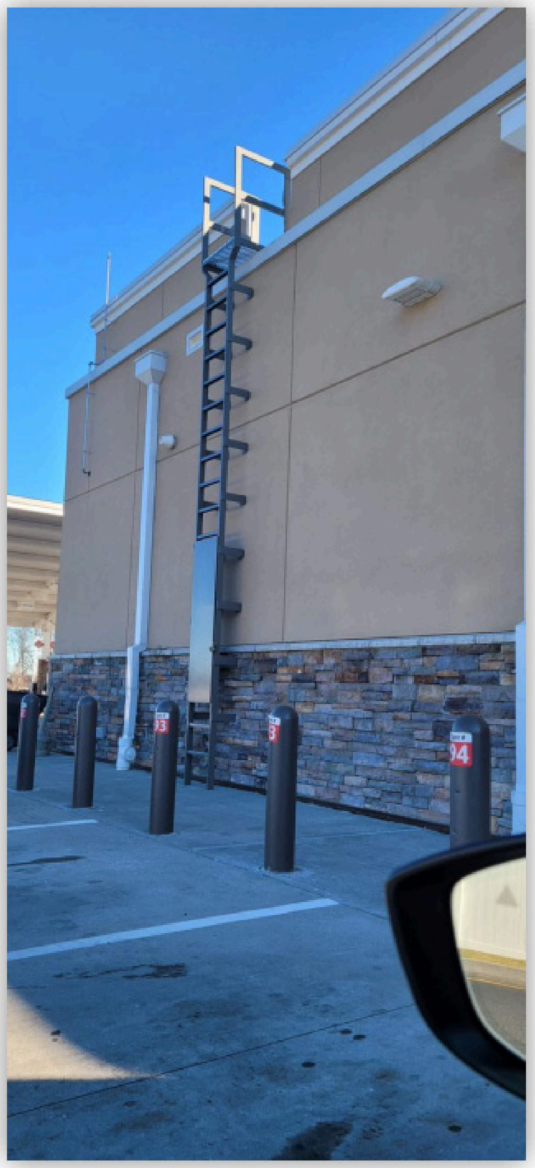
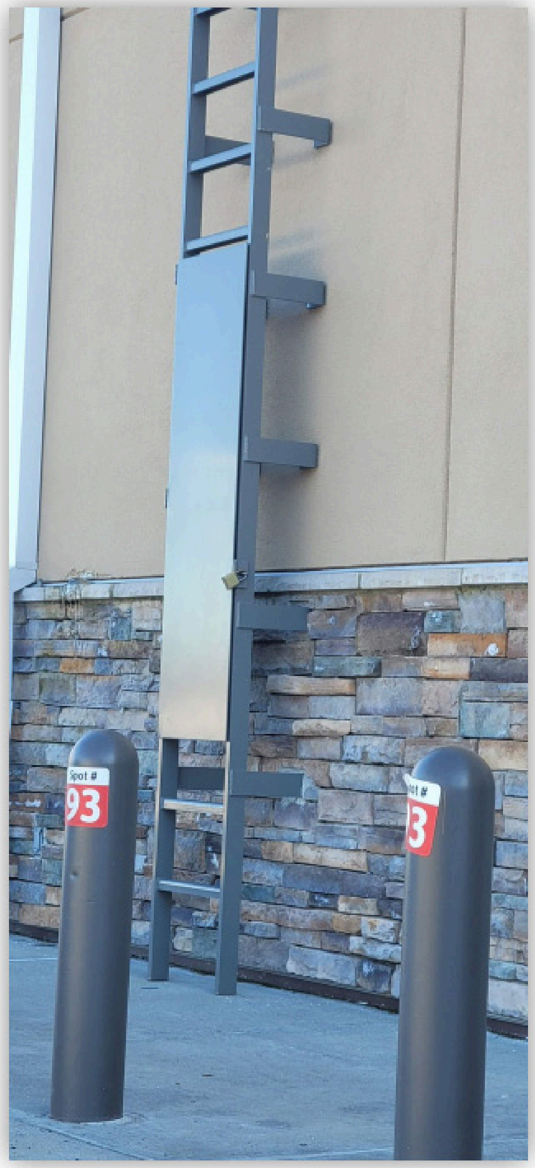
Security Now! #969 - 04-09-24
Minimum Viable Secure Product

This week on Security Now!

When is it far better for a security researcher to just keep their mouth shut? Are all Internet-based secure note exchanging sites created equal? What's been happening in the lucrative and slimy world of 0-days for pay? And what has NASA just learned about the state of Voyager 1? Something momentous has happened with SpinRite, and we're going to take a deep dive into an important industry initiative that just acquired an important new contributor.

"This fancy locked panel will keep everyone off the roof!"
(And so... here we have a failure of imagination.)

Security News

Out-of-support DLink NAS devices contain hard coded backdoor credentials

What year is it? It's 2024? And it's still the case that publicly accessible Internet-connected high profile devices are being found to have manufacturer hardcoded remote access credentials. I suppose that would be more difficult for us to believe if it wasn't so long ago that Cisco's internal audit of their own devices, which was spurred by repeated discovery of hard coded credentials in their equipment, kept turning up instance after instance of the same thing. So even the market leader at the high end of enterprise-grade networking gear was suffering from the inertia of the way they had always done things before.

It's taken some time, but we finally have a set of straightforward and widely recognized best practices:

1. **No default manufacturer-set credentials of any sort anywhere.**
   The first time a device's firmware boots and it sees that its credentials are blank, after it has had the chance to generate sufficient internal entropy for random number generation, it should create a very strong new default credential from scratch and present it to its user. The user will be free to then change that to whatever they want... but by default, any newly installed device will give its own security a head start by defaulting to something strong.
   If that makes technical support more difficult that's just tough. Once upon a time it might have been fine to default username and password to Admin/Admin and to print that on the quick start guide. But those days are long past. Security is inherently porous and the pressure against our porous security is steadily increasing.

