Security Now! #821 - 06-01-21
Epsilon Red

This week on Security Now!

This week we begin by examining the recent advances made by the just-released Chrome 91 and revisit Google's configurable long-term activity logging. On the ransomware front we look at yet another likely addition to the ransomware ecosystem: trusted 3rd-party file decryptors. We anticipate next week's activation of the Amazon Sidewalk ultra-wide area network, look at the questionable claims of another massive cyberattack, and at WhatsApp's privacy struggles with India and Brazil — couldn't happen to nicer folks. Then we'll touch on just a single bit of trivia before plowing into a detailed examination of the operation of the newest ransomware in town: Epsilon Red.

Web Browser News

Chrome advances to 91

I needed to trigger the update by going to About/Chrome, despite 91's release last Tuesday and my using Chrome every day since. So, presumably the migration to 91 will be gradual and casual. So as of last Tuesday, all of the channels are advanced by one with the Stable desktop channel moving to 91, the Beta channel moving to 92 and the Canary channel getting 93.

https://chromereleases.googleblog.com/2021/05/stable-channel-update-for-desktop_25.html

The release announcement shows that Chrome 91 for the desktop fixes 32 security vulnerabilities, eight of them designated high severity. And of those 32 vulnerabilities fixed, 21 were reported by external security researchers. And there's money to be made doing this. The researcher who reported a heap buffer in Chrome's Autofill earned themselves $20,000 dollars. This was one of the eight high severity problems. But even a report that Chrome rated as low severity, an out of bounds read in the V8 JavaScript engine netted its discoverer $15K. And there were a bunch of $3K, $5K and $7500 awards. So it might be possible to pay some bills while helping to make Chrome a bit safer for everyone.

Chrome 91 supports for the first time the use of the clipboard for pasting content into the browser, or example for webmail. Until now, data transfer could be done using drag and drop. But it was not possible to paste into the web browser from the clipboard.

And in a continued expansion of protection against NAT slipstreaming, Chrome 91 adds the block for port 10080 which we previously discussed as coming soon. Firefox has been blocking this port since November of last year.

But there's also more good news for Chrome and for all Chromium browsers. As a result of a new JavaScript compiler and the use of the new way of optimizing code location in memory, Google is reporting that Chrome 91 will execute JavaScript code 23% faster. Chrome's Product Manager Thomas Nattestad said:

"In 91, Chrome is now up to 23% faster with the launch of a new "Sparkplug" compiler and "short builtin calls". These save over 17 years of our users' CPU time each day. Sparkplug, is a new JavaScript compiler that fills the gap between needing to start executing quickly and optimizing the code for maximum performance. Short builtin calls optimize where in memory we put generated code to avoid indirect jumps when calling functions."

I dug into both of those new feature systems to see whether there was anything I could usefully report back to our listeners. But they both involve such deep compiler voodoo and machine-level architecture considerations that after coming back up for air the only thing I can meaningfully summarize is to say "Holy Crap! — I'm sure glad that these guys are on our side. And tht we get the benefit of apparently endless truly amazing technology for free."

Since I couldn't just leave it at that for anyone who IS interested in how it could be possible to still find another 23% to squeeze out of any already squeezed and re-squeeezed system, I've included links to the deeply technical articles I found in the V8.dev blog, which document and describe exactly what "Sparkplug" and "short builtin calls" accomplish, and how:

https://v8.dev/blog/sparkplug   https://v8.dev/blog/short-builtin-calls

We've talked previously about Chrome's decision to begin enhancing the URL field with commands. It's possible to type "delete history", "wipe cookies", "edit credit card", "update card info", "launch incognito mode" or just type "incognito". You can also enter "edit passwords", "update credentials", "update browser" or "update Google chrome". I could have used either of those last too this before writing this to update Chrome... If only I'd remembered. Perhaps next time. But that's also the trouble with this approach. The command line is arguably a far more powerful way of getting things done. But it assumes a certain never of commitment, memory, interest and expertise. Meanwhile the world has moved to multi-level nested discoverable menu systems. So unless you are "clearing cache" or "wiping cookies" often — and perhaps you are — it seems unlikely that you're going to hold seldom used browser URL bar commands in your head.

But in any event, the reason I bring this up today is that the Chrome folks apparently think they're really onto something with URL commands. So back at the beta level, Chrome is adding:

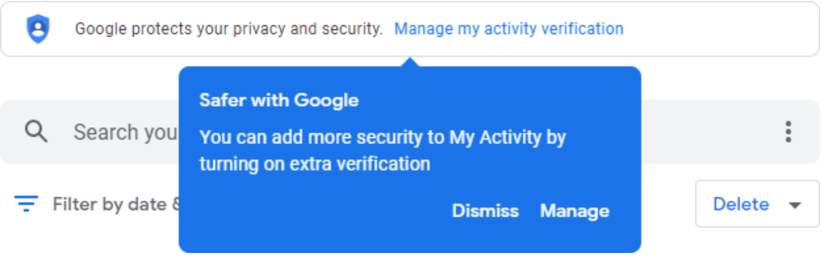
- “Run Chrome safety check”
  This Chrome Action will make it simple for you to run a Chrome safety check, directly from the address bar. The safety check can help keep you safe from data breaches, bad extensions, and more. To use it, type “Chrome safety check” or “Run password checkup”

- “Create doc”
  This Chrome Action will make it simple to create a new Google document, directly from the address bar. This saves time, so if you want to take notes or start an essay fast, you can do it quickly right from the address bar. To use it, type “New Google doc” or “Create Google doc”

