Security Now! #789 - 10-20-20

Anatomy of a Ryuk Attack

This week on Security Now!

This week we examine the coming controversial changes to the WebExtension API. We look at the revelations and fallout from last week's Patch Tuesday, and at Zoom's latest announcement of this week's roll-out of end-to-end encryption. We make sure everyone knows about the latest horrific SonicWall vulnerability and Microsoft's pair of not-that-worrisome out-of-cycle patches. We share a bit of miscellany and closing-the-loop feedback. Then we examine an actual Ryuk Ransomware intrusion and attack... step-by-step.

Browser News

Edge to be updated with browser extensions "Manifest V3"

The proposed changes to the WebExtensions API, generically known as "Manifest V3" were first announced by Google two years ago, in October 2018, for its Chromium project. And as we'll recall, Google's stated plans did not go over well. When Google announced their planned changes they said that the main intent was to improve extension security, improve extension performance and give users greater control over what extensions did and with which sites they could interact.

But extension developers quickly pointed out that the "Manifest V3" updates contained changes that crippled the ability of ad blockers, antivirus, parental control enforcement, and various privacy-enhancing extensions to do their job as they had been. As a consequence, Google's announcement triggered a significant backlash from users, extension developers, and even other browser makers. Since, among other things, the extensions had the effect of limiting the power of ad blockers to block ads, the non-Google community was unhappy to see Google — an advertising-based company — moving to limit our ability to limit the ads our browsers subjected us to.

As I've often mentioned, when I sometimes use a browser lacking a competent ad blocker I'm shocked by the experience. It's horrific. I would choose a browser ENTIRELY based upon whether or not it allowed me to say 'no' to ads.

At the time, browsers including Opera, Brave, and Vivaldi quickly distanced themselves from Google's plans, announced their intentions to ignore the Manifest V3 updates and thus allow users to keep using ad blockers. And Mozilla, which had implemented the WebExtensions API inside Firefox for compatibility with the rest, also explicitly denounced Chrome's plans and said it would not be following Google's WebExtensions API update to the letter and would, instead, make its own changes to allow ad blockers to continue working as always.

I would argue that Google had its heart in the right place, but that they did, perhaps willfully, under appreciate the importance of allowing for dynamic extension-based page filtering.

The original Web Request API allowed developers to install complete and powerful in-line filters. A query filter would inspect and perhaps modify any browser queries on the way to remote web servers. And a reply filter would receive remote web server replies before the browser saw them. This would allow the extension to make extensive edits of the received page, among other things, blocking subsequent requests for secondary page assets, including ads.

Google's V3 re-engineered solution discards all of that in favor of what they called the "Declarative Net Request API." Google explained that it would prevent extensions from inspecting web requests made on a page while providing much of the same functionality.

Again, I'll say that I think Google's heart was in the right place because the pre-V3 filtering capability was awesomely powerful. Two years ago, at the time of the announcement, Simeon Vincent, the Developer Advocate for Chrome Extensions, said that 42% of all malicious extensions Google had detected that year, January 2018, were abusing that API for nefarious purposes. He said: "With Web Request, Chrome sends all the data in a network request to the listening extension - including any sensitive data contained in that request like personal photos or emails. Because all of the request data is exposed to the extension, it makes it very easy for a malicious developer to abuse that access to a user's credentials, accounts, or personal information.”

With Google's Declarative Net Request API, an extension pre-registers rules that the browser reads and applies to each web page before and after it's loaded. This hugely improves security since extensions never receive and see any page data, and the browser makes all the modifications on behalf of the extension only when one or more pre-declared "rules" are met. And, in addition to enhanced privacy and security, this allows Chrome's optimized paths to handle all of the web request filtering, rather than leaving this to an extension's possibly-slow JavaScript code.

These changes promised to create a number of problems. The first and most obvious was that this WOULD be restricting what extensions would be able to do. The developers of NoScript and uBlock Origin made it clear that the new API's declarative rule system would not provide the same level of control.

But the most glaring limitation was the total number of rules that the new engine could accommodate. Google planned to allow for a maximum of 30,000 rules... Which was quickly revealed to be far insufficient for ad blockers, which often have to filter web requests for hundreds of thousands of ad-related domains. During the debate which ensued the stated requirements ranged from 90,000 to 150,000 with some folks arguing that up to 500,000 ought to be allowed to ensure that ad blockers would not hit an artificially imposed limit. In the end, Google agreed to raise their planned 30,000 rule maximum to 150,000.

