A Bazar start: How one hospital thwarted a Ryuk ransomware outbreak

By Brian Donohue

Published: 2020-10-29 · Archived: 2026-04-05 20:33:08 UTC

In mid-October, a variety of detection analytics alerted the Red Canary CIRT to execution, reconnaissance, and lateral movement activity on the network of a medical center. Within minutes, we observed Cobalt Strike and other malicious tools that all pointed toward a troubling conclusion: the hospital was probably a few hours away from a full-blown Ryuk ransomware outbreak. Thanks in no small part to our incident response partners at Kroll, whose Responder team rapidly engaged and began active containment steps as we detected threats, that didn't happen.

This week, news has spread that many hospitals in the United States are being attacked by Ryuk ransomware—and are very likely experiencing some version of what we've just described. Despite being in the throes of a pandemic that's already over-burdening global public health infrastructure, ransomware crews have been escalating their operations against hospitals for months now.

These attacks are abhorrent. The people responsible for them are despicable. And we, like DHS CISA, Mandiant, and others in the information security community, want to help the hospitals that care for all of us however we can. So we're sharing the details of how we thwarted these operators earlier this month—in the hopes you can take this information and better protect your own organizations.

Background

We've been following all the recent reporting and tweets about hospitals being attacked by Ryuk ransomware. But Ryuk isn't new to us… we've been tracking it for years. More important than just looking at Ryuk ransomware itself, though, is looking at the operators behind it and their tactics, techniques, and procedures (TTPs)—especially those used *before they encrypt any data*. The operators of Ryuk ransomware are known by different names in the community, including "WIZARD SPIDER," "UNC1878," and "Team9." The malware they use has included TrickBot, Anchor, Bazar, Ryuk, and others.

Many in the community have shared reporting about these operators and malware families (check out the end of this blog post for links to some excellent reporting from other teams), so we wanted to focus narrowly on what we've observed: BazarLoader/BazarBackdoor (which we're collectively calling Bazar) used for initial access, followed by deployment of Cobalt Strike, and hours or days later, the potential deployment of Ryuk ransomware. We have certainly seen TrickBot lead to Ryuk ransomware in the past. This month, however, we've observed Bazar as a common initial access method, leading to our assessment that Bazar is a greater threat at this time for the eventual deployment of Ryuk.

What we've seen and how you can detect it

While every ransomware outbreak can play out in different ways, we want to focus on the attack we saw in mid-October and stopped before ransomware was deployed. **As we walk through this specific attack, we'll identify 10 detection opportunities that work for us—and we hope they'll work for you too.** This attack can serve as a functional example for what you might expect to see if you're responsible for defending a healthcare organization.

If you're interested in the MITRE ATT&CK® techniques covered by this incident, check out the ATT&CK Navigator layer here. You can learn more about ATT&CK Navigator here.

This graphic provides an overall representation of how the attack unfolded. We'll dive into the details, complete with detection opportunities, below.

Initial access came by way of a phishing email containing a PDF attachment. The user opened this attachment and clicked on a link in the PDF, which connected to Google Drive and downloaded a file named `Report[mm]-[dd].exe` (for example, the file name would be `Report10-29.exe` if the email was delivered on October 29). This `.exe` is known as Bazar, which has different components known by the community as BazaLoader, BazarLoader, and BazarBackdoor.

Detection Opportunity 1: Process hollowing of `cmd.exe`

This `.exe` file used process hollowing techniques to inject into `cmd.exe`. You can identify this process hollowing, as we did, by looking for instances of the Windows Command prompt (`cmd.exe`) executing without any command-line parameters and establishing a network connection. If that's too noisy, you could try limiting the network connections to port 443 or 53. You could also limit false positives by looking for child processes spawned by the hollowed `cmd.exe` process. Typical child processes associated with Bazar include: `cmd.exe` , `svchost.exe` , `explorer.exe` , `nltest.exe` , and `net.exe` , as shown in the process tree below.

Detection Opportunity 2: Enumerating domain trusts activity with `nltest.exe`

We then observed several reconnaissance commands associated with Bazar. Specifically, we observed the adversary using `nltest.exe` to make domain trust determinations. While you probably can't disable `nltest.exe`, looking for instances of it executing with a command line that includes `/dclist:<domain>`, `/domain_trusts` or `/all_trusts` has proven to be a very high-fidelity analytic for us to catch both Bazar (in this incident) as well as TrickBot (in past incidents). In fact, based on this overlap, it appears likely that Bazar may be reusing some code from TrickBot, which could lead to some confusion over which malware family is which.

Detection Opportunity 3: Enumerating domain admins with net group

We also saw the adversary attempting to enumerate Windows domain administrator accounts, a behavior that we commonly associate with ransomware operators. In particular, we find it useful to look for `net group "domain admins" /dom` and `net group "domain admins" /domain`.

Detection Opportunity 4: Process hollowing of `explorer.exe`

During this phase, we also saw the adversary use process hollowing with both `explorer.exe` and `svchost.exe`. We observed `explorer.exe` spawning `svchost.exe` —this isn't normal, so you should look for that in your environment. More broadly, you can look for `svchost.exe` processes where the parent is not `services.exe` to identify this and other malicious activity. (If you've never checked it out, we highly recommend looking at the SANS Hunt Evil poster!)

Another way we detected this activity was by looking for `svchost.exe` with no command-line options. Legitimate instances of `svchost.exe` should almost always have command-line options that include `-k` and the name of a service the process manages. Instances of `svchost.exe` with no command-line options are suspicious and may indicate that `svchost.exe` has been spawned to host injected code—like we saw in this incident.

Detection Opportunity 5: Attempted lateral movement via WMI + PowerShell + Cobalt Strike

Next, a Cobalt Strike binary was dropped on the endpoint as a `.dll` file and executed by `rundll32.exe`. With that, the intrusion began spreading laterally via Cobalt Strike. The operators used Windows Management Instrumentation (WMI) in their lateral movement attempt. WMI spawned `cmd.exe`, which subsequently spawned PowerShell with an encoded command line. This encoded PowerShell creates another Cobalt Strike Beacon. We've found that looking for encoded PowerShell is a great way to catch this specific evil and a lot of other evil, too. In this incident, we saw a command line that began with:
