Ursnif beacon decryptor

github.com/mlodic/ursnif_beacon_decryptor

mlodic

mlodic/
ursnif_beacon_decryptor

Ursnif beacon decryptor

1 Contributor | 1 Issue | 26 Stars | 3 Forks

Ursnif v.3 aka Dreambot/Gozi/ISFB

This is a simple script that could be used to:

- check if a suspicious URL is correlated to Ursnif activity
- decrypt the URL check-in on the fly to get the data sent to the C2 server

Usage:

```
python3 ursnif_beacon_decryptor.py -u <url> -k <key>
```

A different version inserts the encrypted path as data of a POST request.

In case, use the option `-o` instead of `-u`

Example:

Input:

```
python3 ursnif_beacon_decryptor.py -u
"http://qjdyugisselle.club/images/NM_2Ff8mqmMQjmr/c842xf8TIJp_2FlmC5/Ulz244kFh/KMjQpHv
 -k "10291029JSJUYNHG"
```

Output:

```
[2019-04-15 11:24:25 - INFO] c2 domain: 'qjdyugisselle.club'
[2019-04-15 11:24:25 - INFO] path to analyze:
/images/NM_2Ff8mqmMQjmr/c842xf8TIJp_2FlmC5/Ulz244kFh/KMjQpHVvOnBhk6eOvBBW/R_2FCf2Bk9wZ

[2019-04-15 11:24:25 - INFO] Congrats! decoded data:
fjidtflrb=bdaxhhfg&soft=3&version=217173&user=a618b5f78c4ff30be60d08c7ba561278&server=
```

Params:

- fjidtflrb -> junk param, always present at the start of the uri to generate randomness (and always different)
- soft -> major version
- version -> minor version
- user -> unique user id
- server -> unique c2 server id
- id -> bot group id
- crc -> payload to retrieve (1-DLL32b, 2-DLL64b, 3-ps1)
- uptime -> time elapsed from initial infection (seconds)

Different versions could have more parametres. Example: `hash time action os system tor`

Requirements

- Python3
- cryptopp library (libcrypto++6 debian repo)

If you run a Linux environment, you can just run the python script that will load `Driver.so` that is needed to perform decryption.

If you run a Windows OS, you should re-compile `Driver.cpp` and change the library loaded with the new one.

Additional info

The script needs the key that the malware uses for encryption.

Some reverse engineering is required to get that info.

However, observations led to the fact that the key is usually shared among a lot of samples and rarely changed.

If you don't have one, you could just run the script that would try our predefined keys that we saw the malware used in the wild

Welcome to:

- tips on observed different behaviours of the malware
- decryption for other phases of the communication with the C2 infrastructure
- everything that can help to fight this threat

Twitter: https://twitter.com/matte_lodi