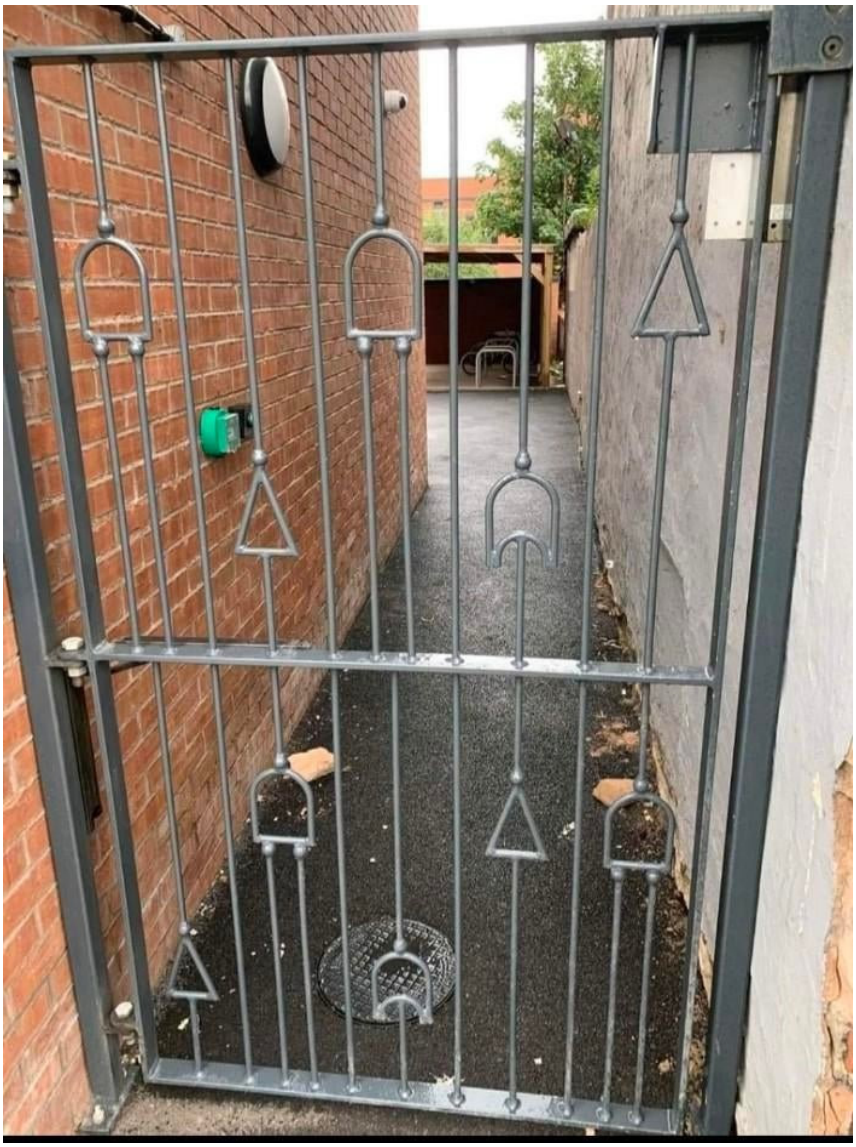
Security Now! #838 - 09-28-21

autodiscover.fiasco

This week on Security Now!

This week we examine a new pair of 0-days which have forced emergency updates to their respective products. We examine the growing annoyance of those who are reporting bugs to Apple, Epik's belated confirmation of their mega data breach, Windows 11's further progress toward its release, and its new and much more useful PC Health Check tool. We look at some additional fallout from this month's ever-exciting Patch Tuesday and take notice of a clever new approach for bypassing anti-malware checking under Windows. And after a quick check-in about the first two episodes of AppleTV's Foundation series, we settle in to examine the week's most explosive, worrisome and somewhat controversial disclosure of yet another huge Microsoft screw-up which caused this week's episode to be given the domain name: autodiscover.fiasco.

A Logic Gate



0-Day Watch

Chrome's 12th 0-day this year

Last week's Chrome emergency 0-day update left us at v93.0.4577.82. But that one didn't last long. Last Friday, Chrome was updated to v94.0.4606.61 with a fix for a single high priority update for yet another 0-day that Google says they are aware of being exploited in the wild.

Tracked as CVE-2021-37973, the vulnerability is a use after free in Chrome's new support for the Portals API — more on that in a moment. A researcher with Google's TAG team — their Threat Analysis Group — discovered and reported the flaw. And, as is usual and understandable in such cases, no one is releasing any additional information at this time and probably never, since who will care by the time no one cares? (If that sounds like a trick question, the correct answer is: No one will care.)

And for those keeping score at home, this brings the total year-to-date 0-day tally to 12.

Okay, so what's the "Portals" API?: https://web.dev/hands-on-portals/ "Portals" is a new webpage navigation system that enables the user's current webpage to show another page as an inset thumbnail, then perform a seamless transition to that next page by smoothly zooming the thumbnail to full size to replace its parent and becomes the new top-level document.