2. **Physical access required.**
   Changing the security of anything in any dangerous direction, including that administrative username and password, **must** be accompanied by a physical button press, either beforehand to enable admin access or afterward to confirm the application of the new settings. Yes, that will make fully remote administration impossible. But it will also make fully remote administration by bad guys impossible. We have seen over and over again that it's not actually possible to have one without the other.

3. **Only absolute minimum functions publicly-exposed by default.**
   A router, for example, must have its WAN interface publicly exposed to be at all useful as a router. But it does not need a single additional service beyond that. So, therefore, not one additional service should be bound to that interface without explicit manual enabling accompanied by clear cautionary notices and, yes... pressing some physical button. And needless to say, when that service is instantiated it will start off with a fully random and strong username and password.

4. **Autonomous firmware updates received and applied by default.**
   All connected devices must ship from the factory with their auto-firmware-update system enabled by default. These devices should periodically ping a factory server to check for the availability of new firmware. The router's ping should include its current firmware version, and the cryptographically signed ping reply should provide the router with the current version and the urgency associated with moving from where it said it is to the latest.

The device's administrator can decide which levels of urgency are permitted to auto-update without any management oversight, and the device's user interface will strongly encourage the router to allow maximally urgent updates to be applied immediately, day or night.

5. **Firmware for devices must be maintained as long as they are in service.**
This is probably the trickiest one of the five. How long should a supplier be reasonably responsible for maintaining the firmware of an end-of-life and out-of-service device? This is a decision that is reasonably up to each supplier, but whatever their decision may be, it must be clearly and publicly stated so that prospective purchasers can plan accordingly.

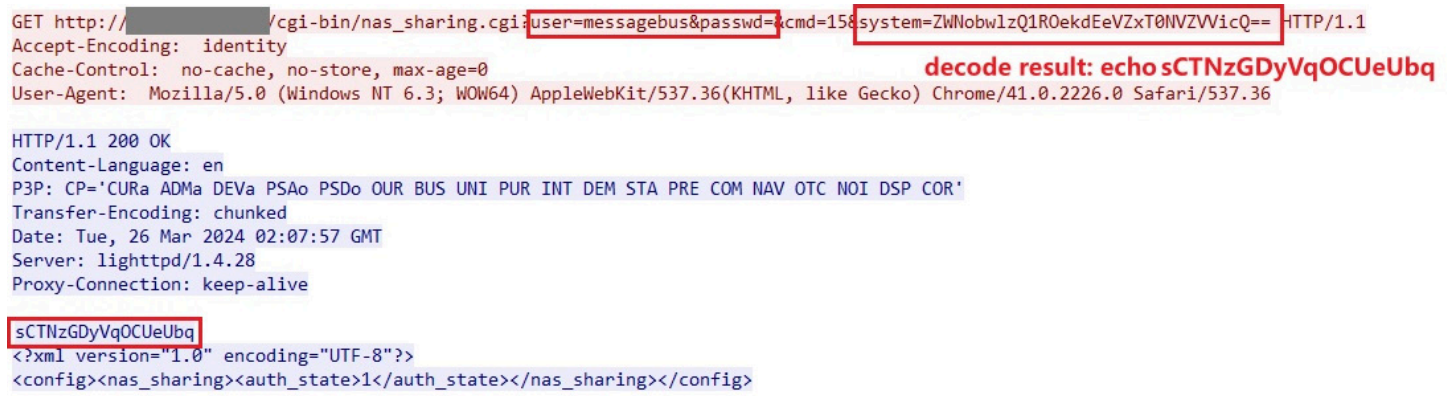
So we have five fundamental principles: (1) No default credentials, (2) physical access required to apply any dangerous configuration changes, (3) no unnecessary services enabled, (4) auto-firmware updates enabled by default, and (5) a clearly stated end-of-life ongoing support commitment. Just think back through all of the individual events, many of them having catastrophic consequences, that we've covered through this podcast's 19 years, every single one of which would have been prevented if these five fundamental principles of safe Internet connectivity had been in place.

And today, as luck would have it, we have news of yet another: A researcher discovered that D-Link Network Attached Storage (NAS) models DNS-320L, DNS-325, DNS-327L, and DNS-340L all share a distinguishing and horrifying characteristic: He found that the firmware they share had been hard coded by D- Link with the same, fixed, remote access backdoor username and password. And even the term "password" is being generous, since it's left blank. His GitLab handle is NetSecFish as in "Network Security Fish" and his page GitHub page (https://github.com/netsecfish/dlink) explains:

*The described vulnerability affects multiple D-Link NAS devices, including models DNS-340L, DNS-320L, DNS-327L, and DNS-325, among others. The vulnerability lies within the **nas_sharing.cgi** URI, which is vulnerable due to two main issues: **a backdoor** facilitated by hard coded credentials, and a command injection vulnerability via the "**system**" parameter. This exploitation could lead to arbitrary command execution on the affected D-Link NAS devices granting attackers potential access to sensitive information, system configuration alteration, or denial of service, by specifying a command.*

His subsequent Internet search discovered 92,589 of these devices currently exposed on the public Internet, with the highest concentration appearing at IP addresses in the United States. The big problem is that this family of D-Link NAS devices were first phased out of D-Link's product line on October 29th, 2017 and two years later on the same day in October 2019, all support for them was terminated by D-Link. Even so, it appears that our Network Security Fish gave D-Link no prior notice of his discovery before its publication, and no opportunity to decide whether they wished to alter their standing policy for out-of-service life products. Two weeks ago, on March 26th, he simply published everything he knew about this. Today, his public GitHub page thoroughly documents how a simple HTTP GET query, containing the username "messagebus" and a blank password can be used to cause any of these 92,589 currently online network attached storage devices to execute commands:

GET /cgi-bin/nas_sharing.cgi?user=messagebus & passwd= & cmd=15 & system=<BASE64_ENCODED_COMMAND_TO_BE_EXECUTED>



```
GET http://[redacted]/cgi-bin/nas_sharing.cgi?user=messagebus&passwd=&cmd=15&system=ZWNobwlzQ1ROekdEeVZxT0NVZVVicQ== HTTP/1.1
Accept-Encoding:  identity
Cache-Control:  no-cache, no-store, max-age=0
User-Agent:  Mozilla/5.0 (Windows NT 6.3; WOW64) AppleWebKit/537.36(KHTML, like Gecko) Chrome/41.0.2226.0 Safari/537.36

HTTP/1.1 200 OK
Content-Language: en
P3P: CP='CURa ADMa DEVa PSAo PSDo OUR BUS UNI PUR INT DEM STA PRE COM NAV OTC NOI DSP COR'
Transfer-Encoding: chunked
Date: Tue, 26 Mar 2024 02:07:57 GMT
Server: lighttpd/1.4.28
Proxy-Connection: keep-alive

sCTNzGDyVqOCUeUbq
<?xml version="1.0" encoding="UTF-8"?>
<config><nas_sharing><auth_state>1</auth_state></nas_sharing></config>
```

We see from his packet capture that he issued a command to a machine at a redacted address – and even doing that would be controversial unless the machine was his – a command to which the machine replied with a valid HTTP/1.1 200 OK. It included other reply headers and XML in the body of its reply. We learn that, among other things, the D-Link NAS devices are running "lighttpd" web server v1.4.28.

I'm unhappy with what D-Link has been found to have done. We'll get to them in a minute. But it's not clear to me that anyone is helped by "Fish's" public disclosure, and that 92,589 people and their networks and their data stand to be attacked and compromised as a result. Before he made this needless disclosure, it ***might*** have been that someone ***might*** have eventually stumbled onto this too – and it's true that we don't know that it hasn't happened. But we do know that it was never public before and that now it is. And another lesson we've learned through the years of shared experience on this podcast is that the height from which the fruit is hanging matters a lot. Far more low hanging fruit is plucked than high-hanging fruit. There is an endless supply of script kiddies able to issue a series of simple HTTP GET queries; how are any of them going to be able to resist any target as tempting as this? This is the definition of a script kiddie vulnerability. So this person's disclosure two weeks ago has, without any question, compromised the security of 92,589 still online, still functioning, still in use, and still sitting on other people's networks, D-Link network attached storage devices.

https://googleprojectzero.blogspot.com/p/vulnerability-disclosure-policy.html

At the same time, a fair question is "What would Google do?" And by "Google" I'm referring to Google's Project Zero with their famous 90-day disclosure policy where manufacturers who Google has notified have 90 days to produce a patch for a discovered flaw before Google releases the flaw's technical details. And as we know, many companies have felt the pressure to fix a vulnerability that Project Zero had found in their products. But in this instance, the answer to the question "what would Google do?" is that I don't know. For one thing, these are long out-of-service devices that are no longer being maintained by their supplier. And their supplier has made it very clear that they have no intention of addressing this issue – ever. But moreover, it appears that this is not actually a bug. This was an undocumented backdoor that D-Link embedded into their devices for some unclear purpose. So because it's old, out of service and not a bug, I'm pretty certain that Google's Project Zero would never even consider taking this up in the first place.

So this brings us back to D-Link, a well known, reputable, popular, Taiwanese network hardware manufacturer that's been around since 1986. I've purchased a bunch of D-Link equipment, mostly network hubs, I think, through the years. So I suppose that at least among those of us here and others who learn of this past behavior, this tarnishes their brand.

With devices that were discontinued six and a half years ago and that went out of support four and a half years ago, we would have to assume that they have no way of remotely updating their firmware, even if they wished to. If the owners of those devices had registered them, D-Link could conceivably reach out to them... but what would D-Link say? *"Uhhhh... you know that NAS that you purchased from us ten years ago? Well, uh... we had planted a secret remote access backdoor into it and it was recently discovered and has been made public. So, you know, we're really sorry about that and you should probably definitely immediately disconnect your D-Link device from the Internet. Have a nice day... and would you like to consider buying a newer NAS from us? We have a bunch of nice shiny new ones for you to consider."*

And I'd be surprised if anything could be done on the legal front. This was a design decision made by D-Link. They may have some justifiable rationale for it. The username "messagebus" kind of suggests that perhaps these devices could be set up in a cluster, where this "messagebus" allowed them to interoperate in some fashion. So it was a bad design not to control who or what could use this interface, but it wasn't an obvious crime on D-Link's part.

But speaking of crime, here it comes. Yesterday, entirely predictably, ArsTechnica's headline read ***"Critical takeover vulnerabilities in 92,000 D-Link devices under active exploitation."*** Yes, that didn't take long did it? Ars reports:

On Monday, researchers said their sensors began detecting active attempts to exploit the vulnerabilities starting over the weekend. Greynoise, one of the organizations reporting the in-the-wild exploitation, said in an email that the activity began around 02:17 UTC on Sunday. The attacks attempted to download and install one of several pieces of malware on vulnerable devices depending on their specific hardware profile. One such piece of malware is flagged under various names by 40 endpoint protection services.

Security organization Shadowserver has also reported seeing scanning or exploits from multiple IP addresses but didn't provide additional details.

The vulnerability pair, found in the nas_sharing.cgi programming interface of the vulnerable devices, ***provide an ideal recipe for remote takeover.*** *The first, tracked as CVE-2024-3272 and carrying a severity rating of 9.8 out of 10, is a backdoor account enabled by credentials hardcoded into the firmware. The second is a command-injection flaw tracked as CVE-2024-3273 and has a severity rating of 7.3. It can be remotely activated with a simple HTTP GET request.*

Netsecfish, the researcher who disclosed the vulnerabilities, demonstrated how a hacker could remotely commandeer vulnerable devices by sending a simple set of HTTP requests to them.

So, thank you and congratulations, “Network Security Fish.” What a nice public service you have performed for 92,589 D-Link users whom you have just single-handedly turned into victims.

Before long, if not already, those 92,589 D-Link devices will be infected with malware, if they aren’t all already – after all, today’s Tuesday. And, as I mentioned at the start of this adventure, the Internet scan distribution shows that networks in the United States contain more of them than any other region. I wonder whose networks those devices may be sitting on, and I wonder who might be interested in finding out.

I suppose the final takeaway lesson for us is that complex devices of this sort that are no longer being actively supported cannot be used safely – at least not in settings, such as connected to the Internet, where the security risk is high. Of course, that’s easily said, right? But it’s difficult to retire a perfectly working device for no obvious reason other than that its manufacturer is no longer actively supporting it. A mature policy would ideally rotate such devices into less security sensitive roles. Give them a place inside the network, behind firewalls where they can live out their lives in peace while continuing to be productive.

I very much hope that the 92,589 owners of these surviving NAS devices do not experience much hardship as a consequence of “Network Security Fish”’s needless, pointless & destructive disclosure. It should be clear that there are times when it’s far better to just say nothing.

Privnote is not so “Priv”

Last Thursday, Brian Krebs of https://krebsonsecurity.com fame, posted a piece that just makes you shake your head. He wrote:

A cybercrook who has been setting up websites that mimic the self-destructing message service privnote.com accidentally exposed the breadth of their operations when they threatened to sue a software company. The disclosure revealed a profitable network of phishing sites that behave and look like the real Privnote, except that [get this] any messages containing cryptocurrency addresses will be automatically altered with a different payment address controlled by the scammers.

Brian then explains:

Launched in 2008, privnote.com employs technology that encrypts each message so that even Privnote itself cannot read its contents. And it doesn’t send or receive messages. Creating a message merely generates a link. When that link is clicked or visited, the service warns that the message will be gone forever after it is read.

Privnote’s ease-of-use and popularity among cryptocurrency enthusiasts has made it a perennial target of phishers, who erect Privnote clones that function more or less as advertised but also quietly replace their own cryptocurrency payment addresses when a note is created that contains crypto wallet addresses.

Last month, a new user on GitHub named fory66399 lodged a complaint on the “issues” page for MetaMask, a software cryptocurrency wallet used to interact with the Ethereum blockchain. Fory66399 insisted that their website — privnote[.]co — was being wrongly flagged by

MetaMask's "eth-phishing-detect" list as malicious.

Fory66399 wrote: "We filed a lawsuit with a lawyer for dishonestly adding a site to the block list, damaging reputation, as well as ignoring the moderation department and ignoring answers! Provide evidence or I will demand compensation!"

So MetaMask's lead product manager Taylor Monahan replied by posting several screenshots of privnote[.]co showing that the site did indeed swap out any cryptocurrency addresses.

After being told where they could send a copy of their lawsuit, Fory66399 appeared to become flustered.

Brian's piece continues with one of his terrific deep research dives into all of the details and he uncovers a large network of all very similar clone websites.

What I found interesting about this was that this is not hacking some fancy new blockchain technology implementation to steal a windfall of $50 million dollars all at once. Instead, this is stealing individual small cryptocurrency transactions from cryptocurrency end users. And you can imagine that dialog: *"I haven't received the drugs I sent you the money for."* ... *"What do you mean you never received the payment? I sent it right after I received the eMail with your wallet address and the money was taken from my wallet. If you didn't get it then where did it go?"*

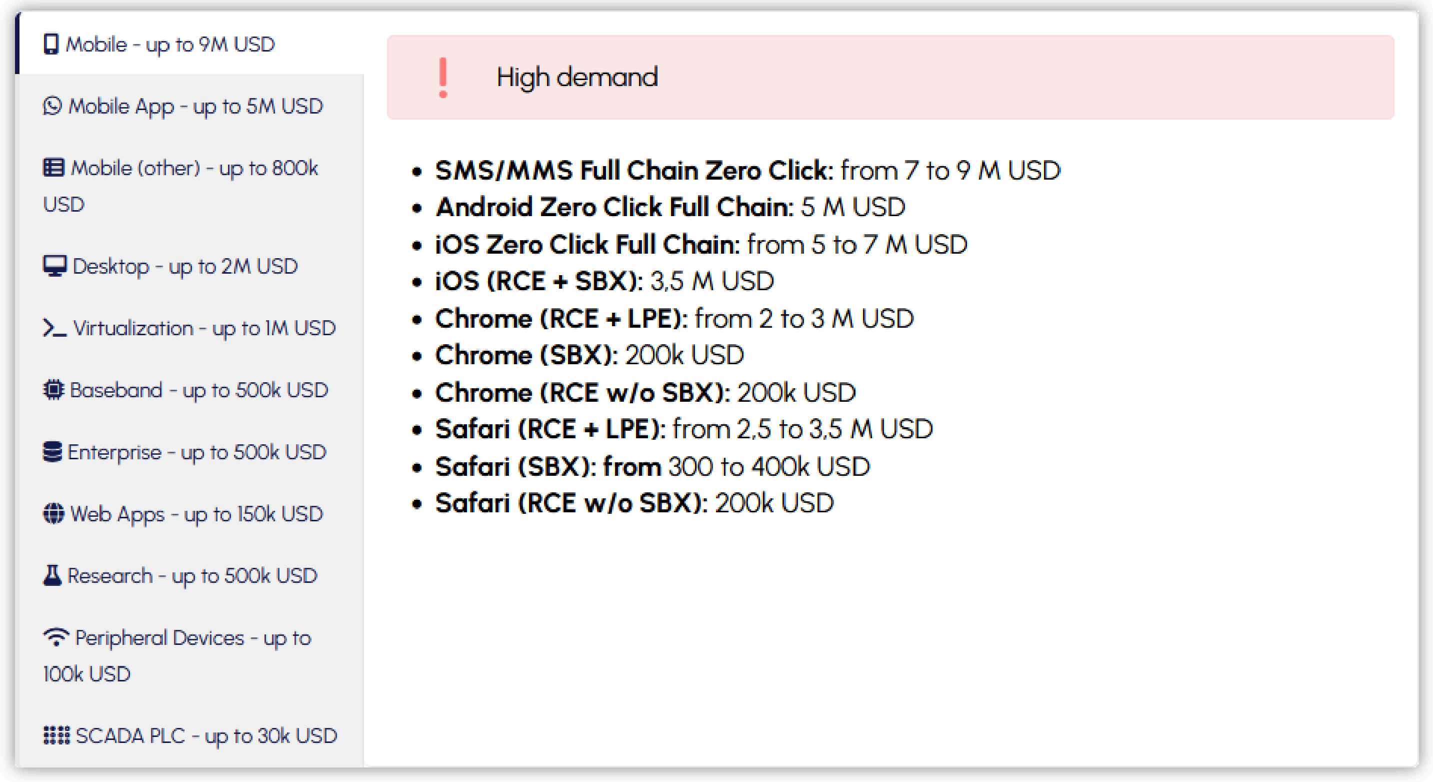
Well ... gee ... it appears likely that it may have made its way into some Russian's pocket, perhaps because you were not paying close attention and used **privnote.co** or **pirvnota.com** or **privatemessage.net** or **privatenote.io** or **tornote.io** or **privnode.com** or **privnate.com** or **prevnóte.com**. Believe it or not, Brian's research traced each of those privnote.com malicious copycat domains back to a handful of apparent Russians who have been enjoying fleecing the corrupt Western capitalists.

