PhotoLoader ICEDID

research.openanalysis.net/icedid/bokbot/photoloader/config/2023/04/06/photoloader.html

OALABS Research April 6, 2023

Overview

Photoloader is the initial loader stage used to load ICEDID,

ICEDID was originally used for banking credential theft with a later pivot as a reconnaissance tool for pre-ransomware intrusions. The webinjects used for credential theft are still active though this malware is most often associated with ransomware incidents.

According to Proofpoint there is a fork of ICEDID that does not have webinject capability and is possibly developed by three separate actors...

> **Standard IcedID Variant** – The variant most commonly observed in the threat landscape and used by a variety of threat actors.
>
> **Lite IcedID Variant** – New variant observed as a follow-on payload in November Emotet infections that does not exfiltrate host data in the loader checkin and a bot with minimal functionality.
>
> **Forked IcedID Variant** – New variant observed by Proofpoint researchers in February 2023 used by a small number of threat actors which also delivers the bot with minimal functionality.

References

- DFIRReport:ICEDID -> Quantum ransomware- ICEDIDs network infrastructure is alive and well
- ICEDID Configuration Extractor
- Fork in the Ice: The New Era of IcedID
- icedid_peloader.py
- New version of IcedID Trojan uses steganographic payloads

Samples

- new loader sample (dfir report)`2db4fadfb2565fd9474e4d5303f953e96ac248de3267014c32e8a669e7e600e0` UnpacMe
- older sample `963397cec08790b25ff273cbe4b133634ae045d5ff8a4492e6f585f2ad14db65`UnpacMe

- old unpacked 32bit photoloader `1b01700425c30c2c498718966aee96cfdebacc2f6167576f7aa56e3f43ec3282` malpedia

Analysis

It looks like the Malpedia `photoloader` yara rules are a bit too loose and match the newer "gzip" variant of the loader. The config location/encryption is different between these two loaders and `photoloader` has not been used in a few years. We are going to create a new rule that will be used to only match the newer variants.

Rule

This rule is **heavily** influenced by the elastic rules in their config extractor

```
rule icedid_loader {
    strings:
        $a1 = "; _gat=" wide fullword
        $a2 = "; _ga=" wide fullword
        $a3 = "; _u=" wide fullword
        $a4 = "; __io=" wide fullword
        $a5 = "; _gid=" wide fullword
        $a6 = "loader_dll_64.dll" ascii fullword
        $config_decryption1 = {45 33 C0 4C 8D 0D ?? ?? ?? ?? 49 2B C9 4B 8D 14 08 49
FF C0 8A 42 ?? 32 02 88 44 11 ?? 49 83 F8 }
        $config_decryption2 = { 00 42 8A 44 01 ?? 42 32 04 01 88 44 0D ?? 48 FF C1 48
83 F9 }
        condition:
            filesize < 60000 and
            (
                (3 of ($a*) and $config_decryption1) or
                $config_decryption2
            )
}
```