Which brings up to today...

Manifest V3 changes are being tested in Chrome's developer channels and much of the post-announcement grumbling has died down, although some ad blocker extension devs have given up on their products' ability to reliably block ads once these changes reach stable versions of Chrome.

The reason this is in the news, aside from it being useful and important browser-side technology, is that last Wednesday Microsoft said that the Manifest V3 changes would shortly be rolling out in Edge and that they would not be crippling ad blockers. Their posting was titled: “Manifest V3 changes are now available to test in Microsoft Edge.”

https://blogs.windows.com/msedgedev/2020/10/14/extension-manifest-chromium-edge/

> In continuation of our commitment to reduce fragmentation of the web for all developers, and to create better web compatibility for our customers, we plan to support the Declarative Net Request API and other changes proposed as part of Manifest V3.
>
> The decision to embrace Manifest V3 changes is based on our dedication to enhance privacy, security & performance for the benefit of our end users as well as to allow developers to extend & provide rich experiences in Microsoft Edge.

> These changes are available for testing in the Beta and Stable channels.
>
> If you are an extension developer, you may already be aware that the background service worker change and the introduction of the Declarative Net Request API would require you to update your extension. Please refer to the Migrating to Manifest V3 document for exact changes required to port your extensions from Manifest V2.
>
> We believe that these changes will not compromise the capabilities of your extension or reduce the potential that the extension ecosystem has. These changes should reduce the time taken to review each submission, and improve certification turnaround time, thus reducing the overall cost of developing and maintaining extensions.
>
> We recognize the value of content blocking extensions and appreciate the role they play in honoring user’s choice by blocking advertisements and enhancing privacy by blocking cookies and we want developers to continue to offer these capabilities.
>
> After an extensive review of the concerns raised by content blockers and the community, we believe that a majority of those concerns have been resolved or will be resolved before Web Request API is deprecated. If you continue to face issues, we encourage you to share your feedback, where our team can engage to understand and address your feedback.

My hope is that a workable compromise has been, or can be, reached. I love the power of a full filtering ad blocker, though it does come with serious security and privacy tradeoffs. Given what a curmudgeon Gorhill is, I would trust him and his work including uBlock Origin completely. But those who listen to this podcast know how to be cautious. We're a vanishingly small minority.

If V3 limits our use of good extensions, perhaps we’ll still be able to turn on the original V2 API on a per-browser-instance basis.

Security News

Last (Patch) Tuesday

Last Tuesday Microsoft issued fixes for 87 security vulnerabilities — so, yeah, it was a slow month... though only when measured against everything year-to-date. Those 87 fixes included a pair of critical remote code execution (RCE) flaws, one in the core Windows TCP/IP stack and another in Microsoft Outlook.

I'll come back to those in a moment. There are also 9 other Critical flaws, 75 ranked as Important, and one classified as Moderate. They collectively affect Windows, Office and Office Services and Web Apps, Visual Studio, Azure Functions, .NET Framework, Microsoft Dynamics, Open Source Software, Exchange Server, and the Windows Codecs Library. Although, fortunately, none of these flaws are known to be under active attack, six vulnerabilities are listed as publicly known at the time of release.

Now, hopefully, most of our listeners will have perked up, or perhaps started digging a new bunker at the mention of a core flaw in the Windows TCP/IP stack. Those are never good. And...

Did I mention that once this is weaponized from a BSOD into an active remote code execution attack, everyone who has looked at it is saying that it's 100% wormable?

Rapid7 & McAfee both have very good write-ups. Here's how McAfee summarized the situation:

> Today, Microsoft announced a critical vulnerability in the Windows IPv6 stack, which allows an attacker to send?maliciously crafted packets to potentially execute arbitrary code on a remote?system. The proof-of-concept shared with MAPP (Microsoft Active Protection Program) members is both extremely simple and perfectly reliable. It results in an immediate BSOD (Blue Screen of Death), but more so, indicates the likelihood of exploitation for those who can manage to bypass Windows 10 and Windows Server 2019 mitigations. The effects of an exploit that would grant remote code execution would be widespread and highly impactful, as this type of bug could be made wormable. For ease of reference, we nicknamed the vulnerability “Bad Neighbor” because it is located within an ICMPv6 Neighbor Discovery “Protocol”, using the Router Advertisement type.

