointer Authentication Codes (PAC) in the A12 Bionic chip, to protect code flow integrity in the presence of memory corruption. The strong success of this defensive mechanism in increasing exploitation complexity left no doubt that the deep integration of software and hardware security would be key to addressing some of our greatest security challenges. With PAC behind us, we immediately began design and evaluation work to find the most effective way to build sophisticated memory safety capabilities right into Apple silicon.*

To put this into perspective, the earliest efforts at building barriers around memory to protect its misuse were implemented in software. They were useful and effective, but they turned out to fall short of being “absolute”. As a consequence, while the bar was meaningfully raised, this just meant that the bad guys needed to work a lot harder and that, in turn, the governments needed to pay more as exploits became significantly more rarified. Unfortunately for journalists, political activists and other targeted individuals, governments have no shortage of funds, nor willingness to pay a competitive price.

After adding address space layout randomization, kernel address space layout randomization, stack cookies, reference counting and other software-based mitigations, all of which were eventually worked around by highly motivated attackers, the ante had been upped and it was time to start adding explicit anti-exploitation features to the underlying hardware. Apple wrote:

> *Arm published the Memory Tagging Extension (MTE) specification in 2019 as a tool for hardware to help find memory corruption bugs. MTE is, at its core, a memory tagging and tag-checking system, where every memory allocation is tagged with a secret; the hardware guarantees that later requests to access memory are granted only if the request contains the correct secret. If the secrets don’t match, the app crashes, and the event is logged. This allows developers to identify memory corruption bugs immediately as they occur.*

Let me pause here to highlight this distinction, because it becomes important. Arm’s MTE was introduced six years ago in 2019 with the ARM v8.5-A architecture. Its intention, design and focus was to assist debuggers – both software and people – during code development time debugging. Running code under a debugger that would attempt to verify and validate every memory access would introduce prohibitive overhead. We’ll be talking a lot about overhead in a bit – everything is about overhead. So Arm’s MTE was added to the ARM architecture to allow the hardware, while running at full speed, to detect instances of “use after free” and “out of bounds” accesses. It’s not possible to do this without hardware assistance.

By tagging memory allocations with what were known as “colors” consisting of 4-bit tags, and then checking those against pointer tags at runtime, MTE was able to provide a low-overhead, always-available bug-trapping mechanism in hardware.

Since we’re going to be talking about “tagging” a lot, let me clarify what’s going on here. When an application running on behalf of its user or some process in the kernel needs the use of a block of memory, for example, a buffer to store some incoming communications data, the app or kernel process makes a request of the operating system’s memory management system. For decades, a memory manager will locate some free memory, increment its usage count to show it as being in use, and return a pointer to the requested memory to its requestor. From that point on, that memory would be considered to be “owned” by the requesting application and it would be free to do anything with it that it wished. Unfortunately, the required flexibility of access required that the memory’s ownership not be enforced – any other process that knew where the memory was located could also access it. This is what the introduction of MTE changed. Under Arm’s Memory Tagging Extension, the requestor would receive not only a pointer to a block of memory that satisfied its request, but also a short “tag” – secret key that would need to be present any time that memory was accessed. The theory was that while bad guys might be able to arrange to determine where some memory was that had recently been freed or might still be in use, requiring that they would need to determine that memory’s access tag significantly raised the bar for memory access abuse.

But, MTE proved to be insufficient for Apple’s needs. They wrote:

> *We conducted a deep evaluation and research process to determine whether MTE, as designed, would meet our goals for hardware-assisted memory safety. Our analysis found that, when employed as a real-time defensive measure, the original Arm MTE release exhibited weaknesses that were unacceptable to us, and we worked with Arm to address these shortcomings in the new Enhanced Memory Tagging Extension (EMTE) specification, released in 2022. More importantly, our analysis showed that while EMTE had great potential as specified, a rigorous implementation with deep hardware and operating system support could be a breakthrough that produces an extraordinary new security mechanism.*
>
> *Consider that MTE can be configured to report memory corruption either synchronously or asynchronously. In the latter mode, memory corruption doesn’t immediately raise an exception, leaving a race window open for attackers. We would not implement such a mechanism. We believe memory safety protections need to be strictly synchronous,* ***on*** *by default, and working continuously. But supporting always-on, synchronous MTE across key attack surfaces while preserving a great, high-performance user experience is extremely demanding for hardware to support.*
> *In addition, for MTE to provide memory safety in an adversarial context, we would need to finely tune the operating system to defend the new semantics and the confidentiality of memory tags on which MTE relies. Ultimately, we determined that to deliver truly best-in-class memory safety, we would carry out a massive engineering effort spanning all of Apple — including updates to Apple silicon, our operating systems, and our software frameworks. This effort, together with our highly successful secure memory allocator work, would transform MTE from a helpful debugging tool into a groundbreaking new security feature.*
>
> *Today we’re introducing the culmination of this effort: Memory Integrity Enforcement (MIE), our comprehensive memory safety defense for Apple platforms. Memory Integrity Enforcement is built on the robust foundation provided by our secure memory allocators, coupled with Enhanced Memory Tagging Extension (EMTE) in synchronous mode, and supported by extensive Tag Confidentiality Enforcement policies.* ***MIE*** *is built right into Apple hardware and software in all models of iPhone 17 and iPhone Air and offers unparalleled, always-on memory safety protection for our key attack surfaces including the kernel, while maintaining the power and performance that users expect. In addition, we’re making EMTE available to all Apple developers in Xcode as part of the new Enhanced Security feature that we released earlier this year during WWDC.*
>
> *The rest of this post dives into the intensive engineering effort required to design and validate Memory Integrity Enforcement.*

Okay. So let’s get all these abbreviations straight. Originally, to aid in debugging, ARM-designed and introduced MTE – Memory Tagging Extension – back in 2019. But MTE was never designed to be used in an adversarial environment. It was designed to be a debugging aid. So, for example, it was acceptable if it operated asynchronously from the code, notifying of a violation sometime after the fact. That would be acceptable for a debugger. But in an adversarial setting the damage might have already been done by the time an exception was raised.

After experiencing for themselves MTE’s limitations, three years later in 2022, Apple worked closely with Arm on the development and implementation of EMTE, their “Enhanced Memory Tagging Extension”.

Original MTE allowed non-tagged memory regions, for example global or static allocations or untagged regions, to be accessed without any tag checks, meaning that attackers could exploit out-of-bounds writes into such regions. EMTE addressed this by requiring access from a tagged memory region into non-tagged memory to respect tag knowledge. This prevented the use of untagged memory from being used as a tag bypass. EMTE also brings more comprehensive enforcement of tag mismatches, especially in synchronous mode, so that buffer overflows and use-after-free bugs are blocked immediately, not just signalled later or more coarsely. There's a lot more to the improvements that EMTE brought over its predecessor MTE but with their A19 ARM chips Apple has already moved on to their next generation of even more rigorous protections.

Apple's MTE can best be seen as an evolution of EMTE where it adds various final touches to EMTE's already very useful protections. At first glance these 4-bit tags might not appear to be very useful because 4 bits, having just 16 possible states, cannot contain much security entropy. But the way they're employed is very clever. Allocations are made with the same granularity as memory pages, which are 16 Kbytes each. One of the guarantees made by the system's memory allocator is that adjacent allocations of memory will always have differing tags. This cleverly nips buffer overflows in the bud. If some adversary were able to arrange to compromise an application to obtain access to both its memory and its associated memory access tag, it would be unable to read or write outside of the application's allocated memory region because those adjacent "buffer overflow" regions would be guaranteed to be using differing tags, with neither the benign application nor its malicious compromiser having any way of knowing or predicting any adjoining allocation tag's differing 4-bit value. Thus, the infamous buffer overwrites are stopped cold.

The equally pernicious and ubiquitous use-after-free vulnerabilities are similarly prevented by having the updated EMTE memory allocator change the access tags of all freed memory as it is being freed. Thus, in the same way, if an application has been compromised so that malware obtains access to the memory pointer and tag of memory it has just released back to the system, any subsequent attempt by the malware to use that memory after it has been freed will be trapped and blocked immediately. No more using memory after it's been freed!