- “Manage Google Account”
  This Chrome Action makes it extra simple to control your Google Account, including personal info, payments and subscriptions, and more. To use it, type “Control my Google Account” or “Adjust my Google Account”

And finally, though more a new Google account feature than a Chrome feature, I thought that our privacy conscious listeners might like to know that Google accounts “My Activity” monitoring page and report can now be password protected for added privacy:

https://myactivity.google.com/myactivity



Google protects your privacy and security. Manage my activity verification

Safer with Google

You can add more security to My Activity by turning on extra verification

To test this out, I went to the "My Activity" page using the link in the show notes. And, sure enough, Google immediately and helpfully offered to have me lock down that display so that any “Lookie Loos” who might be poking around would not have access to everything I've been doing recently. If you opt to turn this on, you'll need to verify that it's really you by entering your Google account password. And henceforth, anytime you wish to view your recent activity you'll again be required to re-prove that it’s really you.

I don’t think that I’ve ever looked at my activity, though I know we’ve talked about it before. While we’re again on the topic, there are two other options to consider. You can opt to have Google auto-delete your activity once it ages page 3 months, 18 months or 36 months. Or, against the threat of receiving a less personalized web experience, you can instruct Google not to save any data at all. The last time I visited that page I apparently set it to auto-delete after 18 months. But in revisiting it just now for the podcast I decided that I could live with a less personalized web experience. So I’ve instructed Google to delete everything and to no longer save anything. I wasn’t really worried about anyone seeing what I had previously been searching for. But now that history is no longer being kept.

Ransomware

As our main topic today, we'll be taking a close look at “Epsilon Red,” a curious newly discovered breed of ransomware. But something else is emerging that’s worth examining. The other ransomware news of note, despite ongoing attacks here and there, was the news that New Zealand’s based Emsisoft, has created their own ransomware decryption tool which, when it's provided with the master key from the attackers, is able to safely, reliably, and far more quickly decrypt a victim's machines than the decryptors provided by the ransomware attackers themselves.

Emsisoft's tool is aware of the encryption employed by 50 different breeds of ransomware, and it has the added benefit of not having been written by the same people who attacked the victim in the first place. Rather, it was written by a reputable security firm. This means that it's not necessary to take the added time to have an attacker-provided tool reverse engineered, analyzed and validated to verify that it won’t make a bad situation worse.

Growing experience with attacker-provided decryptors shows that they are not all reliable, some inadvertently mangle larger files, and most decrypt far less quickly than they encrypt. I have no explanation for that, since bulk data encryption will be performed by a symmetric cipher which should be symmetric not only in its keying but in its performance.

We do know that the bad guys will be motivated to get as much encryption done as quickly as possible, since any discovery of their operation will result in plugs being pulled out of walls with all possible alacrity. So it behooves the attackers to optimize their **en**cryption speed wherever possible. But as for **de**cryption? It’s easy to imagine that there’s no big rush there. They will have obtained their ransom payment, they will have provided a functioning unlock master key, and they will feel that they have met their obligation. But their victim will still be offline and out of business until all of their machines have been decrypted... which, all reports indicate can take quite some time.

But in both cases of the two high-profile attacks we've recently discussed — the first against Colonial Pipeline and the second against HSE, Ireland's national public healthcare system — their respective decryptors, the first for the DarkSide ransomware and the second for the Conti ransomware — were too slow to be of use. Colonial Pipeline, after paying $4.4 million in ransom, wound up restoring the bulk of their files from their backups. It wasn't clear whether they might have selectively decrypted some individual files that had been critically changed since those backups were made. And HSE used Emsisoft's decrypter which ran at twice the speed of the decryptor provided by the Conti gang while being, naturally, far more trustworthy.

So we may be seeing another component added to the ransomware ecosystem, with faster and far more trustworthy file decrypters being sourced by trusted and well known security firms. For properly designed encryption, the master decryption key will still be required. But it certainly makes sense for the ransomware attackers to even publish their file encryption formats — which, thanks to the miracle of public key crypto — in no way weakens their encryption. Doing so would serve to further mollify the victims and provide additional assurance of reliable file recovery.

Security News

Stepping off the Sidewalk

Security Now! episode #796, recorded December 8th, 2020, was titled “Amazon Sidewalk.”

The podcast presented our typical deep technical dive into the detailed design and operation of Sidewalk. And independent of the creepy feeling that some people get from the idea of enabling bidirectional sharing of heavily-encrypted low-bandwidth Bluetooth and 900 MHz LoRa radio over the participating networks of those in close proximity, we concluded at the time that from a technology standpoint, Amazon appears to have done everything right.

Sidewalk is back in the news for us this week because Amazon has announced that one week from today, next Tuesday on June 8th, the Sidewalk system will go live. And, as we discussed and expected at the time, it will be enabled by default for all those using compatible devices. So unless individual users preempt its auto opt-in, their Echo speakers, Ring Video Doorbells, Ring Floodlight Cams, and Ring Spotlight Cams will be participating in this communal signalling network.

Predictably, the click-seeking tech press is jumping up and down in a froth over neighbors stealing our WiFi — which has nothing whatsoever to do with Sidewalk, since it doesn’t have anything to do with sharing Wifi. As we concluded from our careful analysis late last year, the system's total bandwidth usage is extremely low, only being useful for signaling-class applications. And it is quite thoroughly encrypted. Amazon's intention here is clear: They want the system to be adopted so they’ve designed themselves out of it. The upside of enabling a low-power roaming Bluetooth or LoRa device to access an unknown Amazon user's network over a triple-layered deeply encrypted tunnel has all sorts of compelling use cases.