And Rapid7 wrote:

> If you’re in the U.S. and were waiting for an “October surprise”, look no further than CVE-2020-16898 which is a remote code execution (RCE) vulnerability in the Windows TCP/IP stack, or what our own Tod Beardsley likes to call “exploiting poor implementations of core IETF RFCs”.
>
> The vulnerability arises when the TCP/IP stack does not properly handle ICMPv6 Router Advertisement packets. Successful exploitation requires sending specially-crafted ICMPv6 Router Advertisement packets to a remote Windows computer and could give an attacker the ability to execute code on the target server or client. CVE-2020-16898 carries a CVSSv3 base score of 9.8.
>
> Our talented crew of Rapid7 vulnerability researchers have a technichal analysis up on AttackerKB, and security firm McAfee has their own technical analysis of CVE-2020-16898. Their research and engineering teams note that the Microsoft-provided exploit is “both extremely simple and perfectly reliable, and results in an immediate [Blue Screen of Death] (BSoD)”.
>
> Before we go any further, we would like to strongly encourage you to patch this vulnerability if you are running Windows 10, Windows Server 2019, or Windows Server Core 1903, 1909, or 2004. You really don’t want to mess around when the word “wormable” is being used and so many eyes are on the non-BSOD prize of a fully-working RCE. If you cannot patch, consider disabling ICMPv6 Recursive DNS Server (RDNSS) as a workaround (which is, unfortunately, only available for Windows 1709 and above) via the PowerShell command:
>
> ```
> netsh int ipv6 set int *INTERFACENUMBER* rabaseddnsconfig=disable
> ```
>
> As noted above, there are many folks who have access to the known BSoD exploit and more who are currently burning through cases of Mountain Dew while working to replicate the BSoD then to weaponize the unpatched vulnerability.

> In the short term (and, possibly long term) you should be more wary of disruption and distraction campaigns using this weakness, especially since IPv6 is very likely running on your internal network (where Bad Neighbor attacks are really most likely to occur) without you being aware of it.
>
> You and your organization should really be prepared to have between 1-5 critical “patch now” events each month for the foreseeable future. That may seem disruptive, but the spate of critical bugs in core business and remote access technologies has become the new normal and the only way to handle it is to make it part of the plan.

I mostly appreciated their observation that *“the spate of critical bugs in core business and remote access technologies has become the new normal and the only way to handle it is to make it part of the plan.”*

If at first you don't succeed, Zoom Zoom again.
Last Wednesday, Zoom announced that THIS week their 30-evaluation of end-to-end encrypted video conferencing would begin.

https://blog.zoom.us/zoom-rolling-out-end-to-end-encryption-offering/

As we know, they've pretty much blown the implicit assumption of trust we might have been willing to confer on them due to their botched and incredibly mis-communicated response to their early problems, and their plans to sort of, maybe mostly, but-we-really-want-to encrypt. But, that said, we wouldn't be fair if we didn't acknowledge that what they announced last week does really — on its surface — sound exactly right. Here's what Zoom said:

> We’re excited to announce that starting next week, Zoom’s end-to-end encryption (E2EE) offering will be available as a technical preview, which means we’re proactively soliciting feedback from users for the first 30 days. Zoom users – free and paid – around the world can host up to 200 participants in an E2EE meeting on Zoom, providing increased privacy and security for your Zoom sessions.
>
> We announced in May our plans to build an end-to-end-encrypted meeting option into our platform, on top of Zoom’s already strong encryption and advanced security features. We’re pleased to roll out Phase 1 of 4 of our E2EE offering, which provides robust protections to help prevent the interception of decryption keys that could be used to monitor meeting content.
>
> To be clear, Zoom’s E2EE uses the same powerful GCM encryption you get now in a Zoom meeting. The only difference is where those encryption keys live.
>
> In typical meetings, Zoom’s cloud generates encryption keys and distributes them to meeting participants using Zoom apps as they join. With Zoom’s E2EE, the meeting’s host generates encryption keys and uses public key cryptography to distribute these keys to the other meeting participants. Zoom’s servers become oblivious relays and never see the encryption keys required to decrypt the meeting contents.

