Alert (AA20-302A)

More Alerts

Ransomware Activity Targeting the Healthcare and Public Health Sector

Original release date: October 28, 2020 | Last revised: October 29, 2020

Summary

This advisory was updated to include information on Conti, TrickBot, and BazarLoader, including new IOCs and Yara Rules for detection.

> *This advisory uses the MITRE Adversarial Tactics, Techniques, and Common Knowledge (ATT&CK®) version 7 framework. See the ATT&CK for Enterprise version 7 for all referenced threat actor tactics and techniques.*

This joint cybersecurity advisory was coauthored by the Cybersecurity and Infrastructure Security Agency (CISA), the Federal Bureau of Investigation (FBI), and the Department of Health and Human Services (HHS). This advisory describes the tactics, techniques, and procedures (TTPs) used by cybercriminals against targets in the Healthcare and Public Health (HPH) Sector to infect systems with ransomware, notably Ryuk and Conti, for financial gain.

CISA, FBI, and HHS have credible information of an increased and imminent cybercrime threat to U.S. hospitals and healthcare providers. CISA, FBI, and HHS are sharing this information to provide warning to healthcare providers to ensure that they take timely and reasonable precautions to protect their networks from these threats.

Click here for a PDF version of this report.

Key Findings

- CISA, FBI, and HHS assess malicious cyber actors are targeting the HPH Sector with TrickBot and BazarLoader malware, often leading to ransomware attacks, data theft, and the disruption of healthcare services.
- These issues will be particularly challenging for organizations within the COVID-19 pandemic; therefore, administrators will need to balance this risk when determining their cybersecurity investments.

Technical Details

Threat Details

The cybercriminal enterprise behind TrickBot, which is likely also the creator of BazarLoader malware, has continued to develop new functionality and tools, increasing the ease, speed, and profitability of victimization. These threat actors increasingly use loaders—like TrickBot and BazarLoader (or BazarBackdoor)—as part of their malicious cyber campaigns. Cybercriminals disseminate TrickBot and BazarLoader via phishing campaigns that contain either links to malicious websites that host the malware or attachments with the malware. Loaders start the infection chain by distributing the payload; they deploy and execute the backdoor from the command and control (C2) server and install it on the victim's machine.

TrickBot

What began as a banking trojan and descendant of Dyre malware, TrickBot now provides its operators a full suite of tools to conduct a myriad of illegal cyber activities. These activities include credential harvesting, mail exfiltration, cryptomining, point-of-sale data exfiltration, and the deployment of ransomware, such as Ryuk and Conti.

In early 2019, the FBI began to observe new TrickBot modules named Anchor, which cyber actors typically used in attacks targeting high-profile victims—such as large corporations. These attacks often involved data exfiltration from networks and point-of-sale devices. As part of the new Anchor toolset, TrickBot developers created `anchor_dns`, a tool for sending and receiving data from victim machines using Domain Name System (DNS) tunneling.

`anchor_dns` is a backdoor that allows victim machines to communicate with C2 servers over DNS to evade typical network defense products and make their malicious communications blend in with legitimate DNS traffic. `anchor_dns` uses a single-byte `XOR` cipher to encrypt its communications, which have been observed using key `0xB9`. Once decrypted, the string `anchor_dns` can be found in the DNS request traffic.

TrickBot Indicators of Compromise

After successful execution of the malware, TrickBot copies itself as an executable file with a 12-character randomly generated file name (e.g. `mfjdieks.exe`) and places this file in one of the following directories.