Leo, just last week you and I were talking about how once upon a time many, if not most of us, were deliberately running with open unencrypted WiFi networks because we wanted to share our Internet connectivity with our neighbors. It was considered neighborly. Today, of course, we no longer do that — we know better. But allowing Amazon's Sidewalk to remain enabled is not the same at all. The design is clearly intended to absolutely prevent any possible abuse of the system. A roaming wireless device that reaches out and connects to SideWalk has ZERO access to the hosting network it's connecting though, just as the hosting network has ZERO access to the roaming WiFi device's data. It was very well designed and we know how to do this now.

You could think of it a little bit like a VPN tunnel. And recall from our previous discussion that not even Amazon has access to the Sidewalk data. They are completely blinded to it. So the hysterical press which talks about yet another intrusion into our privacy has absolutely no idea what they're talking about.

But, all that said, I fully understand that some of our listeners might be thinking “no f'ing way am I letting Amazon do this.” In that case it’s absolutely possible to opt your devices out of participation in the Sidewalk communal network.

Using the Alexa app: Open More > select Settings > Account Settings > Amazon Sidewalk, where you'll find an on/off toggle.

And with the Ring app tap the “three-lined” menu > Control Center > Sidewalk to find the enable/disable control.

I have little doubt that “the tyranny of the default” will be alive and well, and that one week from today an interesting new and probably very useful long range low bandwidth wireless network will spring into existence. It's going to be interesting to see how it develops.

Just another phishing attack

I suppose that after the high profile Colonial Pipeline and HSE/Ireland attacks the press is a bit keyed up for any news of cyber-shenanigans. Consequently, when Microsoft announced last Tuesday that another attack had occurred, although there was really nothing particularly special about this one, the popular press jumped on it as if it was big news.

Tom Burt, Microsoft's Corporate VP for Customer Security & Trust triggered this by writing:

> *This week we observed cyberattacks by the threat actor Nobelium targeting government agencies, think tanks, consultants, and non-governmental organizations. This wave of attacks targeted approximately 3,000 email accounts at more than 150 different organizations. While organizations in the United States received the largest share of attacks, targeted victims span at least 24 countries. At least a quarter of the targeted organizations were involved in international development, humanitarian, and human rights work. Nobelium, originating from Russia, is the same actor behind the attacks on SolarWinds customers in 2020. These attacks appear to be a continuation of multiple efforts by Nobelium to target government agencies involved in foreign policy as part of intelligence gathering efforts. Nobelium launched this week’s attacks by gaining access to the Constant Contact account of USAID.*

So, okay, yeah. This was a significant **phishing** attack enabled by the breach of a single mass mailing account at the mass mailing service, Constant Contact. If USAID's account is compromised, then the result is pretty much guaranteed to be exactly that happened, and with exactly the demographics that we saw. This wasn't in any way targeting those specific organizations. The targets were entirely a function of the account that was compromised. Now, sure, perhaps USAID was targeted. But were this not tied back to the same group who were behind SolarWinds — though this bears zero resemblance to that amazing work — it would never have made the news.

And, by the way, everybody has their own name for these guys. Microsoft wants to call them Nobelium. But we know them better as APT29, or maybe “The Dukes” or “Cozy Bear.” FireEye calls them UNC2452, Palo Alto Networks' Unit 42 refers to them as SolarStorm, Crowdstrike calls them StellarParticle, Volexity refers to them as Dark Halo and Secureworks likes to call them Iron Ritual. It would be far less confusing if we could all agree to just call them APT29. But, no.

And as I said, otherwise this was just your run-of-the-mill eMail phishing attack. The eMail sent to those individuals on the USAID mailing list had an HTML attachment. When the HTML was opened by the eMail’s recipient, JavaScript in the HTML would write an ISO file to disc and then encourage its recipient to open it. That would result in the ISO file being mounted, at which point a shortcut link would auto-execute a DLL contained in the ISO, which would, in turn, result in the Cobalt Strike Beacon being executed on the system. In other words... yeah, don’t click links in eMail. But since it really did come from USAID, and was likely convincing, some recipients might have opened the attachment and proceeded to get themselves infected.

Again, not good, but not any sort of high level dastardly sophisticated attack. If we want to blame anyone, I’d look at Microsoft. Why exactly is it that opening an attachment in an eMail can launch an HTML page that can run JavaScript to write an ISO file to our local machine’s mass storage and then mount the ISO and launch a DLL it contains? How is that ever going to be a safe thing to let users do?

In any event, if you happened to hear about Microsoft warning of some huge new attack targeting 150 different organizations... those were just 3,000 phishing eMails sent to the addresses that were reachable from inside USAID's Constant Contact account. No biggie.

The Great Encryption Struggle

India recently put in place new regulations that would require messaging apps — such as WhatsApp — to trace the "first originator" of messages shared on the platform, thus breaking encryption protections.

Since India contains WhatsApp's largest user base by count, at 530 million, WhatsApp has sued the Government of India (good luck with that) over these new Internet regulations. A WhatsApp spokesperson said:

"Requiring messaging apps to 'trace' chats is the equivalent of asking us to keep a fingerprint of every single message sent on WhatsApp, which would break end-to-end encryption and fundamentally undermines people's right to privacy. We have consistently joined civil society and experts around the world in opposing requirements that would violate the privacy of our users."

Okay, wait a minute. Wasn't it WhatsApp that was changing their privacy agreement, in contravention of their original commitment to never share data with their parent company, Facebook... to now do exactly that? And in going so, triggered a mass exodus from the WhatsApp platform, whereupon they quickly back-pedalled? Yeah, well...