I have to say, it's kinda slick since it's the sort of effect that we're used to seeing on fancy OS platform UI's. Once it catches on, I'm sure we're going to be seeing a lot of this effect. (Probably more than we want, since it is a bit cutsie-poo.) Of course, once that happens, GRC is going to appear even more stone age. But that's okay. Once I get SpinRite caught up, probably after v7.2 where we'll have operation on BIOS and UEFI and native support for all drive technologies, I might go for a change of pace and spend some time on the website. But on the other hand... what we've seen with weird SSD read timings might be too much for me to resist exploring! :)

Next up on this week's 0-day Watch... is Apple.

Urgent Apple iOS and macOS updates have just been released to fix actively exploited 0-Days.

The day before Google pushed out that most recent high-priority update to Chrome, Apple released security updates to fix multiple security vulnerabilities appearing in older versions of iOS and macOS. Apple says (because Google told them) that they've been detected in exploits in the wild. Last Thursday's updates also expanded earlier patches for a previously resolved vulnerability that was being abused by the NSO Group's Pegasus surveillance tool, which is used in targeted attacks on iPhone users.

The most worrisome was a type confusion flaw that resides in the kernel component XNU. It was being exploited within a deliberately developed malicious application to execute arbitrary code with the highest privileges. iOS client applications are never allowed to have kernel root privileges. This flaw was also uncovered by Google's TAG team which said that it had detected the vulnerability being "used in conjunction with remote code execution targeting WebKit."

The patches are available for devices running macOS Catalina and iPhone 5s, iPhone 6, iPhone 6 Plus, iPad Air, iPad mini 2, iPad mini 3, and iPod touch (6th generation) running iOS 12.5.4.

Security News

Apple appears to be annoying their bug reporters.

Back on September 9th, the Washington Post ran a story titled: "Apple pays hackers six figures to find bugs in its software. Then it sits on their findings." The subtitle added: "Lack of communication, confusion about payments and long delays have security researchers fed up with Apple's bug bounty program." As I was assembling the podcast to discuss the mega Meris botnet, I read what the Washington post had to say. Basically, it was grumbling from researchers over how Apple's security team was leaving bug reports unsolved for months, shipping incomplete fixes, low-balling monetary rewards, or banning researchers from their program when they complained. That's all worrisome, but I decided that it was mostly generalizations that didn't have sufficient meat for the podcast. This week, however, we're not asking "where's the beef?"

Last Thursday, a Russian security researcher named Denis Tokarev who uses the handle "Illusion of Chaos", having finally given up waiting for Apple to acknowledge and repair three of the four vulnerabilities he had informed them of in April, published full details and proofs of concept (on Github) for the three vulnerabilities which Apple had not addressed — even with the recently released iOS 15. The three problems are:

A vulnerability in the Gamed daemon that can grant access to user data such as AppleID emails, names, auth token, and grant file system access.
(PoC: https://github.com/illusionofchaos/ios-gamed-0day)

Any app installed from the App Store may access the following data without any prompt from the user:

- Apple ID email and full name associated with it
- Apple ID authentication token which allows to access at least one of the endpoints on *.apple.com on behalf of the user
- Complete file system read access to the Core Duet database (contains a list of contacts from Mail, SMS, iMessage, 3rd-party messaging apps and metadata about all user's interaction with these contacts (including timestamps and statistics), also some attachments (like URLs and texts))
- Complete file system read access to the Speed Dial database and the Address Book database including contact pictures and other metadata like creation and modification dates (The researchers noted that he had just checked on iOS 15 and this last one is now inaccessible, so it must have been quietly fixed)

A vulnerability in the nehelper daemon that can be used from within an app to learn what other apps are installed on a device. (PoC: https://github.com/illusionofchaos/ios-nehelper-enum-apps-0day)

An additional vulnerability in the nehelper daemon can also be used from within an app to gain access to a device's WiFi information. (PoC: https://github.com/illusionofchaos/ios-nehelper-wifi-info-0day)

Denis also published his proof of concept code for a fourth issue which affects the iOS Analyticsd daemon. This was the fourth of the bugs he reported to Apple in April and was the only of his four issues patched in iOS 14.7 in July.

His blog posting Last Thursday was titled: "Disclosure of three 0-day iOS vulnerabilities and critique of Apple Security Bounty program" He begins... (https://habr.com/en/post/579714/)

I want to share my frustrating experience participating in the Apple Security Bounty program. I've reported four 0-day vulnerabilities this year between March 10 and May 4, as of now three of them are still present in the latest iOS version (15.0) and one was fixed in 14.7,

[I'll interrupt here to note that calling these 0-days is not correct. They are vulnerabilities. What makes 0-days special is that they are discovered being in-use in the wild. That really is, and should be, kept as a special case for vulnerabilities. If we don't require that then everything is a 0-day and it loses all significance, except to sound more scary.]