- C:\Windows\
- C:\Windows\SysWOW64\
- C:\Users\[Username]\AppData\Roaming\

Once the executable is running and successful in establishing communication with C2s, the executable places appropriate modules downloaded from C2s for the infected processor architecture type (32 or 64 bit instruction set), to the infected host's `%APPDATA%` or `%PROGRAMDATA%` directory, such as `%AppData\Roaming\winapp`. Some commonly named plugins that are created in a Modules subdirectory are (the detected architecture is appended to the module filename, e.g., `importDll32` or `importDll64`):

- `Systeminfo`
- `importDll`
- `outlookDll`
- `injectDll` with a directory (ex. `injectDLL64_configs`) containing configuration files:
  - `dinj`
  - `sinj`
  - `dpost`
- `mailsearcher` with a directory (ex. `mailsearcher64_configs`) containing configuration file:
  - `mailconf`
- `networkDll` with a directory (ex. networkDll64_configs) containing configuration file:
  - `dpost`
- `wormDll`
- `tabDll`
- `shareDll`

Filename `client_id` or `data` or `FAQ` with the assigned bot ID of the compromised system is created in the malware directory. Filename `group_tag` or `Readme.md` containing the TrickBot campaign IDs is created in the malware directory.

The malware may also drop a file named `anchorDiag.txt` in one of the directories listed above.

Part of the initial network communications with the C2 server involves sending information about the victim machine such as its computer name/hostname, operating system version, and build via a base64-encoded `GUID`. The `GUID` is composed of `/GroupID/ClientID/` with the following naming convention:

`/anchor_dns/[COMPUTERNAME]_[WindowsVersionBuildNo].[32CharacterString]/`.

The malware uses scheduled tasks that run every 15 minutes to ensure persistence on the victim machine. The scheduled task typically uses the following naming convention.

`[random_folder_name_in_%APPDATA%_excluding_Microsoft]`

`autoupdate#[5_random_numbers] (e.g., Task autoupdate#16876)`.

After successful execution, `anchor_dns` further deploys malicious batch scripts (`.bat`) using PowerShell commands.

The malware deploys self-deletion techniques by executing the following commands.

- `cmd.exe /c timeout 3 && del C:\Users\[username]\[malware_sample]`
- `cmd.exe /C PowerShell \"Start-Sleep 3; Remove-Item C:\Users\[username]\[malware_sample_location]\"`

The following domains found in outbound DNS records are associated with `anchor_dns`.

- `kostunivo[.]com`
- `chishir[.]com`
- `mangoclone[.]com`
- `onixcellent[.]com`

This malware used the following legitimate domains to test internet connectivity.

- `ipecho[.]net`
- `api[.]ipify[.]org`
- `checkip[.]amazonaws[.]com`
- `ip[.]anysrc[.]net`
- `wtfismyip[.]com`
- `ipinfo[.]io`
- `icanhazip[.]com`
- `myexternalip[.]com`
- `ident[.]me`

Currently, there is an open-source tracker for TrickBot C2 servers located at https://feodotracker.abuse.ch/browse/trickbot/.

The `anchor_dns` malware historically used the following C2 servers.

- `23[.]95[.]97[.]59`
- `51[.]254[.]25[.]115`
- `193[.]183[.]98[.]66`
- `91[.]217[.]137[.]37`
- `87[.]98[.]175[.]85`