Config Extractor

This is a modified version of the elastic config extractor

```
import pefile
import re
import struct

file_data =
open('/tmp/samples/963397cec08790b25ff273cbe4b133634ae045d5ff8a4492e6f585f2ad14db65',
'rb').read()
pe = pefile.PE(data = file_data)
```

```
IMAGE_SCN_CNT_CODE = 0x00000020

def xor(data, key):
    out = []
    for i in range(len(data)):
        out.append(data[i] ^ key[i % len(key)])
    return bytes(out)

def is_ascii(s):
    return all((c < 128 and c > 39) or c == 0 for c in s)


key = None
domain = None
campaign_id = None


mapped = False
if pe.sections[0].get_data()[:100] == b'\x00'*100:
    print("Mapped!")
    mapped = True

for s in pe.sections:
    if (s.Characteristics & IMAGE_SCN_CNT_CODE) == 0:
        if mapped:
            section_data = file_data[s.VirtualAddress:s.VirtualAddress +256]
        else:
            section_data = s.get_data()
        if len(section_data) < 250:
            print("Section too small")
            continue
        # This is a hack to skip stuff that doesn't look like a key
        tmp_key = section_data[:32]
        if b'\x00'*10 in tmp_key:
            print("Too many nulls in key")
            continue
        data = section_data[64:96]
        tmp_config = xor(data, tmp_key)
        domain = None
        try:
            domains = tmp_config[4:]
            domains = domains.split(b"\x00")
            if not is_ascii(domains[0]):
                        continue
            domain = domains[0].decode("UTF-8")
        except:
            print("Domain decode error")
            continue
        if len(domain) < 5:
            print("Domain too small")
            continue
        # If we are here we have a config!
```

```
        campaign_id = struct.unpack('<I', tmp_config[:4])[0]
        key = tmp_key.hex()
        break

assert key is not None
assert domain  is not None
assert campaign_id is not None

config = {
            "campaign_id": campaign_id,
            "domains": domain,
            "key": key,
         }


print(config)

Mapped!
Too many nulls in key
{'campaign_id': 3581911946, 'domains': 'smockalifatori.com', 'key':
'5e845c90daccb7f15a824ddf17ebccb65094b386fa909bf6c802f42a295e5dc1'}
```

```
def is_ascii(s):
    return all((c < 128 and c > 39) or c == 0 for c in s)

def extract_config(file_path):
    file_data = open(file_path, 'rb').read()
    pe = pefile.PE(data = file_data)

    mapped = False
    if pe.sections[0].get_data()[:100] == b'\x00'*100:
        #print("Mapped!")
        mapped = True

    key = None
    domain = None
    campaign_id = None

    try:
        for s in pe.sections:
            if (s.Characteristics & IMAGE_SCN_CNT_CODE) == 0:
                if mapped:
                    section_data = file_data[s.VirtualAddress:s.VirtualAddress +256]
                else:
                    section_data = s.get_data()
                if len(section_data) < 250:
                    #print("Section too small")
                    continue
                # This is a hack to skip stuff that doesn't look like a key
                tmp_key = section_data[:32]
                if b'\x00'*10 in tmp_key:
                    #print("Too many nulls in key")
                    continue
                data = section_data[64:96]
                tmp_config = xor(data, tmp_key)
                domain = None
                try:
                    domains = tmp_config[4:]
                    domains = domains.split(b"\x00")
                    if not is_ascii(domains[0]):
                        continue
                    domain = domains[0].decode("UTF-8")
                except:
                    #print("Domain decode error")
                    continue
                if len(domain) < 5:
                    #print("Domain too small")
                    continue
                # If we are here we have a config!
                campaign_id = struct.unpack('<I', tmp_config[:4])[0]
                key = tmp_key.hex()
                break

        assert key is not None
```

```
        assert domain  is not None
        assert campaign_id is not None

        config = {
                    "campaign_id": campaign_id,
                    "domains": domain,
                    "key": key,
                }
    except:
        return {}
    return config



# import required module
import os
# assign directory
directory = '/tmp/samples/'

# iterate over files in
# that directory
for filename in os.listdir(directory):
    f = os.path.join(directory, filename)
    # checking if it is a file
    if os.path.isfile(f):
        print(f)
        config = extract_config(f)
        print(config)
```

```
/tmp/samples/884cdf248d0235d77adc1d88603d460d64c88c517d5e571b75749be42364d6a8
{'campaign_id': 3248465841, 'domains': 'qsertopinajil.com', 'key':
'e999037e2b4084f0c1284ac991eca172030d99a0fd13ad6061af36c0d26bd1c0'}
/tmp/samples/a2158fb6574d9d8f473eee19b6cba91f2d5c0fc5289e9245bb4d290380e22226
