Nimar Loader

By Joshua Platt

Published: 2021-03-01 · Archived: 2026-04-06 00:59:03 UTC

Baza (BazarLoader & BazarBackdoor) has been attributed to the organized cybercrime group behind Trickbot by multiple security vendors over the past year. Initially appearing around April of 2020 , the malware was spread in email campaigns utilizing infrastructure previously used to distribute Trickbot. [1] The term BazarLoader was coined due to the reliance and use of Blockchain-DNS and the associated bazar domains used to communicate with the controllers. Since then, the terms Baza or Bazarloader have been used interchangeably to reference this particular malware family.

After the initial appearance of Baza, multiple reports attributing various code to the malware family appeared. [2] [3] It became increasingly obvious that development was on going and multiple projects were under construction. The downside to such a public following, is the common acceptance of activity based on previous TTPS. Combining TTPs such as loading powershell stagers into red team utilities along with similar campaign structures and C2 traffic patterns, can quickly lead to incorrect attribution of malware families and actors.

On Feb 3, we detected campaigns previously attributed to Baza but the malware utilized request headers that were different. Several others noticed oddities as well.[4][5] Further analysis indicated something completely different and new. Nimar loader did not share the same codebase as the baza family.

NimarLoader or Nimrod recently mistaken as BazarLoader, is coded in Nim. The malware was obfuscated but did not contain the typical hardening seen in large scale campaigns. Since it was something newly developed, crypting the sample was unnecessary but it is curiously absent making analysis less difficult. From the strings, hints at the capability can be clearly seen.

```
verb in @["GET", "POST"]
@https
@fKHMP]
@SPMVDSD
job_type
job_args
job_results
@heartbeat set
@HEARTBEAT set Failed
@HEARTBEAT set
@Shellcode Done
@heartbeat
@shellcode
@handshake
@No update data recieved
agentId
```

```
pathAdj
pathNoun
seqNum
seqTotal
jobIdHeader
userAgent
serverPubKeyEncoded
jitterParamStr
sessionKey
sessionDie
```

The command table allows to handshake, heartbeat, shellcode, powershell or cmd, as evidenced in the strings seen during execution.