> Zoom's CEO, Eric S. Yuan said: "End-to-end encryption is another stride toward making Zoom the most secure communications platform in the world. This phase of our E2EE offering provides the same security as existing end-to-end-encrypted messaging platforms, but with the video quality and scale that has made Zoom the communications solution of choice for hundreds of millions of people and the world's largest enterprises."

Okay, now... that was all great until Eric opened his mouth. It "provides the same security as existing end-to-end-encrypted messaging platforms"... WHAT?!?! No. If you're taking key management **out** of the hands of Zoom's servers and moving it to the meeting host, who then generates keys and distributes individual keys to each participant, then that's a FAR more secure architecture than any centralized key distribution system. Really, Eric does **not** need to have a quote stuck into every announcement. **He's** the reason Zoom so badly botched their previous attempts to impress us with their security plans.

So, Zoom's E2EE will be available as a technical preview this week. To use it, customers must enable E2EE meetings at the account level and opt-in to E2EE on a per-meeting basis.
https://blog.zoom.us/zoom-rolling-out-end-to-end-encryption-offering/

The posting goes on with a Q&A, answering the questions:

- How does Zoom provide end-to-end encryption?
- How do I turn on E2EE?
- When would I use E2EE?
- Do I have access to all the features of a regular Zoom meeting?
- Do free Zoom users have access to end-to-end encryption? [Yes, by the way]
- How is this different from Zoom's enhanced GCM encryption?
- How do I verify that my meeting is using end-to-end-encryption?
- How will you continue to provide a safe and secure platform?
- What is the rest of the timeline for E2EE?

Your SonicWall Network Security Appliance (NSA) MUST be patched now!

While we're on the topic of super-critical remote code execution network vulnerabilities with criticality ratings above 9 out of 10... we need to make sure that any of our listeners who might be responsible for the operation of one or more of the nearly 800,000 vulnerable SonicWall NSA devices that are currently exposed to the Internet has recently patched their device(s).

The vulnerability was discovered by the Tripwire VERT security team and was given CVE-2020-5135. It impacts the SonicOS which is the operating system running on SonicWall Network Security Appliance (NSA) devices. As its name suggests, the SonicWall NSAs are used as firewalls and SSL VPN portals to filter, control, and allow employees to access internal and private networks.

The Tripwire researchers explained that SonicOS contains a bug in a component that handles custom protocols. The vulnerable component is exposed on the WAN (public internet) interface, meaning that any attacker can exploit it remotely. And, moreover, Tripwire teased that exploiting the bug is trivial even for unskilled attackers. In its simplest form, the bug can cause a denial of service and crash devices, but "a code execution exploit is likely feasible."

Tripwire reported the bug to the SonicWall team, which released patches last Monday. Tripwire, announced the vulnerability 2 days later but declined to provide any further details. They wrote: https://www.tripwire.com/state-of-security/vert/sonicwall-vpn-portal-critical-flaw-cve-2020-5135/

> Tripwire VERT has identified a stack-based buffer overflow in SonicWall Network Security Appliance (NSA). The flaw can be triggered by an unauthenticated HTTP request involving a custom protocol handler. The vulnerability exists within the HTTP/HTTPS service used for product management as well as SSL VPN remote access.
>
> An unskilled attacker can use this flaw to cause a persistent denial of service condition. Tripwire VERT has also confirmed the ability to divert execution flow through stack corruption indicating that a code execution exploit is likely feasible. This flaw exists pre-authentication and within a component (SSLVPN) which is typically exposed to the public Internet. As of the date of discovery, a Shodan search for the affected HTTP server banner indicated 795,357 hosts.

Yes. 795,357 initially vulnerable hosts. This won't be an instant attack, since some reverse engineering and skilled R&D will be needed. But now that our ransomware ecosystem contains highly motivated penetration hackers who know they'll be able to sell their intrusions to high-bidding ransomware attackers, this will be FAR too juicy to be passed up.

