National Cyber Security Centre
a part of GCHQ

Turla group update Neuron malware

Version 1.0

Reference: NCSC-Ops/04-18

18 January 2018

About this Document

This NCSC report provides new intelligence on the Neuron malware, a tool used by the Turla group to target the UK. It contains IOCs and signatures for detection and network monitoring.

Handling of the Report

Information in this report has been given a Traffic Light Protocol (TLP) of WHITE, which means it can be shared within and beyond the CiSP community with no handling restrictions.

Disclaimer

This report draws on reported information, as well as information derived from industry sources.

Contents

Introduction

In November 2017, the NCSC released an advisory highlighting the Turla Group's use of the tools Neuron and Nautilus.[1]

Since then, the NCSC has identified a new version of the Neuron malware. The new version has been modified to evade previous detection methods.

Neuron operates on Microsoft Windows platforms, primarily targeting mail servers and web servers. The NCSC has observed this tool being used by the Turla group to maintain persistent network access and to conduct network operations.

The compile times contained within these new binaries show that the actor implemented the required modifications to Neuron approximately five days after public releases by the NCSC and other vendors.

This NCSC report provides new intelligence on the Neuron malware, a tool used by the Turla group to target the UK. It contains IOCs and signatures for to be used for network monitoring and detection.

The files analysed in this report are available on VirusTotal.

Summary of changes to Neuron malware

- The .NET payload is loaded in-memory as opposed to being dropped to disk;
- Communications have been modified to avoid detection;
- Some encryption methods have replaced RC4 with AES;
- The modifications are sufficient to avoid previously released signatures & IOCs.

[1] https://share.cisp.org.uk/docs/DOC-6912

Neuron Updates

A sample of Neuron was recently uploaded to VirusTotal. This sample appears to be an updated version of Neuron. Changes have primarily been made to the dropper and loading mechanisms.

The PDB string embedded within the binary supports the assumption that this is a newer version by referring to itself as “neuron2”.

```
D:\Develop\sps\neuron2\x64\Release\dcomnet.pdb
```

This sample contains sufficient modifications to frustrate detection, allowing Turla operations to continue.

Loader

With previous versions of Neuron, a native dropper was utilised to write the main payload to disk, establish persistence and ensure execution. This latest version uses a native x64 loader to execute the .NET payload in-memory. The payload is encrypted within the loader, which ensures the payload never touches disk in plaintext. This modification has likely been made to evade detection during disk scans performed by anti-virus products, however anti-virus products that scan memory will still likely be able to detect the payload running.

The loader has the required exports to enable the configuration as a service, therefore it's believed this will be the method used for persistence.

The loader can also specify which endpoints (HTTP(S) or pipe) to listen on by passing them to the .NET executable as arguments. In this sample the endpoints specified are different to previous versions:

- http://*:80/OWA/OAB/
- https://*:443/OWA/OAB/

If no arguments are provided the payload will use the following defaults for HTTP(S) or pipes:

- http://*:80/W3SVC/
- https://*:443/W3SVC/
- pipe://*/Winsock2/baseapi_http

Error handling has been added to the new payload. If the webserver encounters an exception it will attempt to use the default values above, if another exception occurs then the payload will revert to using the default HTTP (port 80) value.

Payload

The main payload is still a .NET executable, but several modifications have been made to its operation which are described below

Encryption

Previous versions of Neuron used RC4 for the encryption of data stored on disk or sent over the network. Portions of the updated Neuron service have been migrated to AES, however, some components still rely on the RC4 implementation, such as encrypting command information.

The actors have configured multiple hardcoded encryption keys rather than using one for everything. For example, one is used for normal communication between nodes, and another is used if the node is proxying a request. These modifications are likely implemented to make detection and decryption by network defenders more difficult.

Communications

The communication between clients and servers has also changed to avoid detection. The server expects a POST request, but rather than using the previous pre-defined parameter names (cid, cadata etc.), the new function loops through each parameter looking for certain characters within that parameter's value to determine what functionality should be performed. This will allow the parameter names to be randomly generated and/or regularly changed, making it more difficult for network defenders to reliably detect communications.

As an example, the following characters are looked for (in the order shown) to determine which functionality should be performed:

Character	**Functionality**
#	Set the AES salt
(and)	Return list of storage files
(	Get and return defined storage file
)	Add specified storage file to local storage (write to disk)
-	Send RSA encrypted encryption key (machine GUID)
_	Proxy request through to another address
, but not _	Perform specified command and return result

Associated Files

Name	dcomnet.dll
Description	Neuron2 Loader (x64)
MD5	60bcc6bc746078d81a9cd15cd4f199bb
SHA1	c9fc7ce10aba20894ef914d2073021a48995db17
SHA256	51616b207fde2ff1360a1364ff58270e0d46cf87a4c0c21b374a834dd9676927
Size	170496
Compile Time	28 Nov 2017 06:25:24

Name	neuron2.exe
Description	Neuron2 Payload
MD5	d891c9374ccb2a4cae2274170e8644d8
SHA1	2fb145c64263006a95a0771b57e967977f63954d
SHA256	83d8922e7a8212f1a2a9015973e668d7999b90e7000c31f57be83803747df015
Size	59392
Compile Time	28 Nov 2017 04:44:26

Neuron Yara

```
rule neuron2_loader_strings {
  meta:
    description = "Rule for detection of Neuron2 based on strings within the loader"
    author = "NCSC"
    hash = "51616b207fde2ff1360a1364ff58270e0d46cf87a4c0c21b374a834dd9676927"
  strings:
    $ = "dcom_api" ascii
    $ = "http://*:80/OWA/OAB/" ascii
    $ = "https://*:443/OWA/OAB/" ascii
    $ = "dcomnetsrv.cpp" wide
    $ = "dcomnet.dll" ascii
    $ = "D:\\Develop\\sps\\neuron2\\x64\\Release\\dcomnet.pdb" ascii
  condition:
    (uint16(0) == 0x5A4D and uint16(uint32(0x3c)) == 0x4550) and 2 of them
}
```

```
rule neuron2_decryption_routine {
  meta:
    description = "Rule for detection of Neuron2 based on the routine used to decrypt the
payload"
    author = "NCSC"
    hash = "51616b207fde2ff1360a1364ff58270e0d46cf87a4c0c21b374a834dd9676927"
  strings:
    $ = {81 FA FF 00 00 00 0F B6 C2 0F 46 C2 0F B6 0C 04 48 03 CF 0F B6 D1 8A 0C 14 8D 50
01 43 32 0C 13 41 88 0A 49 FF C2 49 83 E9 01}
  condition:
    (uint16(0) == 0x5A4D and uint16(uint32(0x3c)) == 0x4550) and all of them
}
```

```
rule neuron2_dotnet_strings {
  meta:
    description = "Rule for detection of the .NET payload for Neuron2 based on strings
used"
    author = "NCSC"
    hash = "83d8922e7a8212f1a2a9015973e668d7999b90e7000c31f57be83803747df015"
  strings:
    $dotnetMagic = "BSJB" ascii
    $s1 = "http://*:80/W3SVC/" wide
    $s2 = "https://*:443/W3SVC/" wide
    $s3 = "neuron2.exe" ascii
    $s4 = "D:\\Develop\\sps\\neuron2\\neuron2\\obj\\Release\\neuron2.pdb" ascii
  condition:
    (uint16(0) == 0x5A4D and uint16(uint32(0x3c)) == 0x4550) and $dotnetMagic and 2 of
($s*)
}
```