TrickBot YARA Rules

```
rule anchor_dns_strings_filenames {
  meta:
    description = "Rule to detect AnchorDNS samples based off strings or filenames used in malware"
    author = "NCSC"
    hash1 = "fc0efd612ad528795472e99cae5944b68b8e26dc"
    hash2 = "794eb3a9ce8b7e5092bb1b93341a54097f5b78a9"
    hash3 = "9dfce70fded4f3bc2aa50ca772b0f9094b7b1fb2"
    hash4 = "24d4bbc982a6a561f0426a683b9617de1a96a74a"
  strings:
    $ = ",Control_RunDLL \x00"
    $ = ":$GUID" ascii wide
    $ = ":$DATA" ascii wide
    $ = "/1001/"
    $ = /(\x00|\xCC)qwertyuiopasdfghjklzxcvbnm(\x00|\xCC)/
    $ = /(\x00|\xCC)QWERTYUIOPASDFGHJKLZXCVBNM(\x00|\xCC)/
```

```
    $ = "start program with cmdline \"%s\""
    $ = "Global\\fde345tyhoVGYHUJKIOuy"
    $ = "ChardWorker::thExecute: error registry me"
    $ = "get command: incode %s, cmdid \"%s\", cmd \"%s\""
    $ = "anchorDNS"
    $ = "Anchor_x86"
    $ = "Anchor_x64"
  condition:
    (uint16(0) == 0x5A4D and uint16(uint32(0x3c)) == 0x4550) and 3 of them
}
```

```
rule anchor_dns_icmp_transport {
  meta:
    description = "Rule to detect AnchorDNS samples based off ICMP transport strings"
    author = "NCSC"
    hash1 = "056f326d9ab960ed02356b34a6dcd72d7180fc83"
  strings:
    $ = "reset_connection <- %s"
    $ = "server_ok <- %s (packets on server %s)"
    $ = "erase successfully transmitted packet (count: %d)"
    $ = "Packet sended with crc %s -> %s"
    $ = "send data confimation to server(%s)"
    $ = "data recived from <- %s"
    $ = "Rearmost packed recived (id: %s)"
    $ = "send poll to server -> : %s"
  condition:
    (uint16(0) == 0x5A4D and uint16(uint32(0x3c)) == 0x4550) and 3 of them
}
```

```
rule anchor_dns_config_dexor {
  meta:
    description = "Rule to detect AnchorDNS samples based off configuration deobfuscation (XOR 0x23 countup)"
    author = "NCSC"
    hash1 = "d0278ec015e10ada000915a1943ddbb3a0b6b3db"
    hash2 = "056f326d9ab960ed02356b34a6dcd72d7180fc83"
  strings:
    $x86 = {75 1F 56 6A 40 B2 23 33 C9 5E 8A 81 ?? ?? ?? ?? 32 C2 FE C2 88 81 ?? ?? ?? ?? 41 83 EE 01 75 EA 5E B8 ?? ?? ?? ?? C3}
    $x64 = {41 B0 23 41 B9 80 00 00 00 8A 84 3A ?? ?? ?? 00 41 32 C0 41 FE C0 88 04 32 48 FF C2 49 83 E9 01 75 E7}
  condition:
    (uint16(0) == 0x5A4D and uint16(uint32(0x3c)) == 0x4550) and any of them
}
```

```
rule anchor_dns_installer {
  meta:
    description = "Rule to detect AnchorDNS installer samples based off MZ magic under one-time pad or deobfuscation loop code"
    author = "NCSC"
    hash1 = "fa98074dc18ad7e2d357b5d168c00a91256d87d1"
    hash2 = "78f0737d2b1e605aad62af252b246ef390521f02"
  strings:
    $pre = {43 00 4F 00 4E 00 4F 00 55 00 54 00 24 00 00 00} //CONOUT$
    $pst = {6B 65 72 6E 65 6C 33 32 2E 64 6C 6C 00 00 00 00} //kernel32.dll
    $deob_x86 = {8B C8 89 4D F8 83 F9 FF 74 52 46 89 5D F4 88 5D FF 85 F6 74 34 8A 83 ?? ?? ?? ?? 32 83 ?? ?? ?? ?? 6A 00 88 45 FF 8D 45 F4 50 6A 01 8D 45 FF 50 51 FF 15 34 80 41 00 8B 4D F8 43 8B F0 81 FB 00 ?? ?? ?? 72 CC 85 F6 75 08}
    $deob_x64 = {42 0F B6 84 3F ?? ?? ?? ?? 4C 8D 8C 24 80 00 00 00 42 32 84 3F ?? ?? ?? ?? 48 8D 54 24 78 41 B8 01 00 00 00 88 44 24 78 48 8B CE 48 89 6C 24 20 FF 15 ?? ?? ?? ?? 48 FF C7 8B D8 48 81 FF ?? ?? ?? ?? 72 B8}
  condition:
    (uint16(0) == 0x5A4D and uint16(uint32(0x3c)) == 0x4550)
    and
      (  uint16(@pre+16) ^ uint16(@pre+16+((@pst-(@pre+16))\2)) == 0x5A4D
        or
        $deob_x86 or $deob_x64
      )
}
```

```
import "pe"