If you'll pardon the pun, "ARMed" with this background, Apple's further explanations will make more sense. They wrote:

> *A key weakness of the original MTE specification is that access to non-tagged memory, such as global variables, is not checked by the hardware. This means attackers don't have to face as many defensive constraints when attempting to control core application configuration and state. With Enhanced MTE, we instead specify that accessing non-tagged memory from a tagged memory region requires knowing that region's tag, making it significantly harder for attackers to turn out-of-bounds bugs in dynamic tagged memory into a way to sidestep EMTE by directly modifying non-tagged allocations.*
>
> *Finally, we developed Tag Confidentiality Enforcement to protect the implementation of our secure allocators from technical threats and to guard the confidentiality of EMTE tags — including against side-channel and speculative-execution attacks.*

> *Our typed allocators and EMTE both rely on confidentiality of kernel data structures from user applications, and of the tags chosen by the allocator. Attackers might attempt to defeat EMTE, and in turn Memory Integrity Enforcement, by revealing these secrets. To protect the kernel allocator backing store and tag storage, we use the Secure Page Table Monitor, which provides strong guarantees even in the presence of a kernel compromise. We also ensure that when the kernel accesses memory on behalf of an application, it's subject to the same tag-checking rules as userspace.*
>
> *Attacks based on speculative execution can also be used to expose secrets. To improve performance, modern CPUs predict the execution of instructions that follow prior, potentially longer latency instructions. If the prediction is correct, computation is very fast. If it's wrong, the CPU discards the prediction, and computation is slower. Unfortunately, discarded predictions have observable effects that can reveal system state and data, and because speculative attacks never cause the system to crash or misbehave in observable ways during their use, they're particularly useful for an attacker. For example, evaluating a pointer authentication instruction speculatively exposed timing differences in our original implementation of Pointer Authentication Codes (PAC), which would allow the valid signature to be isolated. During the design phase for Memory Integrity Enforcement, we identified and addressed the three speculative vulnerabilities that could undermine tag confidentiality.*
>
> 1. *First, when EMTE is active, requests to access memory cause the hardware to check tags. It's crucial that evaluating a tag-checking instruction speculatively doesn't expose timing differences that would allow an attacker to isolate the valid tag. From the start, we designed the Apple silicon implementation so that tag values cannot influence speculative execution in any way. Recently published security research (StickyTags, TikTag) demonstrates that the MTE implementation on Google's Pixel devices is vulnerable to this type of attack, allowing MTE to be bypassed in Google Chrome and the Linux kernel.*
>
> 2. *Second, allocators assign random tags to memory, and attackers must not be able to predict tag values that the system will choose. We address this issue by frequently re-seeding the underlying pseudo-random generator used to select new tags.*
>
> 3. *Third, Spectre variant 1 (V1) is a speculative-execution vulnerability that allows attackers to exploit conditional branches to leak data, including MTE tag values. To date, there has been no solution to this problem in consumer operating systems, because general Spectre V1 mitigations such as Speculative Load Hardening have a prohibitive CPU cost. The presence of EMTE leaves Spectre V1 as one of the last avenues available to attackers to help guide their attacks, so we designed a mitigation that limits the effective reach of Spectre V1 leaks at virtually zero CPU cost.*

So Arm began with MTE which Apple utilized once it was available. But its limitations caused Apple to work with Arm on EMTE – Enhanced MTE. But Apple was able to obtain sufficient real world experience with EMTE – specifically examining the many various ways it could be and was still being bypassed out on the field – that they then further enhanced the already enhanced memory tagged extension to make it even stronger. At that point, rather than calling it EEMTE they opted for MIE – Memory Integrity Enforcement.

Apple wrote:

> *Our mission with **Memory Integrity Enforcement** is to protect **all users by default** and to provide an extraordinary disruption to the exploitation of memory corruption vulnerabilities.*
>
> *To do so, we considered a wide set of threats, including some of the most challenging ones — such as side channels — and arrived at this extensive combination of features not present in other MTE implementations. Google took a great first step last year when they offered MTE to those who opt in to their program for at-risk users.* [Remember: Basic level MTE was added to the Arm architecture by Arm six years ago back in 2019] *But even for users who turn it on, the effectiveness of MTE on Android is limited by the lack of deep integration with the operating system, which is what distinguishes Memory Integrity Enforcement with its use of EMTE on Apple silicon.*
>
> *For the new A19 and A19 Pro chips to support Memory Integrity Enforcement, we dedicated an **extraordinary** amount of Apple silicon resources to security — more than ever before — including CPU area, CPU speed, and memory for tag storage. And to fully realize this hardware investment, we designed all of the new operating system elements of **MIE** jointly with our hardware work, including secure allocators, EMTE, and tag confidentiality protections.*
>
> *Because EMTE tag checking imposes a performance cost, we designed Memory Integrity Enforcement to take advantage of our secure allocators first and use EMTE to protect only smaller individual allocations within a type bucket, which software allocators can't defend on their own. Then, by knowing where and how we would deploy EMTE, we could accurately model the tag-checking demand of the operating system, and design our silicon to satisfy it. Our hardware implementation influenced additional software design decisions, reducing the overhead of tag checks even further. Importantly, deploying EMTE with this level of precision supports our strategy to provide as many memory safety improvements as possible to users on previous iPhone generations, which don't support EMTE.*
>
> *For the security evaluation of Memory Integrity Enforcement, we involved our offensive research team from the very beginning. From 2020 to 2025, they continuously analyzed and attacked the system — first conceptually, with theoretical exploitation avenues, then with practical attacks in simulated environments, and eventually on new hardware prototypes. Prolonged engagement from our offensive research team allowed us to identify and eradicate entire attack strategies and techniques before attackers could ever discover them, leading to a stronger, more mature feature from the outset.*
>
> *Our offensive research team identified where and how attackers are most likely to break into the system, and our deployment of Memory Integrity Enforcement is deeply guided by their findings. Notably, this includes making sure that this powerful new protection is available to third-party apps that are likely entry points for attackers — such as social networks, messaging apps, or any other app where a specific user can be targeted. Starting immediately with the launch of MIE, any developer can begin testing this powerful protection for their app, including EMTE on hardware that supports it, using the Enhanced Security settings in Xcode.*
>
> *The meticulous planning and implementation of Memory Integrity Enforcement made it possible to maintain synchronous tag checking for all the demanding workloads of our platforms, delivering groundbreaking security with minimal performance impact, while remaining completely invisible to users.*

So far, what we have has been a lot of very good theory, with improvements driven by feedback from successful real world experience. What would lead us to believe that the post-MIE world will actually be all that much different from the world before the iPhone 17 and the A19 chips? Apple says that they decided to get the jump on attackers by investing quite heavily in attacking themselves. They explained:

> *Memory Integrity Enforcement started with a deeply ambitious goal: to make it immensely more expensive and difficult to develop and maintain mercenary spyware attacks based on memory corruption against our platforms. While there’s no such thing as perfect security, MIE is designed to dramatically constrain attackers and their degrees of freedom during exploitation.*
>
> *Throughout the design and implementation of Memory Integrity Enforcement, our* ***offensive*** *research team evaluated our progress by looking at sophisticated exploit chains that were previously used against our own platform, recent vulnerabilities, and our own internal research.*
>
> *First, we worked on rebuilding and adapting previously seen exploit chains to systems protected by MIE. But it’s not sufficient to consider only previous chains that were developed before MIE existed, because attackers will surely adapt in reaction to these new protections. We therefore also evaluated a selection of more recent vulnerabilities that we expected would have the best chance of surviving MIE. For these, we meticulously enumerated all possible exploitation opportunities.*
>
> *Both approaches reached the same conclusion: Memory Integrity Enforcement vastly reduces the exploitation strategies available to attackers. Though memory corruption bugs are usually interchangeable, MIE cut off so many exploit steps at a fundamental level that it was not possible to restore exploitation chains by swapping in new bugs. Even with substantial effort, we were unable to rebuild any of these existing chains to work around MIE. The few memory corruption effects that remained are unreliable and do not give attackers sufficient momentum to successfully exploit these bugs.*

And then Apple proudly concludes:

> *iPhone’s industry-leading security has always meant that the vast majority of our users never face system-level attacks on their devices. Our work on memory safety is aimed primarily at the mercenary spyware and surveillance industry, which spends many millions of dollars to exploit memory corruption vulnerabilities and target a small number of individuals because of who they are and what they do. Over the past five years, we developed a comprehensive approach to memory safety that integrates the best of our hardware and software capabilities, and today’s announcement is the culmination of this ambitious vision. With the introduction of the iPhone 17 lineup and iPhone Air, we’re excited to deliver Memory Integrity Enforcement: the industry’s first ever, comprehensive, always-on memory-safety protection covering key attack surfaces — including the kernel and over 70 userland processes — built on the Enhanced Memory Tagging Extension (EMTE) and supported by secure typed allocators and tag confidentiality protections.*
>
> *Based on our evaluations pitting Memory Integrity Enforcement against exceptionally sophisticated mercenary spyware attacks from the last three years, we believe MIE will make exploit chains significantly more expensive and difficult to develop and maintain, disrupt many of the most effective exploitation techniques from the last 25 years, and completely redefine the landscape of memory safety for Apple products.*

Because of how dramatically it reduces an attacker's ability to exploit memory corruption vulnerabilities on our devices, we believe Memory Integrity Enforcement represents the most significant upgrade to memory safety in the history of consumer operating systems.

So, what Apple has essentially done is take the second generation of MTE, known as EMTE and to allow its best features to be used in an always-on, strongest protection configuration. This moves it from something that developers can use in the lab to find pre-release bugs to real-time protection that every Apple user will receive without any perceptible performance cost. This was made possible by moving the operation of the time consuming tag verification required by EMTE "in silico" to perform that work "synchronously" with new A19-generation chip hardware.

So let's summarize the features Apple has brought to market in their new iPhone 17 products:

- **Synchronous EMTE checking** — Tag verification occurs immediately before memory accesses and any tag mismatch causes a crash of the process to prevent its exploitation. This eliminates opportunities where malicious behavior could slip by due to delayed or asynchronous checking.

- **Always-On / System-Wide Deployment** — MIE is enabled by default across the entire kernel and for more than 70 userland processes. Previous and other systems are forced to rely upon optional or per-app memory tagging to reduce performance overhead.

- **Secure / Typed Allocators** — Apple's memory allocators have been updated to use type information to isolate objects by type, reduce "type confusion" style overlaps and help with placement so that allocations of different types get different tags and are less likely to be misuse targets.

- **Retagging / Memory Reuse Safety** — When memory is freed and reused, Apple's system ensures the free memory tag is changed, so stale pointers with old tags will no longer match.

- **Protection for Overflow Across Adjacent Allocations** — Because tags differ among adjacent allocations, a buffer overflow into a neighboring allocation, having a different tag, is guaranteed to immediately trigger a tag mismatch and abort the process to protect its user.

- **Non-Tagged Memory Access Rules** — Improving upon earlier tagging designs, accesses from tagged memory regions into non-tagged memory will now also be aware of tagging. This helps prevent attackers from bypassing protections by going through non-tagged memory.

- **Tag Confidentiality Enforcement** — Ensures that tag values are not leaked via side-channels, or speculative execution, and that kernel allocator data structures & tag storage are kept secret from user applications, even under adversarial conditions. This helps prevent attackers from guessing tags or using tag information to bypass protections.

- **Hardware / Silicon Support** — Due to the prohibitive performance overhead of doing any of this in software, part of MIE is its deep support from the A19 & A19 Pro chips which now have specific hardware investments for supporting EMTE at scale, including CPU area, tag storage, logic, etc., to eliminate performance overhead.

- **Tooling for Developers** — Application developers will now be able to access EMTE support in Xcode via the "Enhanced Security" feature so that apps can opt in / test their behavior.

It's not difficult to imagine what the team behind MIE, who had just spent the last five years of their lives perfecting all of this new super-hardening technology, were probably feeling with the news just two weeks ago of yet another successful exploit made against the hardware that they had already moved well past and poised to replace with an entirely new system that would almost certainly no longer fall victim to that, or probably nearly any other, attack.

Everyone listening to this podcast knows that where security is concerned we never say never.

We're never going to suggest that there will never be another successful system-level exploit against Apple's latest or future iOS and iPadOS platforms. But there's a distinct possibility that might be the case. We heard a while ago from a past early Apple hobbyist exploit developer who was lamenting that he had long ago hung up his spurs and was no longer attempting to find iPhone exploits because they had become insanely difficult to locate and engineer.

There will come a time – and we might be there today – when the cost to develop any new exploit, if it's even possible, is so high that even the highest-end most capable exploit developers join that earlier hacker in giving up on Apple and switch to more attackable platforms.