In any event, India's new legislation reads: *"Significant social media intermediaries [which are defined as being platforms with 5 million or more registered users in India] providing services primarily in the nature of messaging shall enable identification of the first originator of the information that is required only for the purposes of prevention, detection, investigation, prosecution or punishment of an offence related to sovereignty and integrity of India, the security of the State, friendly relations with foreign States, or public order or of incitement to an offence relating to the above or in relation with rape, sexually explicit material or child sexual abuse material punishable with imprisonment for a term of not less than five years. Intermediary shall not be required to disclose the contents of any message or any other information to the first originator."*

The new legislation also requires the providers of qualifying messaging platforms to remove non-consensual sexually explicit content within 24 hours, and appoint a resident grievance officer for acknowledging and addressing complaints from users and victims.

So this forms another step in the accumulating battle over encryption. States are understandably demanding access to their citizens' communications for the prevention of abuse that is doubtless helped along by unbreakable encryption. WhatsApp is currently also doing battle with Brazil over their proposed legislation that would "force companies to add a permanent identity stamp to the private messages people send."

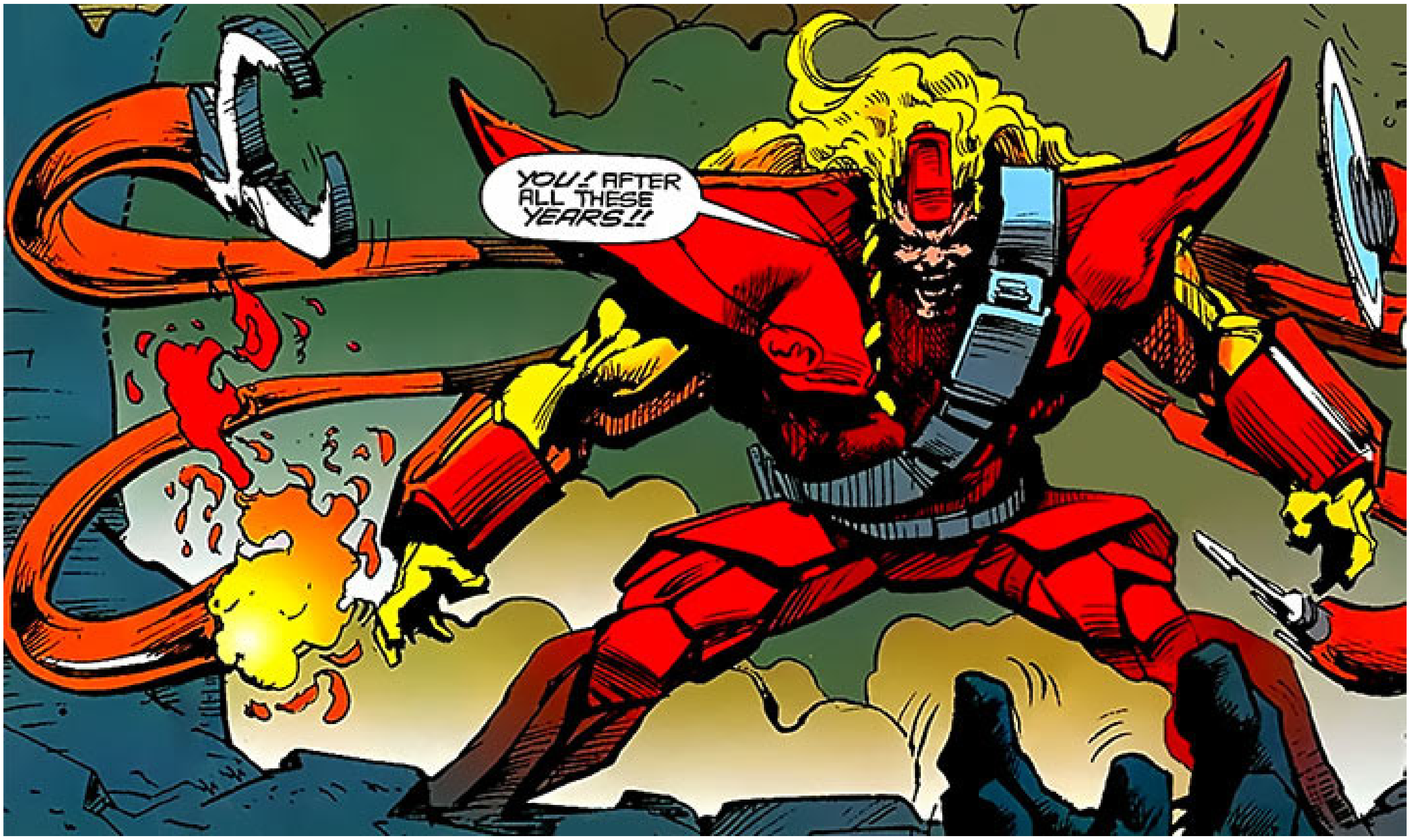
In response to WhatsApp's legal challenge to India's new digital rules, on grounds of violation of user privacy (and, once again, look who's talking), the Indian government last Wednesday said it is committed to the right to privacy of citizens, but added that it's subject to "reasonable restrictions" and that "no fundamental right is absolute."

Unfortunately, mathematics is absolute. You either do the best possible job you can to insure privacy — and offer it as a compelling benefit of your service — or you don't.