rule anchor_dns_string_1001_with_pe_section_dll_export_resolve_ip_domains {
  meta:
    description = "Rule to detect AnchorDNS samples based off /1001/ string in combination with DLL export name string, PE section .addr or IP resolution domains"
    author = "NCSC"
```

```
    hash1 = "ff8237252d53200c132dd742edc77a6c67565eee"
    hash2 = "c8299aadf886da55cb47e5cbafe8c5a482b47fc8"
  strings:
    $str1001 = {2F 31 30 30 31 2F 00} // /1001/
    $strCtrl = {2C 43 6F 6E 74 72 6F 6C 5F 52 75 6E 44 4C 4C 20 00} // ,Control_RunDLL
    $ip1 = "checkip.amazonaws.com" ascii wide
    $ip2 = "ipecho.net" ascii wide
    $ip3 = "ipinfo.io" ascii wide
    $ip4 = "api.ipify.org" ascii wide
    $ip5 = "icanhazip.com" ascii wide
    $ip6 = "myexternalip.com" ascii wide
    $ip7 = "wtfismyip.com" ascii wide
    $ip8 = "ip.anysrc.net" ascii wide
  condition:
    (uint16(0) == 0x5A4D and uint16(uint32(0x3c)) == 0x4550)
    and $str1001
    and (
        for any i in (0..pe.number_of_sections): (
          pe.sections[i].name == ".addr"
        )
      or
        $strCtrl
      or
        6 of ($ip*)
      )
}
```

```
rule anchor_dns_check_random_string_in_dns_response {
  meta:
    description = "Rule to detect AnchorDNS samples based off checking random string in DNS response"
    author = "NCSC"
    hash1 = "056f326d9ab960ed02356b34a6dcd72d7180fc83"
    hash2 = "14e9d68bba7a184863667c680a8d5a757149aa36"
  strings:
    $x86 = {8A D8 83 C4 10 84 DB 75 08 8B 7D BC E9 84 00 00 00 8B 7D BC 32 DB 8B C7 33 F6 0F 1F 00 85 C0 74 71 40 6A 2F 50 E8 ?? ?? ?? ?? 46 83 C4 08 83 FE 03 72 EA 85 C0 74 5B 83 7D D4 10 8D 4D C0 8B 75 D0 8D 50 01 0F 43 4D C0 83 EE 04 72 11 8B 02 3B 01 75 10 83 C2 04 83 C1 04 83 EE 04 73 EF 83 FE FC 74 2D 8A 02 3A 01 75 29 83 FE FD 74 22 8A 42 01 3A 41 01 75 1C 83 FE FE 74 15 8A 42 02 3A 41 02 75 0F 83 FE FF 74 08 8A 42 03 3A 41 03 75 02 B3 01 8B 75 B8}
    $x64 = {4C 39 75 EF 74 56 48 8D 45 DF 48 83 7D F7 10 48 0F 43 45 DF 49 8B FE 48 85 C0 74 40 48 8D 48 01 BA 2F 00 00 00 E8 ?? ?? ?? ?? 49 03 FF 48 83 FF 03 72 E4 48 85 C0 74 24 48 8D 55 1F 48 83 7D 37 10 48 0F 43 55 1F 48 8D 48 01 4C 8B 45 2F E8 ?? ?? ?? ?? 0F B6 DB 85 C0 41 0F 44 DF 49 03 F7 48 8B 55 F7 48 83 FE 05 0F 82 6A FF FF FF}
  condition:
    (uint16(0) == 0x5A4D and uint16(uint32(0x3c)) == 0x4550) and any of them
}
```

```
rule anchor_dns_default_result_execute_command {
  meta:
    description = "Rule to detect AnchorDNS samples based off default result value and executing command"
    author = "NCSC"
    hash1 = "056f326d9ab960ed02356b34a6dcd72d7180fc83"
    hash2 = "14e9d68bba7a184863667c680a8d5a757149aa36"
  strings:
    $x86 = {83 C4 04 3D 80 00 00 00 73 15 8B 04 85 ?? ?? ?? ?? 85 C0 74 0A 8D 4D D8 51 8B CF FF D0 8A D8 84 DB C7 45 A4 0F 00 00 00}
    $x64 = {48 98 B9 E7 03 00 00 48 3D 80 00 00 00 73 1B 48 8D 15 ?? ?? ?? ?? 48 8B 04 C2 48 85 C0 74 0B 48 8D 55 90 48 8B CE FF D0 8B C8}
  condition:
    (uint16(0) == 0x5A4D and uint16(uint32(0x3c)) == 0x4550) and any of them
}
```

```
rule anchor_dns_pdbs {
  meta:
    description = "Rule to detect AnchorDNS samples based off partial PDB paths"
    author = "NCSC"
    hash1 = "f0e575475f33600aede6a1b9a5c14f671cb93b7b"
    hash2 = "1304372bd4cdd877778621aea715f45face93d68"
    hash3 = "e5dc7c8bfa285b61dda1618f0ade9c256be75d1a"
    hash4 = "f96613ac6687f5dbbed13c727fa5d427e94d6128"
    hash5 = "46750d34a3a11dd16727dc622d127717beda4fa2"
  strings:
    $ = ":\\MyProjects\\secondWork\\Anchor\\"
    $ = ":\\simsim\\anchorDNS"
```