... but Apple decided to cover it up and not list it on the security content page. When I confronted them, they apologized, assured me it happened due to a processing issue and promised to list it on the security content page of the next update. There were three releases since then and they broke their promise each time.

Ten days ago I asked for an explanation and warned them that I would make my research public if I didn't receive an explanation. My request was ignored, so I'm doing what I said I would. My actions are in accordance with responsible disclosure guidelines (Google Project Zero discloses vulnerabilities in 90 days after reporting them to vendor, ZDI - in 120). I have waited much longer, up to half a year in one case.

I'm not the first person who is unhappy with Apple Security Bounty program. Here are some other reports and opinions:

[Then he lists eight publications and three tweets. And one of the publications is iMore.]

Then, yesterday he blogged an article with the title "How malware gets into the App Store and why Apple can't stop that" https://habr.com/en/post/580272/

Only after I had published a post detailing three iOS 0-day vulnerabilities and expressing my frustration with Apple Security Bounty Program, I received a reply from Apple:

> We saw your blog post regarding this issue and your other reports.
>
> We apologize for the delay in responding to you. We want to let you know that we are still investigating these issues and how we can address them to protect customers. Thank you again for taking the time to report these issues to us, we appreciate your assistance.
>
> Please let us know if you have any questions.

Indeed, I do have questions. The same ones that you have ignored. I'm gonna repeat them. Why was the fix for analyticsd vulnerability quietly included in iOS 14.7 update but not mentioned on its security content list? Why did you promise to include it in the next update's list but broke your words not once but three times? Why do you keep ignoring these questions?

Given that there's been a lot of coverage of this recently I wanted to give it some attention. I suspect we're dealing with the collision of egos and busy companies. Researchers doubtless work quite hard to find problems. And once found they feel possessive of them, want them to be acknowledged and to be fairly compensated for their work. And while the problems that Denis found and reported may not be remote code execution, information disclosure problems can be significant — especially with Apple increasingly begging us to trust them, allowing them to carry our purchasing cards and to acquire real time health data. And I'd also wager that the signal to noise ratio among all of the reports of problems that Apple receives probably makes wading through an endless stream of non-problems, looking for true problems, annoying and fatiguing.

But we are seeing that incredibly cash-rich companies like Microsoft and Apple do not appear to be budgeting the resources that they perhaps should.

Epik Confirms Hack, Gigabytes of Data on Offer

Last week we talked about the eMail I received when some GRC domain eMail accounts were obtained from the domain registrar and web host Epik. At the time, Epik was denying that anything had happened. Though it took them a week to acknowledge what all of the evidence showed, they finally did. So I just wanted to quickly follow-up.

Threatpost's updated coverage of this quoted the CTO and cofounder of Blue Hexagon. He explained: "This has happened to a lot of the right-wing outlets (Parler and Gab) because they have been brought up in record time to capitalize on current events like the election, vaccines, voting and deplatforming to be able to fundraise or get traction quickly. Unfortunately, this usually means that security takes a back seat [due to] business pressure, which can result in breaches. Usually, hacktivists are not known to be as sophisticated as nation-state groups or the big game ransomware operators, but nowadays a lot of tools and malware are for sale and can be used by anyone who is reasonably technically adept at penetrating networks."

And, of course, last week's topic was about Cobalt Strike which is precisely that sort of turnkey off-the-shelf hacking tool.

Microsoft gets Windows 11 ready for release with a new "Release" build

As the Windows 11 October 5th launch nears, I wanted to let those on the inside know that the Windows Insider 'Release' channel has started offering Win11.

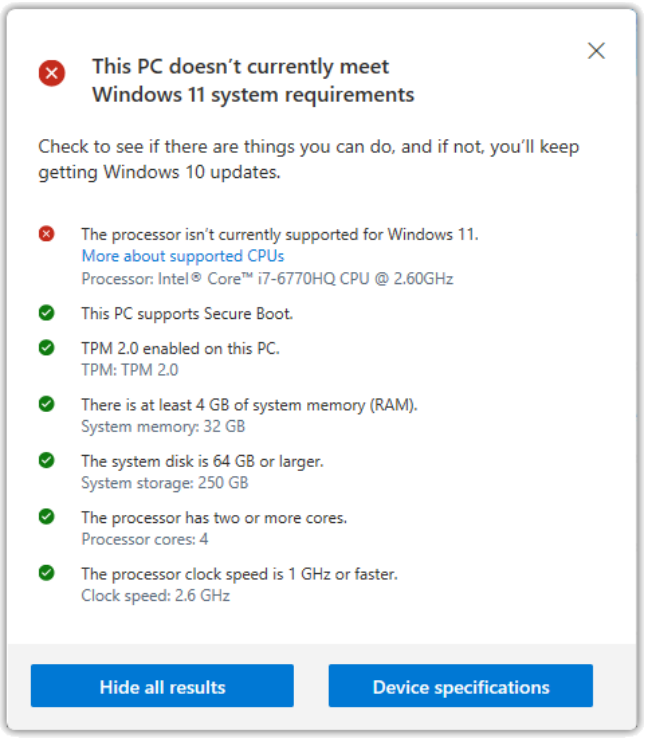
Before now, the Windows Insider Release channel was only offering users Windows 10 21H2 (v19044), which is expected to be released next month. But as of last Thursday, Microsoft is now offering Windows 11 as an optional download within Windows Update for users with compatible hardware. (As for compatible hardware, we'll be talking about that in a minute.) The Win11 build being offered in the Release channel is Build 22000.194, which is the release that became available to users in the Beta channel the week before, on September 16th.