Sci-Fi

Hail Mary — And just a quick note that after finished book #10 of The Frontiers Saga (yes, my 3rd reading through it because it's just so much fun). So I've paused at a good point, and I've just cracked the cover of Andy Weir's Hail Mary. My Kindle tells me that I'm at 10%... and I really am enjoying the story so far. I really appreciate Andy's writing style and his humor. It's just right for me. So I have NO IDEA what's in store, but I may be on the other side of it by our next podcast. :)

Epsilon Red



The Security Tech-Press has jumped on the news of another new player in the burgeoning ransomware field with headlines including:

Sophos: A new ransomware enters the fray: Epsilon Red
BleepingComputer: New Epsilon Red ransomware hunts unpatched Microsoft Exchange servers
Silicon Angle: New 'Epsilon Red' ransomware is targeting unpatched Microsoft Exchange servers
Heimdal Security: Epsilon Red Ransomware Goes After Unpatched Microsoft Exchange Servers
Security Week: Cybercriminals Target Companies With New 'Epsilon Red' Ransomware

The name of the malicious group comes from the Marvel Universe. The character known as "Epsilon Red" — depicted above — is a little-known character, interestingly, a Russian super-soldier with four tentacles who can breathe in space.

What interested me about this was that Sophos encountered this new entry in the field several weeks ago and thoroughly took it apart. Well, inasmuch as there was anything to be taken apart. This thing, such as it is, is predominantly a collection of PowerShell scripts which, for me, begged the question: "What explains the method of this thing's construction?" Upon reflection, if I were to give this ransomware a longer name, I'd call it: "Epsilon Red: Cashing in on a craze."

Sophos security researcher Andrew Brandt phrased it in his report this last Friday, "A bare-bones ransomware offloads most of its functionality to a cache of PowerShell scripts." Andrew wrote:

In the past week, Sophos analysts uncovered a new ransomware written in the Go programming language that calls itself Epsilon Red. The malware was delivered as the final executable payload in a hand-controlled attack against a US-based business in the hospitality industry in which every other earlier-stage component was a PowerShell script.

Based on the cryptocurrency address provided by the attackers, it appears that at least one of their victims paid a ransom of 4.29BTC on May 15th (valued at roughly $210,000 on that date).

While the name and the tooling were unique to this attacker, the ransom note left behind on infected computers resembles the note left behind by REvil ransomware, though it adds a few minor grammatical corrections. There were no other obvious similarities between the Epsilon Red ransomware and REvil.

[So I assume Andrew is suggesting that the purveyors of this new ransomware borrowed the ransomware note used by REvil but otherwise wrote their own malware from scratch, as does appear to be likely.]

It appears that an enterprise Microsoft Exchange server was the initial point of entry by the attackers into the enterprise network. It isn't clear whether this was enabled by the ProxyLogon exploit or another vulnerability, but it seems likely that the root cause was an unpatched [Exchange] server. From that machine, the attackers used WMI (Windows Management Instrumentation) to install other software onto machines inside the network that they could reach from the Exchange server.

[Andrew proceeds to explain that...]

During the attack, the threat actors launched a series of PowerShell scripts, numbered **1.ps1** through **12.ps1** (as well as some that just were named with a single letter from the alphabet), that prepared the attacked machines for the final ransomware payload and, ultimately delivered and initiated it.

The PowerShell orchestration was, itself, created and triggered by a PowerShell script named **RED.ps1** that was executed on the target machines using WMI. The script retrieves and unpacks into the system32 folder a .7z archive that contains the rest of the PowerShell scripts, the ransomware executable, and another executable.

It uses the machine's Task Scheduler to run scripts numbered 1 through 12, except for 7 and 8, it also creates tasks for scripts named "S" and "C."

For example, when attackers ran the **2.ps1** script on a machine, it executed a command that deleted the Volume Shadow Copies from the computer. This is an important precursor to the attack, as these files could be used to recover some or all of the files encrypted by the attackers.

A PowerShell script named **c.ps1** appears to be a clone of an open source tool called Copy-VSS, part of a suite of penetration tester tools named Nishang. The Copy-VSS script permits an attacker to copy the SAM file, which an attacker could use to retrieve and crack passwords saved on the computer.

The PowerShell scripts also use a rudimentary form of obfuscation in which the threat actors appear to have added in some square brackets and braces at random into the script, thus breaking up the lines of PowerShell script code, and then use a command that strips out those brackets.

While this technique doesn't have much of an effect on our ability to analyze the files after the fact, it might be just good enough to evade the detection of an anti-malware tool that's scanning the files on the hard drive for a few minutes, which is all the attackers really need.

The **red.ps1** script unpacks RED.7z into the %SYSTEM%\RED directory, then creates scheduled tasks that run the unpacked scripts. But then it waits one hour, and executes commands that modify the Windows Firewall rules such that the firewall blocks inbound connections on all TCP ports except the Remote Desktop Protocol's 3389/tcp and the communications port used by a commercial tool called Remote Utilities, 5650/tcp.

[Sigh. As I've mentioned before, "Remote Utilities" is an excellent remote desktop facility. It's the one I've chosen for my use. Lorrie uses it to remotely manage the laptops being used by her home neurofeedback clients and my tech support guy, Greg, who runs a computer consulting business on the side, uses it to manage hundreds of his client's machines. So it's annoying to see it abused. But I suppose that's how Mozilla felt when their wonderful free "FireFox Send" facility was abused by bad guys. Anyway... Andrew notes that...]

Oddly, [the port blocking] does this by first blocking inbound traffic to ports 80 and 443, then redundantly blocks entire large ranges of ports that include 80 and 443, but also exclude the RDP and Remote Utilities ports: 1-3388, 3390-5649, and 5651-65352.

Upon closer inspection, one of the first things the attackers did after gaining access to the target's network was to download and install a copy of Remote Utilities and the Tor Browser, so this seems like a way to reassure themselves they will have an alternate foothold if the initial access point gets locked down.

Andrew then notes a few of the attractive features of Remote Utilities. He writes:

The commercial Remote Utilities software, used by the criminals, has several features they might find helpful. For one thing, they can use it for free. Anyone can submit an email address through the company's website and receive a free license key by email that allows them to use the full capability of the product on up to 10 machines, in perpetuity.

The company's "Viewer" software includes the ability for a licensed user to generate a digitally signed executable installer, pre-configured with a password and other preferences embedded into the .exe. Users choose their options, which get transmitted back to the company via the application to generate a unique "One-Click package" executable the program then downloads. The threat actor can then deploy this installer, which runs unattended, and automatically synchronizes to their Remote Utilities Viewer console. And the Viewer console can also serve as a Remote Desktop client utility, as a convenience.

[As I said... "Remote Utilities" is terrifically cool and functional software.]

We found that the attackers had generated at least two of these “One-Click installer” executables, which they downloaded to several machines on the target’s network and ran. The installer was named rutserv.exe and the attackers stored it in different filesystem locations on different machines they downloaded it to.

Initially, the malware runs the scripts numbered 9 and 12, this is followed by a 180 second delay, before then creating the tasks for 1 through 6, 10, 11, and S.ps1 and C.ps1. By default, the attackers extracted these Files to a folder named RED under the %SYSTEM% path. Each of these scripts accomplishes a specific task the threat actors use to prepare the system prior to launching the ransomware. Many of these tasks involve hindering security or backup tools, but also involve disabling or killing processes that, if they were running, might prevent a complete encryption of the valuable data on the hard drive.

It isn’t clear whether the attackers were just being thorough or if they weren’t sure they could do what they set out to do, but in several cases the scripts issue redundant commands to accomplish the same goal using slightly different methods.

For instance, the **1.ps1** file looks for processes that contain any of the following strings in their process name, and attempts to kill them:

'sql','Sql','SQL','BASup','Titan','SBAM','sbam','vipre','Vipre','Cylance','cylance','Senti','senti','sql','backup','veeam','outlook','word','excel','office','ocomm','dbsnmp','onenote','firefox','xfssvccon','infopath','wordpa','isqlplussvc','sql','dbeng50','mspub','mydesktopqos','ocautoupds','thunderbird','encsvc','oracle','mydesktopservice','thebat','agntsvc','steam','ocssd','tbirdconfig','synctime','visio','sqbcoreservice','winword','msaccess','powerpnt','mepocs','memtas','svc$','vss','sophos','crm','quickbooks','pos','qb','sage','SQL','prc','w3wp','java','store','ax32','dbs','wordpad','VeeamAgent','Backup','Cloud','Mbae','MB3','WRSA','rsa','wrsa'

These strings indicate the attackers are not only trying to shut down security tools, but also database services, backup programs, office applications, email clients, QuickBooks, and even Steam, the gaming platform.

2.ps1 deletes all the Volume Shadow Copies on the system by running a single command (vssadmin.exe delete shadows /all /quiet), while **3.ps1** disables automatic repairs that Windows might try to run upon a reboot.

4.ps1 then attempts to delete the Volume Shadow Copies using a different method:

```
wmic shadowcopy delete /nointeractive
Get-WmiObject Win32_ShadowCopy | % { $_.Delete() }
Get-WmiObject Win32_ShadowCopy | Remove-WmiObject
Get-WmiObject Win32_Shadowcopy | ForEach-Object { $_Delete(); }
Get-CimInstance Win32_ShadowCopy | Remove-CimInstance
```

5.ps1 executes two commands that, between them, delete Windows Event Logs, which would hinder an investigation.

Similarly to 1.ps1, **6.ps1** attempts to kill not processes but services, based on a list of strings that may appear in the services' names:

```
'sql','Sql','SQL','BASup','Titan','Cylance','cylance','Defend','NisSvc','Veeam','veeam','backup','Backup','rsa','wrsa','WRSA','RSA'
```

It also disables Windows defender by setting the following Windows Registry key:

```
reg add "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows Defender" /f /v DisableAntiSpyware /t REG_DWORD /d
```

9.ps1, which is executed first, attempts to invoke the Uninstaller for security software from Sophos, Trend Micro, Cylance, MalwareBytes, Sentinel One, Vipre, Webroot, and several cloud backup agents

10.ps1 then, redundantly, runs the dropped p.exe executable, which suspends the processes that contain the following strings, and clears their logs:

```
`'MpCmd','MsMp','Senti','senti','sql','backup','veeam','outlook','word','excel','office','ocomm','dbsnmp','onenote','firefox','xfssvccon','infopath','wordpa','isqlplussvc','sql','dbeng50','mspub','mydesktopqos','ocautoupds','thunderbird','encsvc','oracle','mydesktopservice','thebat','agntsvc','steam','ocssd','tbirdconfig','synctime','visio','sqbcoreservice','winword','msaccess','powerpnt','mepocs','memtas','svc$','vss','sophos','crm','quickbooks','pos','qb','sage','SQL','prc','w3wp','java','store','ax32','dbs','wordpad','VeeamAgent','Backup','Cloud','Mbae','MB3'
```

And **11.ps1** adds yet another layer of redundancy, executing the following commands that delete Volume Shadow Copies (again, for the third time!) as well as changing recovery options and clearing event logs in yet another way.

```
'vssadmin.exe Delete Shadows /All /Quiet',
'bcdedit /set {default} recoveryenabled no',
'wmic shadowcopy delete',
'wbadmin delete backup',
'wbadmin delete systemstatebackup -keepversions:0',
'bcdedit /set {default} bootstatuspolicy ignoreallfailures',
'bcdedit /set {default} recoveryenabled no',
'wevtutil.exe clear-log Application',
'wevtutil.exe clear-log Security',
'wevtutil.exe clear-log System',
'wbadmin delete systemstatebackup',
'wbadmin delete catalog -quiet',
'bootstatuspolicy ignoreallfailures'
```

This level of redundancy may be an indication that this threat actor is unsure of their own tools' capabilities, but aren't willing to take any chances.