And we already know that only some portion of those SonicWall devices will be updated before the inevitable attacks commence. Some of the press coverage of this suggested that SonicWalls were expected to come under active exploitation once proof-of-concept code was made publicly available. But I doubt that. Now that ransomware has created a market for network access, why would a hacker post a proof-of-concept when they could sell this into the dark web?

Microsoft's two out-of-cycle patches

And as if we weren't already have enough excitement, last Friday Microsoft released patches for a pair of critical remote code execution vulnerabilities, one in the HEVC codec and another in Visual Studio Code.

The HEVC problem, tracked as CVE-2020-17022, would enable attackers who can craft and arrange the processing of a malicious image to execute code on an unpatched Win 10 OS. Although all versions of Win10 are impacted, not all Win10 systems will be vulnerable because the installation of the HEVC codec is optional and obtained through the Microsoft Store. And as we have noted before, since it came from the Microsoft Store, it will be the Microsoft Store that provides the update. The HEVC update is only available through the store and Server editions are not vulnerable since the codec is not supported on Windows Server.

The Visual Studio Code bug (CVE-2020-17023) is not a huge worry. It allows attackers to craft malicious package.json file which, when loaded in Visual Studio Code, can execute malicious code. So downloading and running a build of an open source package might get someone who hadn't yet patched in to trouble. And if the user had admin permissions, that attacker's code could execute with those privileges, allowing them full control over an infected host.

Package.json files are regularly used with JavaScript libraries and projects. JavaScript, and especially its server-side Node.js technology, are one of today's more popular technologies. So, Visual Studio Code users are advised to update the app as soon as possible to the latest version.

https://portal.msrc.microsoft.com/en-us/security-guidance/advisory/CVE-2020-17023

Miscellany

Windows 10 "God Mode"

Our listeners may have heard that Microsoft is attempting to simplify and eventually eliminate the traditional Windows Control Panel, replacing it with a "Settings" app and also spreading things around. I prefer to have all of my settings stuff in one place so this effort doesn't impress.

There's recently been some discussion of a so-called "God Mode" which collects all of the Control Panel's various applets — and more — into a single large collection. It's not a bad thing, but it's kind of a kludge. But, in digging around, I discovered an hugely improved free solution — which, happily, also runs on Windows 7 — known as "Extended GodMode."

It's freeware from Peter Panisz' WinTools.info site:
https://www.wintools.info/index.php/extended-godmode

> The original GodMode contains more than 200 items, depending on your configuration and operating system version. Extended GodMode complements these functions with the Admin Tools and Control Panel elements. It displays all setting options in a single interface and allows access to them grouped in several ways according to different criteria. Extended GodMode also includes a powerful search engine. Individual searches can be saved to create groups of settings. Extended GodMode supplements default GodMode with the following features:
>
> - Quick search by item name
> - Searches can be saved
> - Manage favourites
> - Display recently used items
> - Display of most used elements
> - Integration of Control Panel and Admin Tools elements (can be disabled)
> - Quick access to each setting item from the software system tray icon menu
>
> Extended GodMode supports 64-bit Windows 7/8/10 operating systems [32-bit beta version]. The software can be used free of charge, no install is required.

On my Win7 machine it lists 340 individual problem-solving applets. It's a HUGE WIN for the Windows power-user — like probably all of our Windows users. And while you're there — even if this Extended GodMode doesn't capture your fancy — absolutely and definitely check out Peter's 4 pages of free/donationware offerings. Some look very tasty and delicious. The URL is "wintools.info" I think you'll thank me for what you find there.

This gadget is so nice that it inspired me to create a new permanent thread in my blog forum titled “My Favorite Utilities & Services.” I’ve added “Sync.com” and “SyncThing”... and over time I’ll be accumulating a more complete list. I noted that our “The Joy of Sync” was just a little over one year ago, October 1st, 2019... and I remain utterly delighted with both the “sync.com” cloud service for client-side encrypted TNO inter-Windows, Mac, iOS and Android folder synching, and the funky but powerful Swiss Army Knife open source peer-to-peer “SyncThing” solution. And I did hear that Sync.com is working toward a solution for Linux. I’ll be sure to mention when that happens.