Even though the last few Beta channel builds have been feature-stable and have only been fixing bugs, this still seems pretty quick to me given how flaky some of those earlier Win11 releases have been. I know that rounding off some pointy corners and changing the look and feel of the Start menu is no big deal. But this still has a half-baked feel to it. Let's hope I'm wrong.

On the downside is the fact that a bunch of popular longstanding Windows features have been removed from Windows 11 which has upset many users who are asking to have them restored. The currently-missing features include the Taskbar context menu, the ability to drag and drop files onto open Taskbar applications, the ability to move the taskbar to the top or sides of the screen, and the ability to ungroup running applications. It feels as though Microsoft is trying (and succeeding) in dumbing it down. Oh well, change is good, right?

But apparently, **I** won't be needing to worry about anything changing anytime soon, because:

Newly updated PC Health Check tool:
Microsoft has released an updated PC Health Check tool which enumerates the seven ways that the machine it's running on might succeed or fail to meet Windows 11's expectations:



https://aka.ms/GetPCHealthCheckApp **https://grc.sc/check**

To my surprise, my super-snappy Intel NUC containing a quad core Intel i7-6770HQ running at 2.6Ghz fails to make the grade. Since this machine has one of those super-wide curved Dell monitors which has an excess of horizontal space at the cost of vertical space, I run Win10 on this machine with the Taskbar over on the left edge of the monitor. So until that's fixed in Windows 11, I'll be content with Windows 11's rejection of my system. And anyway, I'd rather report on other people's pain than my own.

Windows 10 emergency update "might" resolve some Patch Tuesday troubles.
It turns out that the failed network printing troubles we covered last week were not the only problems caused by September's Patch Tuesday. Microsoft has stated that users may experience app freezes, app crashes, and the inability to launch an application. So, pretty much what an operating system is supposed to do, it might decide not to, after being made better with those updates.

Apparently, the trouble affects users who employ Microsoft's Exploit Protection Export Address Filtering (EAF) feature. It's used to detect dangerous operations used by malicious code or exploit modules. Microsoft said that *"After installing KB5005101 or a later update on devices using Microsoft Exploit Protection Export Address Filtering (EAF), you might have issues with some applications. You might be experiencing this issue if apps fail to open, fail to open files, or you might receive a white window when attempting to login."*

Microsoft also said *"This issue is resolved using Known Issue Rollback (KIR). Please note that it might take up to 24 hours for the resolution to propagate automatically to consumer devices and non-managed business devices. Restarting your Windows device might help the resolution apply to your device faster."*

The terms "might and maybe" are what you get once all of the actual "science" has been removed from the "computer science." But hopefully, by now, two week's after this month's exciting Patch Tuesday, all of the "might's and maybe's" will have had the chance to work themselves out, and things will be working again for everyone... just in time for next month's Patch Tuesday adventure!

This this Cert valid?
Under the category of "this is just too clever to believe"... Google's increasingly prolific TAG team — their Threat Analysis Group — has spotted and reported a diabolically clever new scheme being used by malware to avoid detection by 3rd party anti-malware tools running on Windows:

The attackers figured out a way to create a malformed code signing signature which would be seen and treated as valid by Windows, thus allowing the code to run, while being undecodable and thus unchecked by 3rd party anti-malware systems which use the OpenSSL codebase to perform their various crypto operations. That is just too clever.

This new technique was observed being exploited by a notorious family of unwanted software known as OpenSUpdater. It's used to download and install other suspicious programs on compromised systems. Most targets of the campaign are users located in the U.S. who are prone to downloading cracked versions of games and other grey-area software.

While adversaries have previously relied upon illegally obtained digital certs to sneak adware and other unwanted software past malware detection tools, or by embedding the attack code into digitally signed, trusted software components by poisoning the software supply chain, OpenSUpdater stands out for its intentional use of malformed signature to slip through defenses.

Sci-Fi

A shaky Foundation
Before I describe what I think, so far, about AppleTV's foundation series, I wanted to share two paragraphs from Mashable's context-setting posting:
https://mashable.com/article/foundation-book-vs-show

An adaptation of Isaac Asimov's classic science fiction novels, Foundation is less interested in following its source material to the letter than it is in creating a story within Asimov's universe that would make good TV. The basic plot remains the same: mathematician Hari Seldon (Jared Harris) foretells the fall of the Galactic Empire thanks to his theory of psychohistory. Knowing the fall is inevitable, he establishes the Foundation in order to preserve knowledge and, hopefully, civilization in the years to come.

Foundation takes this story and tweaks it in some pretty big ways, which makes sense when considering the scale of Asimov's work. The Foundation books are collections of interlocking stories and novellas whose events span hundreds of years, not to mention an entire galaxy. Characters who appear in one story may be long dead in the next, and so much happens in between stories that we never fully "see" on the page. These elements make creating a completely faithful TV show rather challenging, which explains several of showrunner David S. Goyer's choices to deviate from the books.

My own take is that, so far, it has definitely not been amazing. So, naturally, it feels like an expensive lost opportunity. Foundation has often been called the story that's impossible to bring to the screen, and so far I think we're seeing this play out. There are several problems that I've seen. I'll point out a few:

For one, I think the series is having a problem with pacing. It seems to swing between moving quickly and moving slowly.

One of my biggest complaints is that the dialog sound track is often muddy and unintelligible. The very first scene of the series has four young friends playing outdoors, bundled up against the cold. They're talking meaningfully, like it's important. Yet it's unintelligible. I paused, backed it up, turned up the volume... still "blub blub blub blub." It seems so inconsiderate of the audience, and I wonder how difficult it could be in this day and age not to have everyone appropriately mic'd up and articulating their lines clearly. (We could just agree that that's something everyone does in the far future.) And, unfortunately, this continued intermittently throughout the two hours, with major characters mumbling to each other, where we're clearly supposed to be listening and obtaining information. And I get it that there's a sense of it being more real and realistic if someone turns to someone standing next to them and mumbles... but if we're not supposed to hear what's said, then just give the other person a meaningful look!

And we were unable to watch both episodes back to back, because there's something heavy and oppressive about it. It's not the story, it's the feel. Both Lorrie and I felt somewhat drained and exhausted after each hour. Maybe it was just from straining to hear what's being said. But whatever it was, an hour of this was a lot.

And speaking of straining, what the hell happened at the end of the second episode? I mean... W. T. F. I had to go online the next day and read a bunch of speculation. It's not just me. Nobody knows for sure since, as has been observed, what we're being dragged through is not the story embodied in Asimov's Foundation trilogy, but rather the same overall concept set in a similar universe.

autodiscover.fiasco

For those who are not familiar with the term "fiasco"—it's a particular favorite of mine—is defined as: "a thing that is a complete failure, especially in a ludicrous or humiliating way."

A recurring problem in security occurs when attempts are made to make complex and secure things less complex. It's often the case that they also become less secure. A classic example was the creation of UPnP — universal plug n' play — which defined a means for essentially bypassing the crucially important firewall protection being created by NAT routers. Why? Because those pesky NAT routers were getting in the way (exactly as they were designed to and as any savvy user wanted them to). So Microsoft and Intel defined an entirely unauthenticated protocol by which any device on the LAN could map external traffic through, to bypass NAT protections. That was bad enough, since attackers figured out how to get our fancy web browsers inside the LAN to send UPnP requests to the LAN's NAT router on behalf of the attackers, thus allowing the attackers into the network. Couple that with the fact that mistakes are invariably made, and we saw many routers which mistakenly also bound their UPnP service to the WAN interface. I immediately added a test for that to ShieldsUP! That was 55,145 positive tests ago.

UPnP is not today's topic. But it serves as a prime example of the absolute sacrifice of security in the name of convenience. And that is the moral of today's topic: You take a bad and fundamentally flawed idea, mix in the inevitability of human error and even, if you can believe it, human hubris, and what you get is Microsoft, today, frantically contacting domain registrars across the world to preemptively register hundreds of domains in every TLD before the bad guys can get them.

So what's all this about?

The original idea was to make it much easier for users to configure their eMail clients without any need to bother with all of those pesky eMail setup details. You know, things like the server's full domain name, which port to connect to, and what sorts of authentication is supported and so on.

The idea is that the user would only need to provide what they know: their eMail address and password. Then the eMail client would use that eMail domain as a starting point and emit a series of HTTP and later HTTPS queries, searching for a server that would answer that query and thus provide the needed configuration information. The trouble was that domain names themselves are not well formed. Sometimes the user's eMail might be @example.com. And sometimes @mail.example.com. And perhaps @mail.example.co.uk. And in what Microsoft refers to as "configuration resiliency" (really, you can't this up) they encourage clients to try everything. To throw everything at the wall to see what might stick.