We know that it's entirely possible to create a simple website that encrypts a visitor's note on their PC so that it cannot be decrypted except by another third party. The problem is, the technology required to do this is not readily visible and auditable by the site's lay user. The site clearly and cleanly claims that they're unable to read anything that's being sent. And the unwitting user has heard of such sites, like privnote.com, that arrange to do just that. It's got a terrific reputation. And privnote.io looks the same, so it's probably just the same people who also got that cool .IO domain. And, after all, it says that it cannot read anything that's sent. So let's just copy and paste a wallet address. What could possibly go wrong? What, indeed.

Crowdfense is willing to pay millions

We talked a lot about Zerodium in the past. They're the folks who offer extremely large bounties for new and unknown 0-click vulnerabilities. And unlike the good guys at HackerOne or ZeroDay, these creeps resell these 0-days, doubtless at a significant profit, to unknown but certainly big time buyers such as Israel's NSO Group for use by the Pegasys spyware and almost certainly to governments and intelligence services. In other words, the platform publishers such as Apple and Google are the last to learn of these exploits.

Well now, on Saturday, TechCrunch brings news of a newcomer named "Crowdfense" that intends to give Zerodium a run for its money... and this is really not a market where we'd like to see competition flourishing. We'd prefer that it didn't exist at all.



Mobile - up to 9M USD

Mobile App - up to 5M USD

Mobile (other) - up to 800k USD

Desktop - up to 2M USD

Virtualization - up to 1M USD

Baseband - up to 500k USD

Enterprise - up to 500k USD

Web Apps - up to 150k USD

Research - up to 500k USD

Peripheral Devices - up to 100k USD

SCADA PLC - up to 30k USD

! High demand

- **SMS/MMS Full Chain Zero Click:** from 7 to 9 M USD
- **Android Zero Click Full Chain:** 5 M USD
- **iOS Zero Click Full Chain:** from 5 to 7 M USD
- **iOS (RCE + SBX):** 3,5 M USD
- **Chrome (RCE + LPE):** from 2 to 3 M USD
- **Chrome (SBX):** 200k USD
- **Chrome (RCE w/o SBX):** 200k USD
- **Safari (RCE + LPE):** from 2,5 to 3,5 M USD
- **Safari (SBX): from** 300 to 400k USD
- **Safari (RCE w/o SBX):** 200k USD

I have a screen shot of Crowdfence's current offering lineup. At the top of the heap are SMS/MMS full chain 0-click compromises, the discovery of which would net someone selling it somewhere between $7 and $9 million US dollars. Full chain meaning something gets the entire job done, not just a "oh, look, it crashed." A full chain 0-click for Android brings in $5 million whereas the same thing for iOS is priced at between $5 and $7 million. And these are all just within the top paying "Mobile Platform" category. Crowdfense is also interested in Mobile Apps, other mobile, Desktop, Virtualization, Baseband radio, Enterprise, Web Apps, and more.

TechCrunch's headline was *"Price of zero-day exploits rises as companies harden products against hackers"* with the subheading: *"A startup is now offering millions of dollars for tools to hack iPhones, Android devices, WhatsApp, and iMessage."* They write:

Tools that allow government hackers to break into iPhones and Android phones, popular software like the Chrome and Safari browsers, and chat apps like WhatsApp and iMessage, are now worth millions of dollars — and their price has multiplied in the last few years as these products get harder to hack.

On Monday, startup Crowdfense published its updated price list for these hacking tools, which are commonly known as "zero-days" because they rely on unpatched vulnerabilities in software that are unknown to the makers of that software. Companies like Crowdfense and one of its competitors, Zerodium, claim to acquire these zero-days with the goal of reselling them to other organizations, usually government agencies or government contractors, which

claim they need the hacking tools to track or spy on criminals.
Crowdfense is now offering between $5 million and $7 million for zero-days to break into iPhones; up to $5 million for zero-days to break into Android phones; up to $3 million and $3.5 million for Chrome and Safari zero-days, respectively; and $3 million to $5 million for WhatsApp and iMessage zero-days.

In its previous price list, published in 2019, the highest payouts that Crowdfense was offering were $3 million for Android and iOS zero-days.

The increase in prices comes as companies like Apple, Google, and Microsoft are making it harder to hack their devices and apps, which means their users are better protected.

Okay. In other words, as 0-days become more rare they naturally become more valuable. It's good news for everyone that they are becoming more rare. TechCrunch continues:

Dustin Childs, the head of threat awareness at Trend Micro's Zers-Day Initiative said: "It should be harder year over year to exploit whatever software we're using, whatever devices we're using." Unlike Crowdfense and Zerodium, ZDI pays researchers to acquire zero-days, then reports them to the companies affected with the goal of getting the vulnerabilities fixed.

Shane Huntley, the head of Google's Threat Analysis Group (TAG), tracks hackers and the use of zero-days. He said: "As more zero-day vulnerabilities are discovered by threat intelligence teams like Google's, and platform protections continue to improve, the time and effort required from attackers increases, resulting in an increase in cost for their findings."

In a report last month, Google said that last year in 2023 it saw hackers use a total of 97 zero-day vulnerabilities in the wild. And the various Spyware vendors – like the NSO Group – which often work with zero-day brokers, were responsible for three quarters of all zero-days targeting Google products and Android.

People in and around the zero-day industry agree that the job of exploiting vulnerabilities is getting harder.

David Manouchehri, a security analyst with knowledge of the zero-day market, said that "hard targets like Google's Pixel and the iPhone have been becoming harder to hack every year. I expect the cost to continue to increase significantly over time."

Paolo Stagno, the director of research at Crowdfense, told TechCrunch: "The mitigations that vendors are implementing are working, and it's leading the whole trade to become much more complicated, much more time-consuming, and so clearly this is then reflected in the price."

The first time I read that I thought "trade"? What trade? Then I realized that they're calling this 0-day vulnerability hunting a trade. I suppose it is, though it feels like ransomware gangs talking about their "profit". Profit? How about theft through extortion. How is that profit? But I suppose that it is, sadly, profitable. Though it hardly seems earned. Anyway, the Stagno guy from Crowdfence explained:

In 2015 or 2016, it was possible for only one researcher to find one or more zero-days and develop them into a full-fledged exploit targeting iPhones or Androids. Now "this is almost *impossible," as it requires a team of several researchers, which also causes prices to go up. Crowdfense currently offers the highest publicly known prices to date outside of Russia, where a company called Operation Zero announced last year that it was willing to pay up to $20 million for tools to hack iPhones and Android devices. The prices in Russia, however, may be inflated because of the war in Ukraine and the subsequent sanctions, which could discourage or outright prevent people from dealing with a Russian company.*

Outside of the public view, it's possible that governments and companies are paying even higher prices.

David Manouchehri previously worked at Linchpin Labs, a startup that focused on developing and selling zero-days. Linchpin Labs was acquired by U.S. defense contractor L3 Technologies (now known as L3Harris) in 2018. (That's encouraging.) Anyway, David said: "The prices Crowdfense is offering researchers for individual Chrome [Remote Code Execution] and [Sandbox Escape] exploits are below market rate from what I have seen in the zero-day industry."

Alfonso de Gregorio, the founder of Zeronomicon, an Italy-based startup that acquires zero-days, agreed, telling TechCrunch that prices could "certainly" be higher.

Zero-days have been used in court-approved law enforcement operations. In 2016, the FBI used a zero-day provided by a startup called Azimuth to break into the iPhone of one of the shooters who killed 14 people in San Bernardino, according to The Washington Post. In 2020, Motherboard revealed that the FBI — with the help of Facebook and an unnamed third-party company — used a zero-day to track down a man who was later convicted for harassing and extorting young girls online.

There have also been several cases where zero-days and spyware have allegedly been used to target human rights dissidents and journalists in Ethiopia, Morocco, Saudi Arabia, and the United Arab Emirates, among other countries with poor human rights records. There have also been similar cases of alleged abuse in democratic countries like Greece, Mexico, Poland, and Spain. (Neither Crowdfense, Zerodium, or Zeronomicon, have ever been accused of being involved in similar cases.)

Zero-day brokers, as well as spyware companies like NSO Group and Hacking Team have often been criticized for selling their products to unsavory governments. In response, some of them now pledge to respect export controls in an effort to limit potential abuses from their customers.

Stagno said that Crowdfense follows the embargoes and sanctions imposed by the United States — even if the company is based in the United Arab Emirates. For example, Stagno said that the company wouldn't sell to Afghanistan, Belarus, Cuba, Iran, Iraq, North Korea, Russia, South Sudan, Sudan, and Syria — all on U.S. sanctions lists.

"Everything the U.S. does, we are on the ball," Stagno said, adding that if an existing customer gets on the U.S. sanctions list, Crowdfense would abandon it. "All the companies and governments directly sanctioned by the USA are excluded."

At least one company, spyware consortium Intellexa, is on Crowdfense's particular blocklist.

Of Intellexa Stagno said: "I can't tell you whether it has been a customer of ours and whether it has stopped being one. However, as far as I am concerned now at this moment Intellexa *could not be a customer of ours."*

In March, the U.S. government announced sanctions against Intellexa's founder Tal Dilian as well as a business associate of his, the first time the government imposed sanctions on individuals involved in the spyware industry. Intellexa and its partner company Cytrox was also sanctioned by the U.S., making it harder for the companies, as well as the people running it, to continue doing business.

Intellexa's spyware has been reported to have been used against U.S. congressman Michael McCaul, U.S. senator John Hoeven, and the president of the European Parliament Roberta Metsola, among others.

De Gregorio, the founder of Zeronomicon, declined to say who the company sells to. On its site, the company has published a code of business ethics, which includes vetting customers with the goal of avoiding doing business "with entities known for abusing human rights," and respecting export controls.

Reading about these so-called export controls one has to wonder how difficult it would be for any major country on the US sanctions list to establish a behind the scenes relationship with another company in a non-sanctioned country to use as a middleman.

In any event, I thought that checking in on the state of the "0-day market" industry would be useful. While it may not be good news that prices are increasing, since that increases incentives to find the fewer and fewer remaining 0-days. But the fact that prices are rising because these remaining 0-days are becoming ever more scarce is certainly good news.

Miscellany

I have one bit of miscellany to share: Last Thursday, NASA updated the world with news of the status of the intrepid Voyager 1 spacecraft. The headline of their posting was "Engineers Pinpoint Cause of Voyager 1 Issue, Are Working on Solution". They explained:

Engineers have confirmed that a small portion of corrupted memory in one of the computers aboard NASA's Voyager 1 has been causing the spacecraft to send unreadable science and engineering data to Earth since last November. Called the flight data subsystem (FDS), the computer is responsible for packaging the probe's science and engineering data before the telemetry modulation unit (TMU) and radio transmitter send the data to Earth.

In early March, the team issued a "poke" command to prompt the spacecraft to send back a readout of the FDS memory, which includes the computer's software code as well as variables. Using the readout, the team has confirmed that about 3% of the FDS memory has been corrupted, preventing the computer from carrying out normal operations.