Network Best Practices

- Patch operating systems, software, and firmware as soon as manufacturers release updates.
- Check configurations for every operating system version for HPH organization-owned assets to prevent issues from arising that local users are unable to fix due to having local administration disabled.
- Regularly change passwords to network systems and accounts and avoid reusing passwords for different accounts.
- Use multi-factor authentication where possible.
- Disable unused remote access/Remote Desktop Protocol (RDP) ports and monitor remote access/RDP logs.
- Implement application and remote access allow listing to only allow systems to execute programs known and permitted by the established security policy.
- Audit user accounts with administrative privileges and configure access controls with least privilege in mind.
- Audit logs to ensure new accounts are legitimate.
- Scan for open or listening ports and mediate those that are not needed.
- Identify critical assets such as patient database servers, medical records, and teleheatlh and telework infrastructure; create backups of these systems and house the backups offline from the network.
- Implement network segmentation. Sensitive data should not reside on the same server and network segment as the email environment.
- Set antivirus and anti-malware solutions to automatically update; conduct regular scans.

Ransomware Best Practices

CISA, FBI and HHS do not recommend paying ransoms. Payment does not guarantee files will be recovered. It may also embolden adversaries to target additional organizations, encourage other criminal actors to engage in the distribution of ransomware, and/or fund illicit activities. In addition to implementing the above network best practices, the FBI, CISA and HHS also recommend the following:

- Regularly back up data, air gap, and password protect backup copies offline.
- Implement a recovery plan to maintain and retain multiple copies of sensitive or proprietary data and servers in a physically separate, secure location.

User Awareness Best Practices

- Focus on awareness and training. Because end users are targeted, make employees and stakeholders aware of the threats—such as ransomware and phishing scams—and how they are delivered. Additionally, provide users training on information security principles and techniques as well as overall emerging cybersecurity risks and vulnerabilities.
- Ensure that employees know who to contact when they see suspicious activity or when they believe they have been a victim of a cyberattack. This will ensure that the proper established mitigation strategy can be employed quickly and efficiently.

Recommended Mitigation Measures

System administrators who have indicators of a TrickBot network compromise should immediately take steps to back up and secure sensitive or proprietary data. TrickBot infections may be indicators of an imminent ransomware attack; system administrators should take steps to secure network devices accordingly. Upon evidence of a TrickBot infection, review DNS logs and use the `XOR` key of `0xB9` to decode `XOR` encoded DNS requests to reveal the presence of `Anchor_DNS`, and maintain and provide relevant logs.

GENERAL RANSOMWARE MITIGATIONS — HPH SECTOR

This section is based on CISA and Multi-State Information Sharing and Analysis Center (MS-ISAC)'s Joint Ransomware Guide, which can be found at https://www.cisa.gov/publication/ransomware-guide.

CISA, FBI, and HHS recommend that healthcare organizations implement both ransomware prevention and ransomware response measures immediately.

Ransomware Prevention

Join and Engage with Cybersecurity Organizations