So, the crux of the problem is that one of the many things thrown at the wall is to emit queries to a subdomain of the eMail domain called "autodiscovery". So an HTTP query is sent to autodiscovery.mail.example.com. But again, because we're dealing with an ad hoc and weakly-defined specification where Microsoft's stated goal is maximum resiliency, Microsoft said that it's not necessary to have the autodiscovery subdomain be a subdomain of the user's eMail

— since that might not be convenient. So, instead of autodiscovery.mail.example.com, the autodiscovery service might also be at "autodiscovery.example.com". Sure. So try that, too. And you know... there are those tricky domains like example.co.uk where no one is really sure where the enterprise's domain name stops and the public domain begins. And, after all, we want to be resilient! So we wouldn't want to miss an available autodiscovery server because we didn't try hard enough or look everywhere. Therefore, since we're not sure how far back up the domain hierarchy we should go, we better keep going until we either find an autodiscovery server or we run out of domain name.

And believe it or not, that's what Outlook does. What could possibly go wrong? How about querying the domain "autodiscovery.com" with an HTTP or HTTPS query CONTAINING the user's eMail address and their unencrypted eMail password... where those credentials are the same as they use to authenticate to their company's entire network domain? It's unbelievable, but it's true.

https://www.guardicore.com/labs/autodiscovering-the-great-leak/

Guardicore's Amit Serper published the result of Guardicore's research titled: "Autodiscovering the Great Leak". Now that we have all of this background, Amit's Executive Summary will make sense:

- ***Autodiscover**, a protocol used by Microsoft Exchange for automatic configuration of clients such as Microsoft Outlook, has a design flaw that causes the protocol to "leak" web requests to Autodiscover domains **outside** of the user's domain but in the same TLD (for example, Autodiscover.com).*

- *Guardicore Labs acquired multiple Autodiscover domains with a TLD suffix and set them up to reach a web server that we control. Soon thereafter, we detected a massive leak of Windows domain credentials that reached our server.*

  - *Between April 16th, 2021 to August 25th, 2021 we have captured:*
    - *372,072 Windows domain credentials in total.*
    - *96,671 UNIQUE credentials that leaked from various applications such as Microsoft Outlook, mobile email clients and other applications interfacing with Microsoft's Exchange server.*

- *This is a severe security issue, since if an attacker can control such domains or has the ability to "sniff" traffic in the same network, they can capture domain credentials in plain text (HTTP basic authentication) that are being transferred over the wire. Moreover, if the attacker has DNS-poisoning capabilities on a large scale (such as a nation-state attacker), they could systematically syphon out leaking passwords through a large-scale DNS poisoning campaign based on these Autodiscover TLDs.*

- *Additionally, we have developed an attack – "The ol' switcheroo" – which downgrades a client's authentication scheme from a secure one (OAuth, NTLM) to HTTP Basic Authentication where credentials are sent in clear text.*

Just as as aside, so that I don't forget to mention it, the way this credential security downgrade attack works is to simply have the intercepting autodiscovery.whatever web server reply to the first OAuth or NTLM query with an "I don't support that fancy protocol" reply — an HTTP 401 response from the server, telling it to use HTTP Basic Authentication.

I'll just share the first three paragraphs of this much longer and more detailed report to reiterate the nature of this danger:

> *As a part of the ongoing security research efforts by the Guardicore Labs team, we have discovered an interesting case of credential leak affecting a large number of people and organizations worldwide.*
>
> *The credentials that are being leaked are valid Windows domain credentials used to authenticate to Microsoft Exchange servers. The source of the leaks is comprised of two issues:*
>
> - *The design of Microsoft's Autodiscover protocol (and the "back-off" algorithm, specifically).*
> - *Poor implementation of this protocol in some applications.*
>
> *As mentioned, Microsoft's Autodiscover protocol was meant to ease the configuration of Exchange clients such as Microsoft Outlook. The protocol's goal is to make an end-user be able to completely configure their Outlook client solely by providing their username and password and leave the rest of the configuration to Microsoft Exchange's Autodiscover protocol. It is important to understand that since Microsoft Exchange is part of the "Microsoft domain suite" of solutions, the credentials that are necessary to login to one's Exchange-based inbox are in most cases their domain credentials. The implications of a domain credential leak in such scale are massive, and can put organizations in peril. Especially in today's ransomware-attacks ravaged-world – the easiest way for an attacker to gain entry into an organization is to use legitimate and valid credentials.*

Skipping way down to some interesting and important details, Amit explains:

> *The protocol has several iterations, versions and modes – their full documentation can be found on Microsoft's website, however, in this article, we will discuss a specific implementation of Autodiscover based on the POX XML protocol.*
>
> *In order to truly understand how Autodiscover works, we need to know what happens "behind the scenes":*
>
> - *The client parses the email address supplied by the user – amit@example.com.*
> - *The client then tries to build an Autodiscover URL based on the email address with the following format:*
>   - *https://Autodiscover.example.com/Autodiscover/Autodiscover.xml*
>   - *http://Autodiscover.example.com/Autodiscover/Autodiscover.xml*
>   - *https://example.com/Autodiscover/Autodiscover.xml*
>   - *http://example.com/Autodiscover/Autodiscover.xml*
>
> ***In the case** that none of these URLs are responding, Autodiscover will start its "back-off" procedure. This "back-off" mechanism is the culprit of this leak because it is always trying to resolve the Autodiscover portion of the domain and it will always try to "fail up," so to speak.*

Meaning, the result of the next attempt to build an Autodiscover URL would be:

http://Autodiscover.com/Autodiscover/Autodiscover.xml

This means that whoever owns Autodiscover.com will receive all of the requests that cannot reach the original domain. For more information about how Autodiscover works, please check out Microsoft's documentation.