12.ps1 grants the "Everyone" group access permissions to every drive letter that might exist on the machine to ensure as many files are encrypted as possible.

The **red.ps1** script also deletes itself, the .7z archive, and the local copy of 7zip from the system when it runs, removing key evidence.

In addition to the ransomware executable itself, Sophos recovered and analyzed another ancillary executable that the attackers deployed on the target machines. The file, just called **p.exe**, appears to be a custom-compiled version of an open source tool called EventCleaner, which was created to erase or manipulate the contents of Windows event logs. The attackers used the p.exe component to clean up evidence of what they had done.

We also mentioned that there were other PowerShell scripts delivered in the .7z archive the attackers dropped on targeted machines. While we saw no evidence they were executed in the context of this attack, the scripts numbered 7, 8, and 9 serve important purposes. **7.ps1** logs off practically all open sessions on the computer; **8.ps1** is a redundant copy of the same firewall rules script included in RED.ps1.

The ransomware itself, called **RED.exe**, is a 64-bit Windows executable programmed in the Go language, compiled using MinGW, and packed with a modified version of UPX.

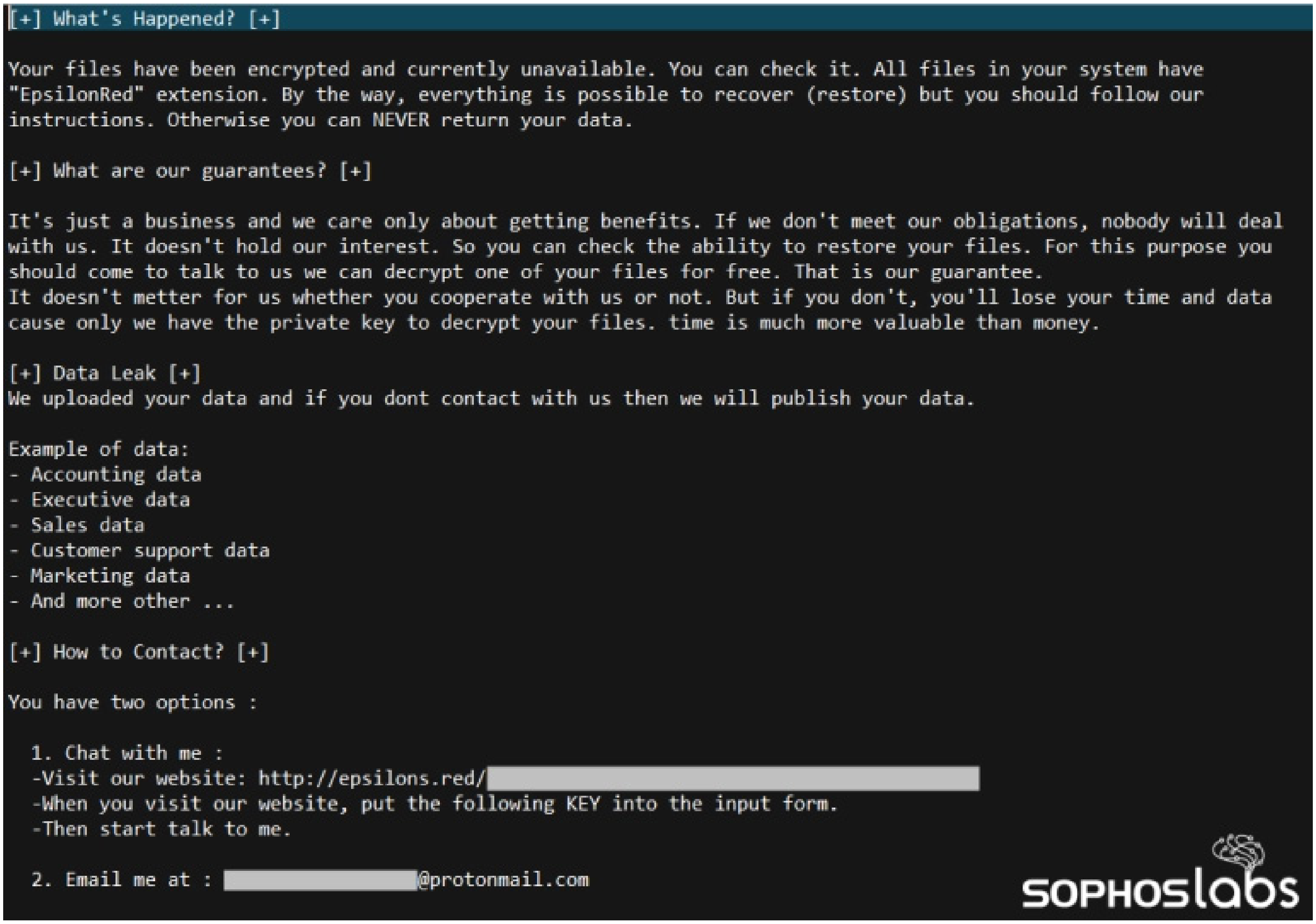
The executable contains some code taken from an open source project called GoDirWalk, which gives it the ability to scan the hard drive on which it's running for directory paths and compile them into a list. The ransomware then spawns a new child process that encrypts each subfolder separately, which after a short amount of time results in a lot of copies of the ransomware process running simultaneously.

