

# Analysis of THREATNEEDLE C&C Communication (feat. Google TAG Warning to Researchers)

---

 [medium.com/s2wlab/analysis-of-threatneedle-c-c-communication-feat-google-tag-warning-to-researchers-782aa51cf74](https://medium.com/s2wlab/analysis-of-threatneedle-c-c-communication-feat-google-tag-warning-to-researchers-782aa51cf74)

S2W

January 29, 2021



S2W

Jan 27, 2021

.

4 min read

**Author:** [hypo](#) (Sojun Ryu)

**Modified History**

- 01/28/2021 **IoC Updated**

- 01/28/2021 **Report Updated**



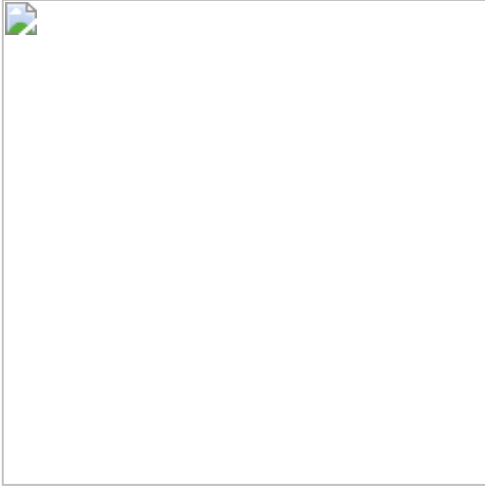
Malware mentioned in “[North Korean hackers have targeted security researchers via social media report](#)” published by Google Threat Analysis Group (TAG) is considered to be a **ThreatNeedle** which is dubbed by Kaspersky. We already disclosed the deep analysis regarding C2 communication of ThreatNeedle at **DCC 2019** and **Kaspersky SAS Lightning Talk 2019**.

In addition, the malware and C2 communication have in common with [Operation MalBus](#).

#### **Additional Reference for Operation MalBus**

> [MalBus Actor Changed Market from Google Play to ONE Store](#)

We briefly delivers only the essential fact in Medium, and for other details, please refer to the **attached PDF file** which is presented at DCC and Kaspersky SAS.



Below is tweet from [Seongsu Park](#) of [Kaspersky](#) GReAT team stating that this malware is ThreatNeedle.

ThreatNeedle is already known that it has been used by the Lazarus group along with [Manuscript](#) from the past. Most of them operate through HTTP communication, and C&C servers are written in languages such as ASP and JSP. ThreatNeedle downloaded during the infection process receives commands from the C&C server and performs malicious actions, and the overall process is as follows.



The overall process of the ThreatNeedle  
The definition of each term is as below.

**Troy** → ThreatNeedle malware installed on Victim

**Proxy** → Control Page of C&C Server (.ASP, .JSP)

**favicon.icon** → Store IDs of Infected devices (only Troy's ID)

**build.xml** → Store Time, IP, ID of all authenticated connections (Troy and Manager)

**desktops.inf** → Store URL of MID server

**[TroyID]** → Identification value for each infected devices randomly generated when infected with ThreatNeedle

**[TroyID].jpg** → Store Victim who has [TroyID] information or command result

**[TroyID].bmp** → Store Attacker's command

## Troy(ThreatNeedle) → Proxy(C&C Page)

---



### List of Troy→Proxy Server Commands

The infected device information is transmitted to Proxy server and waits until specific command is received. Communication with the proxy is possible only when a specific type of parameter is transmitted, and Troy's connection is processed by the proxy's **handleTroy function**. The action for each commands are as follows.

**Query Structure:** [Field 1]=[Cmd]&[Field 2]=?&[Field 3]=[TroyID]&[Field 4]=[Packet ID]&[Field 5]=[Buffer Size]&[Field 6]=[Buffer]

, , and are actually used, but the 2nd and 4th field are not in use.

Proxy initially receives the command 71 from Troy and saves the information of infected devices as **[TroyID].jpg** file on the Proxy server. After checking the connection with the MID by using the command 81, the ID of the infected device, that is, Troy ID, is saved in **favicon.icon\***. After that, if there is a command file called **[TroyID].bmp**, it is read and transmitted to Troy. After the initial authentication is completed, the attacker's remote control command is stored in **[TroyID].bmp** encoded with **base64^0xA4**, and the result of executing the command is stored in the same encoding method in **[TroyID].jpg**. Files used in this process are immediately deleted.

| **\*favicon.icon** → store IDs of Infected devices

## Proxy → MID

---

When 'Troy' connects to the 'Proxy', and the command 81 is executed, Proxy sends its information to MID. An 'attacker' is capable of identifying the logs stored on MID server in order to identify the 'Proxy' servers that are connected with 'Troy.' In order to communicate with 'MID', specific command value and Proxy ID authentication from **[Figure 1]** should be executed. Commands are sent and received only when the MID server is active.

**Query Structure:** id=[CMD]&field=[0 or ProxyURL]&buffer=[ProxyID]&iframe=[1 or 0]&frame=[0 or 1][&uframe=3]

There are and for . 111 is used to update MID server and 112 is used when Troy is connected.

## Manager → Proxy

---

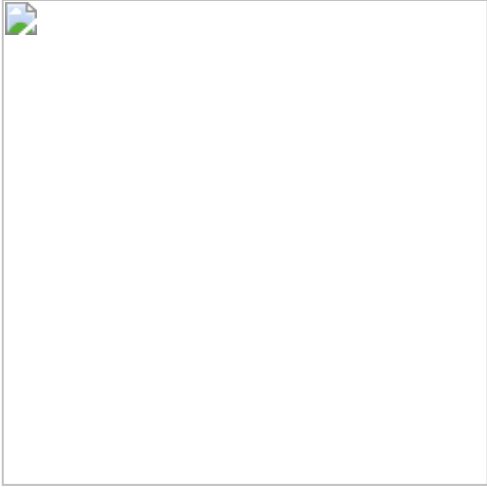


#### List of Manager→Proxy Server Commands

The attacker sends commands to 'Troy' through the 'Manager'. Adversary is capable of handling C&C infrastructure such as collecting the stored log file on Proxy server and updating MID server.

**Query Structure:** [Field 1]=[Cmd]&[Field 2]=[Target TroyID]&[Field 3]=[TroyID]&[Field 4]=[Additional Information]&[Field 5]=[Buffer Size]&[Field 6]=[Buffer]

Cmd, TroyID, Buffer Size and Buffer are always used, but the 2nd/4th fields are used only for the certain command.



## Indicators of Compromise

---

1a327cced0b0c0bf99146f276fb7a93148cd9a396ef06c73ab069365d079c869 | w3wp.exe  
3fd610f69ef1808431b090c40a065621d15f591bbf2470cd8a14f1ae352b6c2f | smss.exe

9f5e407601032063e1f1d263e9a2b11c99fbf094e2a0fe65bfa5ad72716cdbd8 | tab05\_b.mov

e0a62ba2c58b1a8e9484f1c4452aaafcab6a1ccfe44bfd680edbe859044049d2 | iisret.dll  
46196370d2cd24b19bd1272a9c3632e5ff9fbeb986960caa03b1e8186fb37239 | dll

07375a711dda055cfb8777d31aff9cfecb5f5142e88712cf93d41e2a317abe22 | ASP  
011cc019872f75c30cfa1d41201fc2341418bf53457449f8e066379e6df1ad12 | ASP  
cd4658151e41749ec71fe64d9e88b35fcd82afb8d3654bb6db9879bb4854d76a | ASP

## Related Work

---