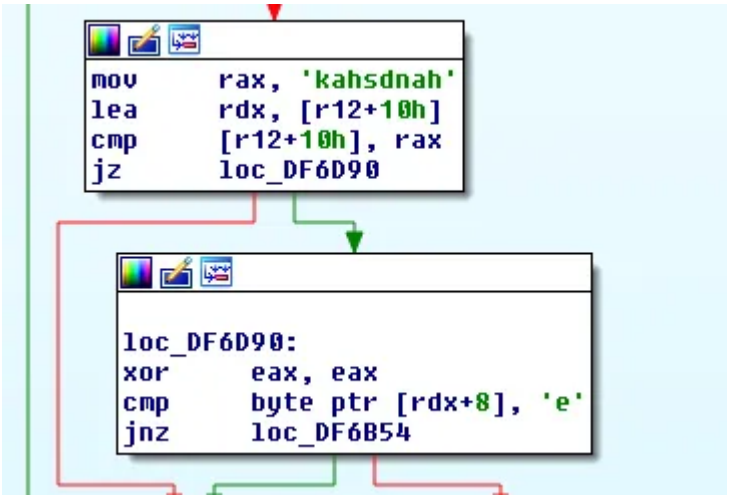

Image1: handshake

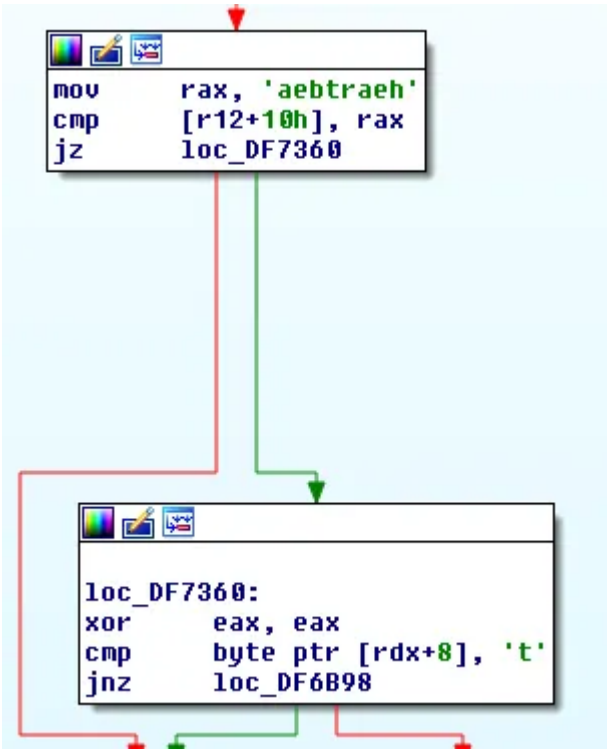

Image2: heartbeat

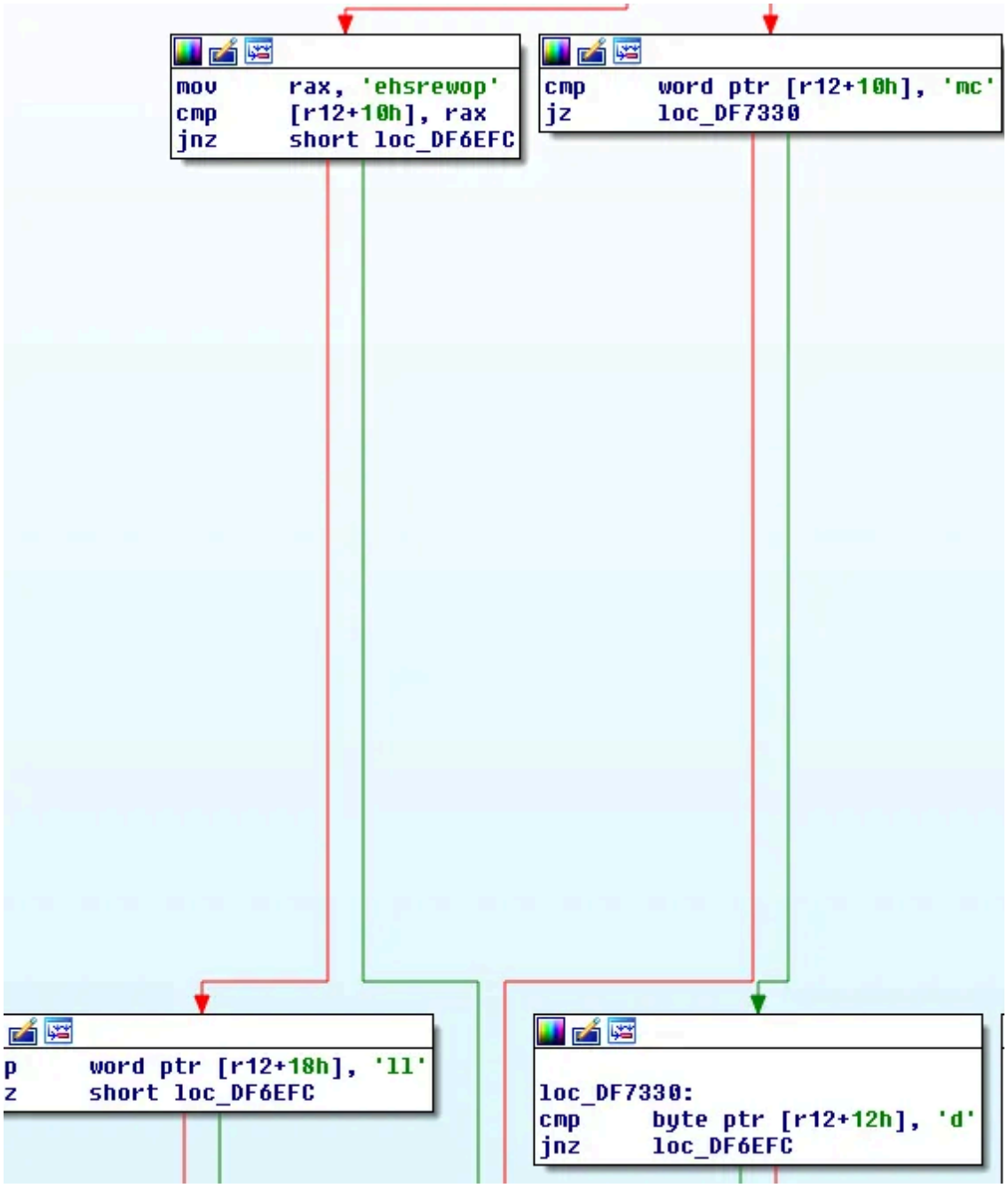

Image3. powershell or cmd

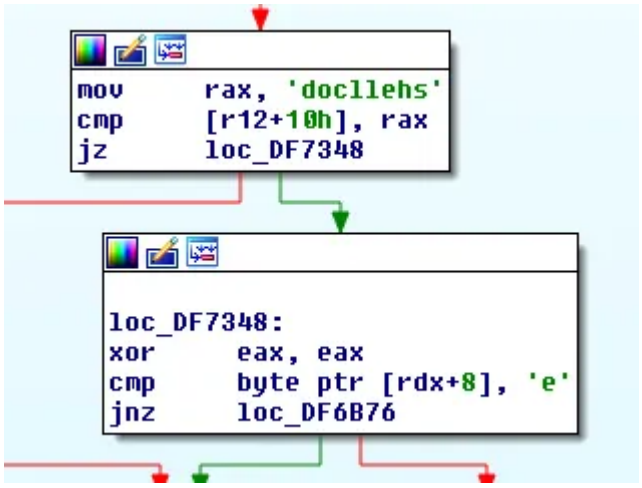

Image4: shellcode

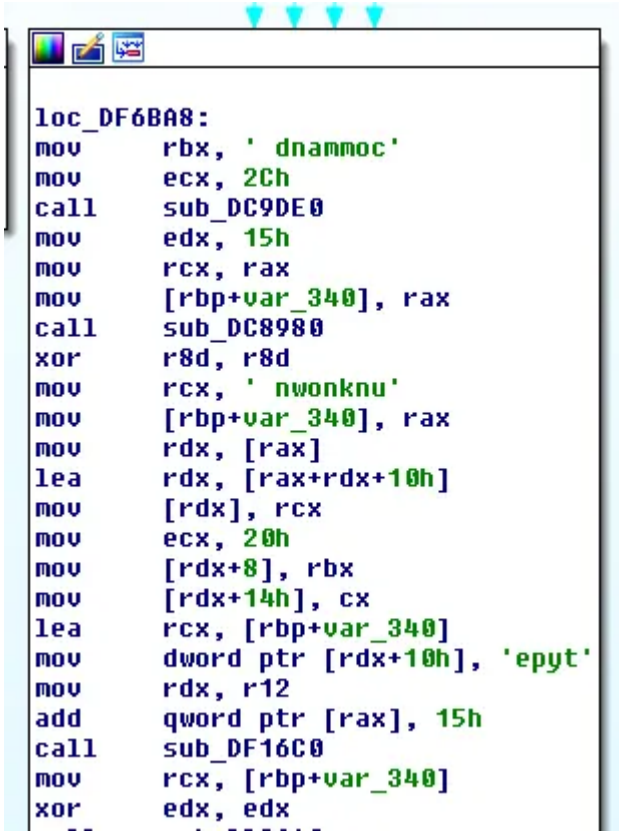

Image5: unknown command

While some of the strings are visible, a string encoding and decoding routine exists.

Image6: Decoding routine

The decoding loop XORs every byte of the encoded string with every byte of the key and then increments the key by one, due to how XOR works this is the same as XOR encoding every byte with a single XOR byte and

incrementing the key by the iterator in a rolling single byte XOR loop.