Closing The Loop

Patrick / @pxltechnica
Hi Steve. Question about Shields Up and port 0. I recently moved and when scanning with SU port 0 is **closed** instead of **stealthed**. I'm using the same ISP and router as before, though I do think the modem is different. Just curious if there is a way to stealth port 0 because based on the few postings I can find this is an ISP issue and I can't do much about it. /Thanks.

Back when I was implementing the full service port scan I encountered some references to port 0 being a bit “ping like” in that, unlike the other 1023 service ports it wasn’t assigned to any specific service. And at the UNIX network API level, a specification of a “null port” when obtaining a network socket is shorthand for asking the OS to pick an unused local port for a connection. But at the network level, those 16-bits of port number **can**, indeed, be all zeroes. So even though it’s unused, it’s legal... so I figured, “what the heck, let’s send out a few TCP SYN packets with all their port bits set to zero and see what happens.” And as Patrick’s question notes, sometimes you get back a reply... which probably means that it **is** a means for an attacker to possibly unstealth someone who had overlooked that possibility... which is why Shields-UP! Sends out those packets.

As for what to do about it? That depends entirely upon where those SYN packets are being bounced back from. Since port 0 is not technically valid, any jump along the way would not really be faulted for thinking “What the heck? Port 0? Let’s just say no!” — Of course, “saying no” is not stealth. But my point is, those port 0 SYN packets might not even be making it all the way to your router. Your ISP certainly has no problem with blocking port 25, ports 137 through 139 and 445, so they might also be returned a “closed” status for port 0. Or perhaps that different cable modem is the culprit. In which case there’s really nothing that can be done.

Anatomy of a Ryuk Attack

The DFIR Report site specializes in forensic analysis of Ransomware attacks. Two days ago, the DFIR Report site posted a fascinating step-by-step forensic walk through of a five-hour Ryuk attack. That's a **total** of 5 hours (300 minutes) from the time that a phishing link was first clicked-on to provide a callback to the malware operators who were initially completely unaware of the resources they had just been provided access to, until, 300 minutes later, ALL of this enterprise's backup servers, non-backup servers, and domain controllers were fully encrypted by Ryuk.

https://thedfirreport.com/2020/10/18/ryuk-in-5-hours/

In order for The DFIR Report's the blow-by-blow walkthrough to make sense, we first need to add some details of three commonly-used malware packages to our knowledge base.

The first malware package is known as "BazarLoader." It's the tip of the spear; the first thing that gets loaded and runs to provide the foothold that an attacker needs for further penetration. CyberReason.com has been watching BazarLoader since its first sighting this past April. Here's a quick summary of what they know about it. And note that this is research about BazarLoader in general and not specifically about its use in this particular malware attack:

- Bazar can be used to deploy additional malware, ransomware, and ultimately steal sensitive data from organizations.
- Bazar malware infections are specifically targeting at professional services, healthcare, manufacturing, IT, logistics and travel companies across the US and Europe.
- Bazar leverages the Twilio SendGrid email platform and [validly] signed loader files to evade traditional security software in conjunction with a fileless backdoor to establish persistence.
- Over the course of their investigation, it is evident that Bazar is under active development. Recently, the active campaigns disappeared, to later reappear with a new version, suggesting that the group is under a development cycle.
- This stealthy loader evades detection by abusing the trust of certificate authorities, much like previous Trickbot loaders. This loader uses EmerDNS (.bazar) domains for command and control and is heavily obfuscated.

> As an aside, what is "EmerDNS"?? It's a completely decentralized blockchain-based DNS alternative that we've never talked about. What's blockchain-based DNS? Get a load of this:
>
> Quoting from the EmerDNS site: "Are you afraid your website could be suspended by authorities? With "the screws tightening" around the world, your fears might well be justified. EmerDNS is safe from any kind of censorship. No other user can modify your record — only the record creator can manipulate its content.

> Uh huh. What could be better for malware Command & Control servers than to rely upon a decentralized blockchain-based DNS that cannot be sinkholed by authorities?

- Anyway… Bazar also uses anti-analysis techniques to thwart automated and manual analysis, and loads the encrypted backdoor solely in RAM.

Okay, so we have a loader obfuscated and signed by valid certificates to evade detection and blocking. And that loader locates and connects back to its command & control servers using an authority-thwarting blockchain-based DNS system.