{'campaign_id': 2615141838, 'domains': 'olifamagaznov.com', 'key':
'287130e4e0b4080bf367a2a3aaede31b5d652977f5e8a702b1c25b1afc7c2368'}
/tmp/samples/056de2c7a57fb4022d19398c6fc2676565afcda5e1f05ba0d91e58284e36d682
{'campaign_id': 133894510, 'domains': 'restorahlith.com', 'key':
'aa677c588cb603932a4965e66ca04fbfb85784e26eb18cb73555537c554a0568'}
/tmp/samples/f1a6325da85adcae0b21cd02592dfc4747fbe5b4eb428dd515000e35cc4e7f47
{'campaign_id': 3278418257, 'domains': 'ariopolanetyoa.com', 'key':
'7bbf03f22f1a464224f55dd9c01de97d218f26e8058e48b749ba1892293b99b1'}
/tmp/samples/4cda20be09dd99ab0ccf618a6c1c62122f0c484fb5925031b6ab3e7ce2f016ae
{}
/tmp/samples/98984bc4bd9af65911d9102bde5cae341ffb9bcc913aca1fe69093466176a058
{'campaign_id': 3248465841, 'domains': 'qsertopinajil.com', 'key':
'e999037e2b4084f0c1284ac991eca172030d99a0fd13ad6061af36c0d26bd1c0'}
/tmp/samples/6391cf53ec2894915b2fe913913212ee074b28e8c52b2cd1039a6a1ba0b7efc0
{}
/tmp/samples/963397cec08790b25ff273cbe4b133634ae045d5ff8a4492e6f585f2ad14db65_rebase.e

{'campaign_id': 3581911946, 'domains': 'smockalifatori.com', 'key':
'5e845c90daccb7f15a824ddf17ebccb65094b386fa909bf6c802f42a295e5dc1'}
/tmp/samples/963397cec08790b25ff273cbe4b133634ae045d5ff8a4492e6f585f2ad14db65
{'campaign_id': 3581911946, 'domains': 'smockalifatori.com', 'key':
'5e845c90daccb7f15a824ddf17ebccb65094b386fa909bf6c802f42a295e5dc1'}
/tmp/samples/e6648ec933fc75d78621f4be11b2391949c65e1822316a4d56d5ebd9a008a544
{'campaign_id': 1228806356, 'domains': 'klepdrafooip.com', 'key':
'c7876ea3c330edfc2acf9d6ba61cccf08411ff1cbf69783738b072af97fe35ee'}
/tmp/samples/866011c18e7db2ad7203600a59f050654787aef69741b718a8f6d938260245b5
{'campaign_id': 1348756909, 'domains': 'utorsabegot.com', 'key':
'5abb51b1624cf8d72e7396508290289506e3ae86106a05777692136daa1fb254'}
/tmp/samples/2c41811701b287107fe0e0218652eba7621988b9ee7aa1090d34679d7c62d4d0
{'campaign_id': 2076641214, 'domains': 'alishabrindeader.com', 'key':
'7eca10dee216a6911fc9010b144bd46467042d0e5a8d9529ab0d3f09c94555f0'}
/tmp/samples/e0c14b14c0715225f710b77ed50bba269549d7bde9bf058fc30e47fe8bbe2a85
{'campaign_id': 3954321778, 'domains': 'ehonlionetodo.com', 'key':
'ffe77b49c0284a4eb902e8dd15d56c910e58c78664408be3b255ff13c1c322d9'}
/tmp/samples/1df8dce6c6807fa52d3e655d3d29cdceff3b4e7f5c2354947e973f8174439639
{'campaign_id': 3324185820, 'domains': 'druidfenixis.com', 'key':
'03b0f4ecb82527644b80fbe33f3c3c686d75a37567b5742f51ae69e0cf6b2815'}
/tmp/samples/244781cba0c9e5cc3763a9a8b450d0b567698ff76541bad100da2f985f115376
{'campaign_id': 1228806356, 'domains': 'klepdrafooip.com', 'key':
'c7876ea3c330edfc2acf9d6ba61cccf08411ff1cbf69783738b072af97fe35ee'}
/tmp/samples/20d215358269e71b4f37714b209c0b2042cd2275f75ba95cc3918aabf4b004a6
{'campaign_id': 2492795688, 'domains': 'greenfairsaid.com', 'key':
'02d91b9c520f7a80db3eaf20bffec772ea2de8c18d6d77b42ea46da40f5f5ebb'}
/tmp/samples/5179c4c9efedcb05458b624397c957608a92a4327860e913cc82fe64c5e8012e
{'campaign_id': 3131022508, 'domains': 'wagringamuk.com', 'key':
'ade2a8e94a5e9d004347aca5b19c7a2a07678a91fb8ed8bdad397354a999270b'}
/tmp/samples/966c0e03cf32173367ef2249877ccb6f9b8b549534841e435e6f2aebb047c8c8
{'campaign_id': 3195585424, 'domains': 'afrodizajoy.com', 'key':
```

```
'8c0d72d85b4e23a9b62f9a0b7011c948ee20fdc7bdf3a62ef078698af877128d'}
/tmp/samples/1cada6c3166a8db10461cc53ac55985d646422a4c69665c2b6952719b4fc4a7f
{'campaign_id': 2611621973, 'domains': 'aproillionsgif.com', 'key':
'60706bb5637cf64d92711de022fdb2c6950b2f2eeedbbab35720eeadd3e9113e'}
/tmp/samples/17edc5bee30e951fb612f78902153189a355854aa54a2d46f14665e7dcf542cd
{'campaign_id': 176945684, 'domains': 'ilioskajyzi.com', 'key':
'a0298811c76505a04a34d2e43cdd95a6d0c8f5ba6276987c1f95be052cf72dea'}
/tmp/samples/e52d056d08d9eb9b4683e6bead0a87d13009e2602f829c4a137722e660d5397e
{'campaign_id': 109932505, 'domains': 'ilekvoyn.com', 'key':
'ae5c00d7b56528d2c82705a49c6511673553142bfc62fd816d35e4dd95364617'}
/tmp/samples/a4934907f560be76607b543bbbe18eb143a10e810d07ccf39a22717967f22da8
{'campaign_id': 2611621973, 'domains': 'aproillionsgif.com', 'key':
'60706bb5637cf64d92711de022fdb2c6950b2f2eeedbbab35720eeadd3e9113e'}
/tmp/samples/2db4fadfb2565fd9474e4d5303f953e96ac248de3267014c32e8a669e7e600e0
{'campaign_id': 2220668032, 'domains': 'alockajilly.com', 'key':
'9c721d32ebaef948d97b5c598d5d245b01f1da2eba9e034139344192534bd8a6'}
/tmp/samples/93fae86b3d23aeee26731a04ccac78b127000cd6d51c2f5096fb230996c8a910
{'campaign_id': 1392658338, 'domains': 'nrncipalmoonw.com', 'key':
'80b33f829e0b667861e6e5efac3825b7751a6e27e326796e62c8f4147176e064'}
/tmp/samples/c139cc51b09e6a0c0415b665c3cec704299cca185f5f7c9e239bdf82e3911245
{'campaign_id': 1732687004, 'domains': 'keepfootbal.com', 'key':
'9c8f084e492b44179c83e50fb2ec25d644429ea9f67060fa0d4962c3a85c3808'}
/tmp/samples/b44eb70fc9830829c73d92b510756df55e0fe0f1be15ec6a12f18517a3147603
{'campaign_id': 2546188793, 'domains': 'anisiderblomm.com', 'key':
'40811aaff35e3c0330a698ed3c0b353e1395ae6301b3edaf4a99a09953580fef'}
```