The ransomware itself is quite small as it only really is used to perform the encryption of the files on the targeted system. It makes no network connections, and because functions like killing processes or deleting the Volume Shadow Copies have been outsourced to the PowerShell scripts, it's really quite a simple program.

In the sample we've seen, it doesn't even contain a list of targeted file types or file extensions. It will encrypt everything inside the folders it decides to encrypt, including other executables and DLLs, which can render programs or the entire system nonfunctional, if the ransomware decides to encrypt the wrong folder path. After it encrypts each file, it appends a file suffix of ".epsilonred" to the files, and drops a ransom note in each folder.

Interestingly, the ransom note closely resembles a shortened version of the note used by REvil. But where the REvil note is riddled with spelling and grammatical errors, the note delivered by Epsilon Red has gone through a few edits to make its text more readable to an audience of native English speakers.

The ransom note reads:



```
[+] What's Happened? [+]

Your files have been encrypted and currently unavailable. You can check it. All files in your system have
"EpsilonRed" extension. By the way, everything is possible to recover (restore) but you should follow our
instructions. Otherwise you can NEVER return your data.

[+] What are our guarantees? [+]

It's just a business and we care only about getting benefits. If we don't meet our obligations, nobody will deal
with us. It doesn't hold our interest. So you can check the ability to restore your files. For this purpose you
should come to talk to us we can decrypt one of your files for free. That is our guarantee.
It doesn't metter for us whether you cooperate with us or not. But if you don't, you'll lose your time and data
cause only we have the private key to decrypt your files. time is much more valuable than money.

[+] Data Leak [+]
We uploaded your data and if you dont contact with us then we will publish your data.

Example of data:
- Accounting data
- Executive data
- Sales data
- Customer support data
- Marketing data
- And more other ...

[+] How to Contact? [+]

You have two options :

  1. Chat with me :
  -Visit our website: http://epsilons.red/[redacted]
  -When you visit our website, put the following KEY into the input form.
  -Then start talk to me.

  2. Email me at : [redacted]@protonmail.com
```

Note that Sophos discovered no indication that any of this hodgepodge of bits and pieces of PowerShell scripting or small single-function executables ever did **any** exfiltration of the victim’s data. So, given everything else we’ve seen about this concoction, it seems almost certain that no actual exfiltration of any kind was done, and that the threat to publish data is entirely empty. They don’t evidence having any infrastructure to back up either their threat or their victim’s data.

And in order to engage with the attackers, their victims are instructed to visit a specific page on a website located, not on the dark web where other ransomware sends their victims, but on the regular normal public Internet at the domain “epsilons” (plural) =dot= “red”. Note that since this was written up the “epsilons.red” domain has disappeared. No surprise there.

Operating styles leave their own sort of fingerprint. What do the facts in evidence suggest?

The encryption malware, as described, is bare bones, but it does get the job done. It uses a publicly available directory recursion tool to build a list of directories. Then it spawns individual encryptors for each discovered directory. Each encryptor operates indiscriminately without any file extension based encryption filtering. It simply encrypts the entire contents of every directory it’s run in. Does it even use a public key? Maybe. But it might simply use a static key. In which case obtaining a copy of the encryption EXE — which this PowerShell based approach takes pains to prevent — would allow for decryption without paying ransom. We’ve certainly seen many instances of lame ransomware whose analysis resulted in the creation of free decrypters.

And besides that one encryption EXE, everything else was either freely obtained and reused, or written as PowerShell scripts. Being scripts, that saved them from issuing the commands by hand. But it is certainly far lower-tech than the ransomware systems that have come before. And this certainly doesn't lend itself to the affiliate model.

This guy used a public website for his extortion and cribbed much of the text of the ransom note being used by the REvil gang. Taken as a whole, more than anything else, this has all the hallmarks of someone in a hurry who's attempting to get into the diminishing pool of Exchange Server machines before they'll all gone. As we all know too well, it's trivial to both locate exchange servers and to penetrate any that haven't been patched since March.

But as I noted at the start, Sophos did track down the cryptocurrency address being used by these guys — or, as seems more likely — this person, since this feels like a one-person shop. And they found that someone had paid the equivalent of $210,000. It would be very interesting to know whether the victim who paid that money ever got their files back. The reason I'm curious is that this attacker didn't bother to set up a Tor-hidden .onion site, and the exfiltrated data extortion threat has all the appearances of being empty. So what else might be? In our current environment of rampant and high profile ransomware, it seems inevitable that there will be low-end attackers — probably like this guy — who trade on the reputation — such as it is — of high-end ransomware which DOES go to some pains to assure the successful recovery of encrypted files.

Any naïve victim, who doesn't know any better, would have no way of discriminating whether they've been attacked by a — and I can't believe I'm saying this — reputable ransomware attacker operating in good faith who actually has developed the capability to restore encrypted systems, versus a half-baked attacker who's trading on the public's knowledge that once ransoms are paid it's actually possible to bring systems back online.

Presumably, the attacker is able to decrypt a single file as proof of their ability to do so. At least he does offer that in his ransom demand. And since all files were encrypted indiscriminately, the affected systems probably no longer boot or run at all. So each one would need to be booted from recovery media in order to gain access to the system's mass storage.

What a mess.