Next up we need to introduce "Cobalt Strike."
What's interesting is that Cobalt Strike is not hiding, since it lives at CobaltStrike.com, where it is being offered for sale for supposedly-legitimate "Red Team" penetration testing puposes: https://www.cobaltstrike.com/ The Cobalt Strike site says:

> Cobalt Strike is software for Adversary Simulations and Red Team Operations.
>
> Adversary Simulations and Red Team Operations are security assessments that replicate the tactics and techniques of an advanced adversary in a network. While penetration tests focus on unpatched vulnerabilities and misconfigurations, these assessments benefit security operations and incident response.
>
> Cobalt Strike gives you a post-exploitation agent and covert channels to emulate a quiet long-term embedded actor in your customer's network. Malleable C2 lets you change your network indicators to look like different malware each time. These tools complement Cobalt Strike's solid social engineering process, its robust collaboration capability, and unique reports designed to aid blue team training. To learn more about Cobalt Strike, watch the "Red Team Operations with Cobalt Strike" course.
>
> And how much does Cobalt Strike cost? New Cobalt Strike licenses cost $3,500 per user for a one year license. License renewals cost $2,500 per user, per year. Request a quote to begin the purchase process.

It's worth noting that despite the hefty price tag, Cobalt Strike has a trial version that's entirely useful and that it was an unlicensed trial version that was successfully used in the Ryuk attack. Malpedia describes Cobalt Strike a little more practically. They say:

> Cobalt Strike is a paid penetration testing product that allows an attacker to deploy an agent named '**Beacon**' on the victim machine. **Beacon** includes a wealth of functionality to the attacker, including, but not limited to command execution, key logging, file transfer, SOCKS proxying, privilege escalation, mimikatz, port scanning and lateral movement.
>
> Beacon is in-memory/file-less, in that it consists of stageless or multi-stage shellcode that once loaded by exploiting a vulnerability or executing a shellcode loader, will reflectively load itself into the memory of a process without touching the disk. It supports C2 and staging over

> HTTP, HTTPS, DNS, SMB named pipes as well as forward and reverse TCP; Beacons can be daisy-chained. Cobalt Strike comes with a toolkit for developing shellcode loaders, called Artifact Kit.
>
> The Beacon implant has become popular amongst targeted attackers and criminal users as it is well written, stable, and highly customizable.

And the third and final tool we need to note, for reasons that will soon become clear, is a 100% legitimate bit of freeware known as "AdFind" — as in "Active Directory Finder."

AdFind is offered by a freeware developer located at www.joeware.net. And I must say that I was somewhat delighted to encounter a website that's still on the net, and being actively maintained, which makes www.GRC.com look quite futuristic! **https://www.joeware.net/**

Once you click the "Enter" link on the homepage containing an odd photo, you get to his secondary landing page which explains: "Welcome to www.joeware.net, this website is dedicated to delivering powerful windows utilities for free so administrators can more effectively manage their environments. Please feel free to browse and download what you like."

And when you then browse through Joe's offerings you'll quickly note that Joe specializes in Active Directory oriented freeware. And that the target of our interest — and the interest of the Ryuk attackers — AdFind, describes itself as a: "Command line Active Directory query tool. Mixture of ldapsearch, search.vbs, ldp, dsquery, and dsget tools with a ton of other cool features thrown in for good measure. This tool preceded dsquery/dsget/etc by years though I did adopt some of the useful stuff from those tools." And the current version of this tool was built on January 11th, written in C++ and compiled with Visual Studio 2019.

An example of the malicious and powerful use of AdFind is this:

```
adfind.exe -f "(objectcategory=person)"
adfind.exe -f "objectcategory=computer"
adfind.exe -f "(objectcategory=organizationalUnit)"
adfind.exe -sc trustdmp
adfind.exe -subnets -f (objectCategory=subnet)
adfind.exe -f "(objectcategory=group)"
adfind.exe -gcb -sc trustdmp
```

- objectcategory=person – Finds all person objects
- objectcategory=computer – Finds all computers in domain
- objectcategory=organizationalUnit
- trustdmp – Dumps trust objects.
- objectcategory=subnet – Finds all subnets

In other words, AdFind immediately and cleanly dumps everything of use, known by an Active Directory server, about its local enterprise network. All the people, all the computers, all the subnets and everything else.