The team suspects that a single chip responsible for storing part of the affected portion of the FDS memory isn't working. Engineers can't determine with certainty what caused the issue. Two possibilities are that the chip could have been hit by an energetic particle from space or that it simply may have worn out after 46 years.
Although it may take weeks or months, engineers are optimistic they can find a way for the

FDS to operate normally without the unusable memory hardware, which would enable Voyager 1 to begin returning science and engineering data again.

SpinRite

I was tempted to name this podcast “SpinRite 6.1” because what happened Sunday afternoon means so much to me. But since it wouldn’t mean that much to the rest of our listeners, it didn’t seem appropriate. What happened on Sunday is that I finally updated GRC for the first time ever to begin offering 6.1 as GRC’s official SpinRite. 6.1 is finally what new purchasers will receive instead of 6.0. So this is huge for me. I’ve been living with my commitment to offer 6.1 for so long now, it’s been more than a decade, and I’ve felt guilty whenever I’ve stolen time from that, that the realization is still dawning on me that this project is nearing completion. I have GRC’s eMail system and then SpinRite documentation to finish, but that’s all going to be a joy.

One thing I wanted to mention, though, to our listeners is that after sharing the experience my wife and I had with her Dell laptop during last weeks podcast, I noticed an uptick in SpinRite sales – in other words, more “Yabba Dabba Doo’s.” I’m fine with that since many people have reported significantly improved system performance after running SpinRite at Level 3 over an SSD which is almost certainly tied to significantly improved long-term SSD reliability. So I can stand behind the benefits that people are likely to see. My only concern was that until the afternoon of this past Sunday, April 7th, unless those new purchasers knew to then go to GRC’s /prerelease.htm page, they would have obtained v6.0. So I wanted to make sure that anyone who may have purchased SpinRite while the website was still showing 6.0 knew that they can put the transaction code they received in their purchase receipt into either our sales or support pages to display their purchase transaction page which now, finally, offers SpinRite 6.1 for download.

The second thing I wanted to share was inspired by someone using the handle “The Big Bear” who posted into GRC’s SpinRite newsgroup. After running SpinRite v6.1 on two Macs he posted:

“And now I have two Macs tested and refreshed. And it makes quite a difference. The frequency with which the colourful beachball had shown was starting to worry me and it has all but disappeared now. And the startup times feel halved at least. Wish I had measured it before and after. Documentation is not my strong suit, but I will put it on my todo list , to write an update mentioning the extra hoops required with the latest Sonoma macOS.”

The reason I’m mentioning that, aside from it being another instance of welcome feedback about my use of the past three and a half years, is that while FreeDOS and SpinRite will run on Intel-based Macs, getting them to boot from a CD or USB can be a bit tricky. So I wanted to remind any would-be Mac purchasers that this is the reason I created GRC’s freeware named “Bootable” in favor of “DOS Boot”. If you can get Bootable to congratulate you on your success in booting it, then exactly the same path can be taken with SpinRite... and Bootable can be freely downloaded at any time. GRC’s web forums at https://forums.grc.com contain a growing knowledge base of help for Mac users.

Minimum Viable Secure Product *(MVSP)*

Our beloved industry is slowly (very slowly) getting its act together. But it IS happening and I've been encouraged by some recent news surrounding the *"Minimum Viable Secure Product"* effort. The group's list of contributors has been growing and it now includes some well known names such as SaleForce, Google, Okta, Slack, Vanta and about 20 others. The reason this is today's primary podcast topic—aside from the whole thing being an extremely worthwhile effort, which it is—is due to the announcement of the effort's latest member, made last Thursday. The posting reads:

> *Today, we're excited to announce that CISA is joining the Minimum Viable Secure Product (MVSP) Working Group. Since launching CISA's global Secure by Design initiative last year, we've received a tremendous amount of feedback (including through our Request for Information that recently closed!).*
>
> *One of the key questions we've gotten is how organizations consuming software can ask the right questions of their software manufacturers. Such a "secure by demand" approach is crucial to drive the uptake of secure by design principles and practices.*
>
> *Too often, procurement questionnaires are filled with long lists of questions which don't always correlate with positive security outcomes. In order to achieve a future where technology is secure by design, companies buying software should have simple and to the point questions for their vendors.*
>
> *The MVSP is an important step forward toward this goal. MVSP offers a simple checklist that organizations can use to strengthen security at multiple stages – to review their software vendors' security during procurement, as a self-assessment tool for their own software, as part of their software development lifecycle (SDLC), or as contractual controls – which can go a long way towards helping ensure secure by design principles are followed. We're excited to join the MVSP working group to help shape the direction of the initiative going forward. The MVSP is composed of a broad coalition of technology manufacturers, and the working group is open for anyone to join.*

Reading through the MVSP's checklist put a smile on my face, as I imagine it will for our listeners since we have carefully examined many of these issues here. What everyone is hoping is that powerful technology procurement bodies — like all of the various branches of the U.S. government, including both its civilian and military bureaucracies — might start making an adherence to these principles more than just requests, but make them mandatory for future purchases.

The MVSP web site at: https://mvsp.dev/ explains their mission by writing:

> *Minimum Viable Secure Product (MVSP) is a list of essential application security controls that should be implemented in enterprise-ready products and services. The controls are designed to be simple to implement and provide a good foundation for building secure and resilient systems and services. MVSP is based on the experience of contributors in enterprise*

> *application security and has been built with contributions from a range of companies.*
>
> *We recommend that all companies building enterprise software and services, or otherwise handling sensitive information, implement the MVSP controls and, where possible, go well beyond them in their application security programs.*
>
> *We welcome constructive feedback to help us continue to improve MVSP and provide a control set that meets the needs of its users.*

Minimum Viable Secure Product (MVSP) is a list of essential application security controls that should be implemented in enterprise-ready products and services.

The controls are designed to be simple to implement and provide a good foundation for building secure and resilient systems and services.

We recommend that all companies building enterprise software and services, or otherwise handling sensitive information, implement the MVSP controls and, where possible, go well beyond them in their application security programs.

So let's see what will put a smile on everyone's faces.

Minimum Viable Secure Product (MVSP) is a list of essential application security controls that should be implemented in enterprise-ready products and services.