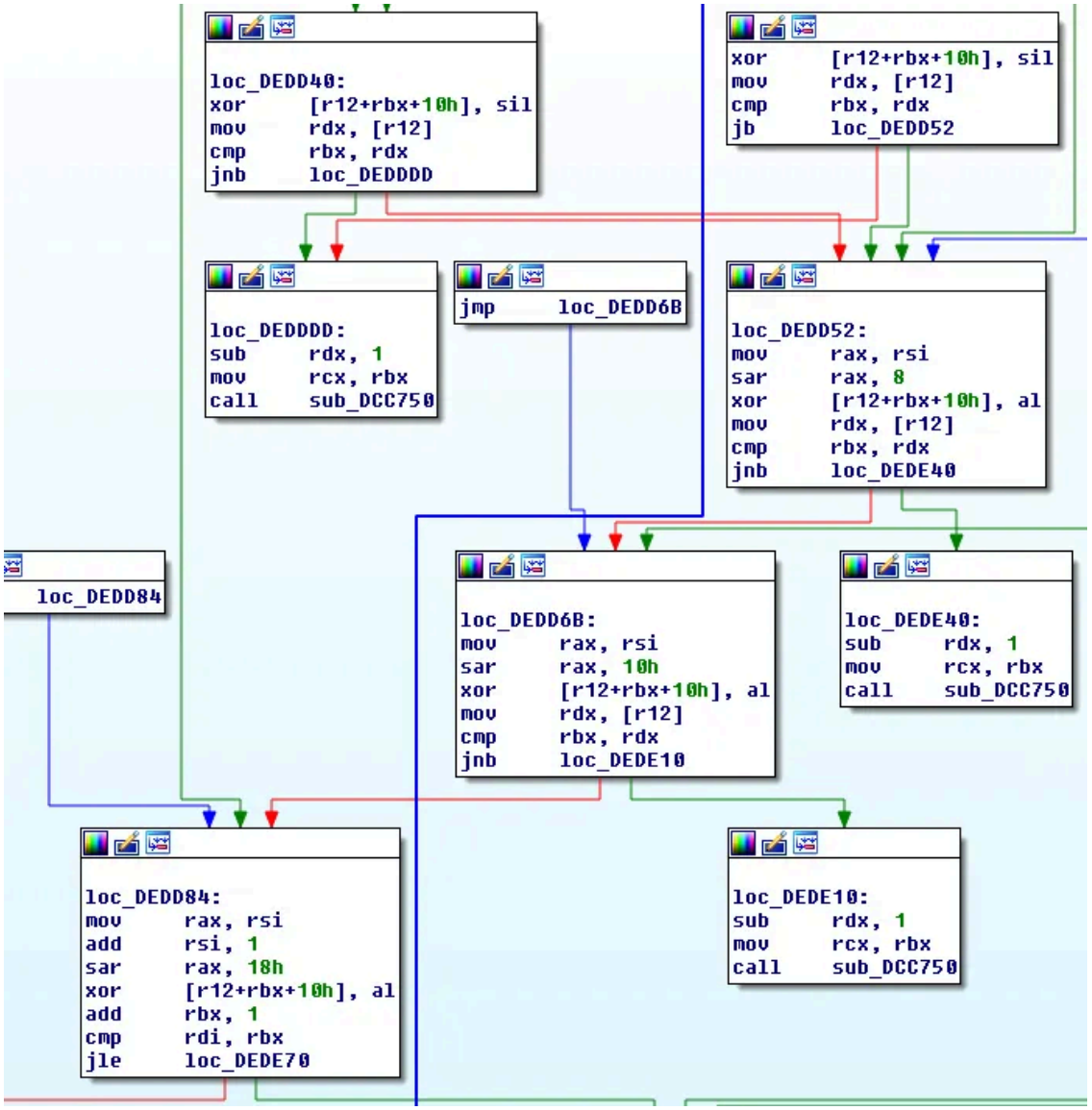

Image7: Decoding Loop

After running the routine the following strings are decrypted:

Press enter or click to view image in full size

```
path_adj
path_noun
seq_num
seq_total
job_id_header
user_agent
server_public_key
https://centralbancshares.com
https://centralbancshares.com
https://gariloy.com
https://liqui-technik.com
WW+f7o2MTjOX6i7ydNk8DA==
EVks0wzksyQGHwnYJ7nL2Q==
cUpEfOPbUtJduS2Rk0YDY0W9smZeNkH81j7osAJVh0c=
e8cbd40fda2500cd496b55df43402d8ed077b8cd965701a205c17f2b0389fce1
pp3Sn8eVj2M87F+N6bOqlg==
EVks0wzksyQGHwnYJ7nL2Q==
cUpEfOPbUtJduS2Rk0YDY0W9smZeNkH81j7osAJVh0c=
about
JSESSIONID
APISID
Cookie:
SID
```

As referenced earlier, the sample functions a bit like a loader, with the ability to process shellcode and execute tasks or jobs via cmd or powershell. Similar to the majority of Baza campaigns, the tasks that were executed came in the form of cobalt strike stagers. The benefit of using this “fileless” mechanism, is nothing touches disk.

Upon execution, a callout is made by the stager to `topservicebin[.]com`. During the campaign, the domain resolved to the ip `45.141.87 .41` which contained the following SSL certificate.

SSL Certificate

```
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 468210062 (0x1be8518e)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: C=US, ST=TX, L=Texas, O=lol, OU=, CN=topservicebin.com
        Validity
            Not Before: Feb  1 18:15:51 2021 GMT
            Not After : Feb  1 18:15:51 2022 GMT
        Subject: C=US, ST=TX, L=Texas, O=lol, OU=, CN=topservicebin.com
        Subject Public Key Info:
            Public Key Algorithm: rsaEncryption
                Public-Key: (2048 bit)
```

Image8: SSL Cert

The following are all Cobalt Strike servers with matching or similar certs, sharing the same hosting provider.

Host	Domain
45.141.87.2	bestserviceupdate.com
45.141.87.3	newserviceboost.com
45.141.87.4	newservicemonster.com
194.26.29.241	two-bytes.pro
45.141.87.41	topservicebin.com
45.141.84.65	topbackupupd.com
45.141.84.85	rargame.com
45.141.84.84	top-serviceupdate.com
45.141.87.42	topserviceupd.com
45.141.84.63	backupupdonline.com
194.26.29.240	backup-supp.com
194.26.29.253	bestserviceboost.com
194.26.29.247	backupupd.com
45.141.84.34	serviceboulder.com
194.26.29.171	first-makler.pro
194.26.29.254	bestservicehelp.com
45.141.87.36	service1elevate.com

Nearly identical infrastructure has been positively associated with the deployment of not only BazarLoader but with the deployment of RYUK by Mandiant last October. [6]

Nimar Loader appears to be partially based on existing code or perhaps the idea originated elsewhere. The actors behind TrickBot have incorporated free utilities and software developed by the CyberSecurity or opensource communities for their own nefarious purposes (MiniLZO, BloodHound, CobaltStrike, PowerSploit, Obfuscator-LLVM, ADVobfuscator…) in the past. While the nim language is not new to the offensive scene, it is quite a departure for the traditional tooling of Trickbot. [7]

References:

1. https://twitter.com/pancak3lullz/status/1252303608747565057
2. https://blog.fox-it.com/2020/06/02/in-depth-analysis-of-the-new-team9-malware-family/
3. https://www.cybereason.com/blog/a-bazar-of-tricks-following-team9s-development-cycles
4. https://twitter.com/James_inthe_box/status/1357009652857196546
5. https://twitter.com/Casperinous/status/1357013722955399170
6. https://www.fireeye.com/blog/threat-research/2020/10/kegtap-and-singlemalt-with-a-ransomware-chaser.html
7. https://github.com/byt3bl33d3r/OffensiveNim