And then comes the proof of concept. Amit writes:

In order to see if the Autodiscover leak scenario is even a viable one, we have purchased the following domains:

- Autodiscover.com.br – Brazil
- Autodiscover.com.cn – China
- Autodiscover.com.co – Columbia
- Autodiscover.es – Spain
- Autodiscover.fr – France
- Autodiscover.in – India
- Autodiscover.it – Italy
- Autodiscover.sg – Singapore
- Autodiscover.uk – United Kingdom
- Autodiscover.xyz
- Autodiscover.online
- Autodiscover.cc
- Autodiscover.studio
- autodiscover.capital
- autodiscover.club
- autodiscover.company
- autodiscover.jp
- autodiscover.me
- autodiscover.mx
- autodiscover.ventures

Later, these domains were assigned to a webserver in our control and we were simply waiting for web requests for various Autodiscover endpoints to arrive. To our surprise, we started seeing significant amounts of requests to Autodiscover endpoints from various domains, IP addresses and clients. The most notable thing about these requests was that they requested the relative path of /Autodiscover/Autodiscover.xml **with the Authorization header already populated with credentials in HTTP basic authentication**.

Now, as you might imagine, Microsoft is a little bent out of shape by this surprise revelation. After Guardicore released their report, Microsoft issued the following huffy statement:

"We are actively investigating and will take appropriate steps to protect customers. We are committed to coordinated vulnerability disclosure, an industry standard collaborative approach that reduces unnecessary risk for customers before issues are made public. Unfortunately, this issue was not reported to us before the researcher's marketing team presented it to the media, so we learned of the claims today." — Jeff Jones, Sr. Director, Microsoft.

And, you know, that would be okay if the title of a presentation given on Friday, March 31st, 2017 at 3:30pm during BlackHat Asia 2017, hadn't been: "All Your Emails Belong to Us: Exploiting Vulnerable Email Clients via Domain Name Collision"

https://www.blackhat.com/docs/asia-17/materials/asia-17-Nesterov-All-Your-Emails-Belong-To-Us-Exploiting-Vulnerable-Email-Clients-Via-Domain-Name-Collision-wp.pdf

The Abstract of that paper explains:

> *The Autodiscover HTTP Service Protocol provides a way for Autodiscover clients to find Autodiscover servers. This protocol extends the Domain Name System (DNS) and directory services to make the location and settings of mail servers available to clients. In this paper, we take a closer look at the Autodiscover protocol and identify its threat model. We analyse Autodiscover client implementations in two mobile built-in email clients to discover flaws which allow remote attackers to collect user credentials through domain name collision. We discover how many clients have vulnerable implementations by collecting and analysing HTTP request information received by our servers, registered with specially crafted domain names. We make our analysis based upon data we collect from 25 different domains. Our dataset contains information on about 11,720,559 requests and we observe 9,726,028 requests containing authentication information. We identify 2473 different email clients which use vulnerable Autodiscover client implementation. Finally we propose different mitigation techniques for users, enterprises, and application developers to improve their email clients.*

In other words, Microsoft, this recent work by Guardicore is a **reminder** of a directly-related issue that was fully documented and disclosed four years ago... which has never been fixed. Apparently, because it wasn't directly aimed at you, and you were not given specific instructions about how to fix it, you just ignored the whole thing until now. Your entire Autodiscover concept has always been an insecure and bad idea, and nothing has changed during the intervening four years to either make it any better, or to resolve its fundamentally broken design. Now, I realize, Microsoft, that you've got important work to do. The world has been clamouring for Windows 11.