The controls are designed to be simple to implement and provide a good foundation for building secure and resilient systems and services.

We recommend that all companies building enterprise software and services, or otherwise handling sensitive information, implement the MVSP controls and, where possible, go well beyond them in their application security programs.

1 Business controls

1.1 External vulnerability reports FAQ

- Publish a vulnerability disclosure policy that outlines the testing scope, provides a legal safe harbor, and gives contact details for security reports. These are all things we've discussed in the past. Vulnerability researchers should be free to research while being legally protected from retribution or reprisals if they hack a supplier's product without malicious intent for the sole purpose of discovering and responsibly reporting vulnerabilities.

- Fleshing that out, the MVSP lays out the required components as: Develop and document procedures for triaging and remediating reported vulnerabilities, respond to reports within a reasonable time frame and patch vulnerabilities similarly.

- They also suggest contracting with a security vendor to perform comprehensive penetration tests of products, services, and dependent systems, at least once per year.

- Notify relevant parties about any security breach that affects sensitive information no later than 72 hours upon discovery, and upon learning any additional details of the breach. Consider reporting the breach to relevant national cybersecurity agencies in line with local guidance and regulations. In any such reporting include the nature of the breach, relevant contact information, the consequences of the breach and the measures taken and needing to be taken to remediate the issue.

- Be certain to sanitize all storage media holding unencrypted production data.

- Implement single sign-on using modern, maintained, and industry-standard protocols for all customers at no additional cost.

- Redirect traffic from HTTP protocol (port 80) to HTTPS (port 443). Exceptions to this are internally secure protocols designed to run over unencrypted connections, such as OCSP, the Online Certificate Status Protocol.

- Include the Strict-Transport-Security header with a long max-age value and set authentication cookies as "Secure" – this prevents the browser from ever sending them out over non-encrypted connections.

- Apply appropriate HTTP security headers to reduce the application attack surface and limit post exploitation. These should include setting a minimally permissive Content Security Policy, limiting the ability to in-line frame the application by enabling framing controls with X-Frame-Options or CSP frame-ancestors and disable caching for APIs and endpoints that return sensitive data.

And, of course the MVSP had a lot to say about password policies. They wrote: "If password authentication is used in addition to single sign-on, then:

- Do not limit the permitted characters that can be used
- Do not limit the length of the password to anything below 64 characters
- Do not use secret questions as a sole password reset requirement
- Require email verification of a password change request
- Require the current password in addition to the new password during password change
- Store passwords in a hashed and salted format using a memory-hard or CPU-hard one-way hash function
- Enforce appropriate account lockout and brute-force protection on account access
- Do not provide default passwords for users or administrators

And what about the use of security-sensitive 3rd-party libraries? They say:

- Use modern, maintained, and industry-standard frameworks, template languages, or libraries that systemically address implementation weaknesses by escaping the outputs and sanitizing the inputs.

- Ensure third-party dependencies are maintained and up-to-date, with security relevant updates having a severity score of "medium" or higher applied in line with your application patching schedule

- Upon becoming aware of a Known Exploited Vulnerability (that's CISA's KEV collection) affecting a third-party dependency, the patch should be prioritized.

- Where dependency patching or upgrades are not possible, equivalent mitigations should be implemented for all components of the application stack

And, hardly surprising, they're big fans of logging, recommending that logs be kept of all Authentication events – both their successes and failures. Log all security relevant configuration changes – including the disabling of logging! – and log application owner access to customer data to provide access transparency. These logs must include user ID, IP address, valid timestamp, type of action performed, and the object of the action. Logs must be stored for at least 30 days at no additional charge to the client or customer, and should not contain sensitive data or payloads.

Though it barely needs saying, they recommend using current, maintained, industry-standard means of encryption to protect sensitive data in transit between systems, and at rest in all online data storages and backups.

And when vulnerabilities are found:

- Produce and deploy patches to address application vulnerabilities that materially impact security within 90 days of discovery.

- For vulnerabilities with evidence of active exploitation, production and deployment of patches should be prioritized. (Okay, that one was a "Duh!")

- Publish a security bulletin that details the vulnerability and its root cause if the remedy requires action from customers

We've seen many instances where permissions were far too open. We've noted that nothing breaks when everyone can access something, so it's too often discovered – after a breach of some kind – that the source of the breach was overly permissive access controls. So they write:

- Limit sensitive data access exclusively to users with a legitimate need. The data owner must authorize such access

- Deactivate redundant accounts and expired access grants in a timely manner

- Perform regular reviews of access to validate need to know

- Ensure remote access to customer data or production systems requires the use of Multi-Factor Authentication

Yeah, every one of those is obvious. And who wants to make time to sit and review access permissions? Because it's so boring, it might not be productive and a million other actually important things need to be done, it never happens. But the result of the resulting inattention are breaches that could have been prevented if someone in the organization was willing to be bored for a while.

They advise that it's also important to maintain a list of third-party companies with access to customer data which can and will be made available to clients and business partners upon request. And since that list might be embarrassing if, for example, it were to contain contractors who were no longer affiliated, this provides some incentive to remove access once relationships terminate. Again, we've seen since that probably doesn't happen automatically, it's another of those things that often fails to happen at all.

They close this comprehensive list of things that everyone knows would be good to do but many organizations are still not doing, by reminding about the need for and importance of backups. They recommend not only securely backing up all data to a different location than where the data is being used, but also maintaining and testing disaster recovery plans in concert with incident response planning, and reviewing these things at least annually or after significant changes.

We all know that if everyone were to be in full compliance with all of these guidelines, the result would be a clear increase in security. And we also know that few things change for the better, or change at all – since inertia is the norm. It's for this reason that the MVSP guidelines exist and it's the reason CISA has added their name to the group's growing list of contributors.

The bottom line is that doing all of these things would come at some cost and most businesses are looking for ways to cut costs rather than ways to incur additional expense. So it's going to be a difficult lift. But at least all of these useful concepts have been pulled together in a single place. And if world governments were to require compliance, well, then anyone who wanted to sell to those major markets would need to clean things up. And that would help everyone.