Okay. So we have the Bazar Loader, the Cobalt Strike Red team pen-testing toolkit with Beacon, and the freeware AdList utility...

The attack began with Bazar being introduced to the victim's environment through a phishing e-mail with a link to what looked like a PDF located in Google Docs — Thus trading on the target's knowledge and presumed trust of Google stuff. However, the link was to a file containing a double extension EmploymentRecord.PDF.exe. Naturally, since well-known extensions are suppressed, the user saw EmploymentRecord.PDF and didn't stop to wonder why THAT filename extension was not suppressed. In fact, it confirmed that the file was a "safe" PDF.

Thus, the "BazarLoader" achieved its first penetration and foothold. It immediately spawned a new Windows explorer process, into which it injected itself for continued stealthful execution and then terminated its spoofed PDF executable. After reestablishing itself in RAM, injected into a fresh explorer process, it reached out to its designated command and control server at 3.137.182.114, establishing a TLS tunnel to port 443.

Since you never know when a phishing attack is going to be successful, the attackers aren't always paying attention. In this instance, two and a half hours passed before they responded to the Bazar Loader's incoming call... whereupon its human controllers took over. However, since the initial penetration occurred at 5:01pm local time, it's also very possible that the attackers were waiting impatiently for two and a half hours to pass so that employees had left for the day and the attacker's subsequent actions would have a greater opportunity to go unnoticed.

In this instance the infected user was simply a Domain User with no privileges. But BazarLoader used the built-in Windows utility "Nltest." NL as in "NetLogon." And among the things that nltest will do, is obtain a list of the network's domain controllers to provide an initial mapping of the domain.

In this new world where the ZeroLogon vulnerability is now a reality the attackers wasted no time. They where in a machine with no privilege. So they immediately reached out and used ZeroLogon to reset the password of the network's primary domain controller, which nltest had identified for them.

They then began moving laterally from machine to machine using SMB file transfers to duplicate themselves and WMI to remotely execute the newly copied instance on the other machines. At onto every useful machine they dropped and executed a "Cobalt Strike" Beacon. Lateral movement was initiated via SMB file transfers and WMI executions of Cobalt Strike Beacons using both PE (portable executable) and DLL versions of Beacon.

The attacker again moved laterally into the secondary domain controller to establish a backup beachhead. That instance reached out and connected to a command and control server located at 88.119.171.94, again over TLS port 443. And a second Beacon DLL was dropped and executed via "RunDll32." It reached out to 5.2.64.174 over TLS 443.

At this point the attacker performed additional domain discovery using Net commands and the PowerShell Active Directory module. They then employed the default named pipe privilege escalation module on the server and used RDP to connect from the secondary domain controller back to the first domain controller using the built in Administrator account. It's believed that this was done due to something going wrong with their initial intrusion into the primary domain controller. So after moving laterally to the secondary domain controller they return to the primary over RDP which was guaranteed to work robustly.

Once on the main domain controller, another Cobalt Strike beacon DLL was dropped and executed. It reached out and connected to 5.2.64.174 over TLS 443 — the same C&C server used by the first Beacon.

Now with Bazar Loader and Beacons loaded and running in RAM on both the Primary and Secondary domain controllers, Joe's well-meaning “AdFind” utility was used to perform deeper domain reconnaissance to list all people and their machines. Once this was completed, after 90 minutes of careful work, moving throughout the network, the attackers were ready to launch their multi-tiered attack.

They first RDP'd into the network's Backup server which had been located during their earlier reconnaissance. A common-sense tactic is to always first encrypt an organization's backup servers to prevent easy recovery of all of the other machines. So Backup servers were targeted for Ryuk encryption first, followed by the organizations non-backup servers, and then all of the enterprise's workstations.

After two and a half hours of active work, and Ryuk now deployed and running on every machine except the primary domain controller — which was hosting the attacker, the attackers concluded their work by also executing Ryuk on the primary domain controller, thus encrypting every machine within the organization. After Ryuk finishes it displays the ransom notice and wipes itself from RAM to remove all remnants and record of the attack.

So, 5 hours following the first innocent click on a phishing link, which downloaded from Google and ran a validly-signed executable masquerading as a PDF... every one of that organization’s workstations, servers, domain controllers and backups were completely encrypted.