So what is Microsoft doing? As I mentioned at the top, Microsoft is frantically registering the "autodiscover" domain in every top level domain they can, just as fast as they can. Friday, BleepingComputer's Lawrence Abrams wrote: "At the time of this writing, BleepingComputer has confirmed that Microsoft registered at least 68 domains related to Autodiscover, which are listed below:

autodiscover.af	autodiscover.tl	autodiscover.pn
autodiscover.ax	autodiscover.gf	autodiscover.pr
autodiscover.as	autodiscover.tf	autodiscover.re
autodiscover.ag	autodiscover.gl	autodiscover.rw
autodiscover.am	autodiscover.gp	autodiscover.lc
autodiscover.ac	autodiscover.gt	autodiscover.pm
autodiscover.by	autodiscover.gy	autodiscover.st
autodiscover.bj	autodiscover.ht	autodiscover.sn
autodiscover.bi	autodiscover.hn	autodiscover.sc
autodiscover.cm	autodiscover.hk	autodiscover.sl
autodiscover.cl	autodiscover.je	autodiscover.sx
autodiscover.do	autodiscover.ke	autodiscover.sk

autodiscover.tl	autodiscover.ly	autodiscover.sb
autodiscover.gf	autodiscover.li	autodiscover.so
autodiscover.tf	autodiscover.mg	autodiscover.so
autodiscover.gl	autodiscover.mw	autodiscover.gs
autodiscover.af	autodiscover.mq	autodiscover.com.es
autodiscover.ax	autodiscover.yt	autodiscover.org.es
autodiscover.as	autodiscover.mn	autodiscover.ch
autodiscover.ag	autodiscover.ms	autodiscover.tj
autodiscover.am	autodiscover.ma	autodiscover.tg
autodiscover.ac	autodiscover.na	autodiscover.tt
autodiscover.by	autodiscover.nz	autodiscover.ug
autodiscover.bj	autodiscover.ni	autodiscover.vi
autodiscover.bi	autodiscover.ng	autodiscover.uz
autodiscover.cm	autodiscover.nf	autodiscover.vu
autodiscover.cl	autodiscover.pa	autodiscover.vn
autodiscover.do	autodiscover.pe	autodiscover.wf

If ever we needed a clear example of a kludge, here it is. What a mess! Lawrence concluded his update by adding:

BleepingComputer also knows of thirty-eight other domains registered since September 22nd whose owners are hidden behind privacy or WHOIS restrictions that were likely registered by Microsoft, researchers, or potentially threat actors.

The actual number of registered domains is likely far larger, as BleepingComputer has seen Microsoft register multiple autodiscover domains for the same TLD, such as autodiscover.com.es and autodiscover.org.es.

One domain, autodiscover.ch, has been registered since at least 2015 and uses microsoftonline.com as the DNS servers, but it is not clear who owns it.

While registering autodiscover.[tld] domains will block some of the leaks, Microsoft will need to issue fixes for the poor Autodiscover implementation in their Microsoft Outlook and Office 365 mail clients to resolve the issue further.

As other non-Microsoft applications also have faulty protocol implementations, Microsoft will also have to release guidance on how to properly create Autodiscover URLs so that credentials are not sent to untrustworthy domains.

For their coverage of this, ThreatPost reached out to Alicia Townsend, OneLogin's technology evangelist. Alicia told Threatpost that it seems "incredible" that a product would send a user's username and password to an untrusted endpoint. "The fact that this is happening with an incredibly popular Microsoft product such as Exchange is even more disheartening." She pointed out that it's not clear how long this design flaw has been around, given that the Exchange Autodiscover feature was introduced in Exchange 2007. But, regardless, it doesn't shine a good light on Microsoft. "Whether the oversight was on the part of early developers or was introduced by more recent developers, it is clear that **Security First** was not their primary objective."

Right. You can imagine that some large enterprise customer came to Microsoft and complained that Exchange's Autodiscovery was not working for them due to their domain's naming structure. So some genius there thought *"Ah! No problem! Let's just have Outlook try everything until it finds something that works!"* What could possibly go wrong?

And, you know, what if Amit **had** instead quietly and privately reminded Microsoft of something crucially important about the fundamental design of one of their protocols that they presumably already knew about? Would they have taken action? And if so, when? Late last year and earlier this year they waited until it was too late, until their Exchange Server was being violently attacked, to begin fixing it, and then that fixing went on through the spring. And as we noted last week, we're now at nine months since Microsoft was first informed about the more serious of the many PrintNightmare problems, which has still not been resolved today.

Microsoft's official response to Amit's public disclosure was to say: *"We are committed to coordinated vulnerability disclosure, an industry standard collaborative approach that reduces unnecessary risk for customers before issues are made public."* But at what point does this become a means of stifling researchers' voices, shaming them into keeping quiet while the software publisher spends their time rounding off the pointy corners of windows and changing the Start menu's alignment?

I have some personal experience with that, myself. As many of us recall, I tried mightily to explain to Microsoft that repackaging their NT-derived Windows 2000 operating system as WinXP, while leaving its unneeded raw socket API in place, would be a real mistake. It wasn't until their own WinXP operating used its unneeded raw socket API to blast them with a devastating DDoS attack — the so-called MSBLAST worm — that they finally understood what I had been trying to tell them all along and limited that API's ability to do harm in a subsequent XP service pack.

Given the evidence, we're seeing that Microsoft has become a company that only responds to force. They must be forced to fix the things that are broken. "Responsible disclosure" is just a courtesy. It's one that the industry might consider withdrawing if publishers do not honor their side of the implicit agreement to fix what's been responsibly disclosed.