When Threat Actors Fly Under the Radar: Vatet, PyXie and Defray777

unit42.paloaltonetworks.com/vatet-pyxie-defray777/4

Ryan Tracey, Drew Schmitt November 7, 2020

By Ryan Tracey and Drew Schmitt

November 6, 2020 at 6:15 PM

Category: Malware, Ransomware, Unit 42

Tags: Defray777, PyXie, Vatet

This post is also available in: 日本語 (Japanese)

Linking Vatet, PyXie and Defray777

While researching these malware families, we found that there were several consistencies between Vatet, PyXie and Defray777 that strongly suggest that all three malware families were created, and are currently maintained by, the same financially motivated threat group.

PDB Path Reuse

As we saw with the Defray777 decryptors, there are numerous victims that have been impacted by Defray777. However, these decryptors also show some overlap with PyXie. One of the decryptors we analyzed shares a common path with earlier versions of PyXie and its

Cobalt Mode downloader.

Defray777 Decryptor	Z:\coding\pyproject\compiled\ransom\ransom.pdb
PyXie	z:\coding\pyproject\python_static_2.7.15\
Cobalt Mode	Z:\coding\pyproject\compiled\cobalt_mode\cobalt_mode.pdb

Table 19. PDB paths shared between Defray777, PyXie and Cobalt Mode.

Additionally, some of the variants of Vatet we observed also have overlapping PDB paths.

Tetris	C:\Users\1\Downloads\tetris-game-master\Release\TetrisGame_zjy.pdb
Notepad	C:\Users\1\Downloads\notepad-master\Debug\notepad.pdb
Rainmeter	C:\Users\1\Downloads\rainmeter-master\x32-Release\Obj\Library\Rainmeter.pdb
Rainmeter	C:\Users\1\Downloads\rainmeter-master\x32-Release\Obj\Application\Rainmeter.pdb
Notepad++	C:\Users\1\Downloads\notepad-plus-plus-master\PowerEditor\bin\npp.pdb

Table 20. PDB paths shared between Vatet variants.

String Encryption

During our research, we observed that the method of string encryption in each of the variants was consistent. Defray777 uses the same string encryption that was used in PyXie. Additionally, the same string encryption methodology was observed in the Tetris variant of Vatet loader.

```
if ( v13 )
{
  v1 = 0;
  v13[52] = 0;
  do
  {
    v13[v1] = byte_41AB5C[v1] ^ (byte_41AB24[v1] + (v1 & 0x7F));
    byte_41AB27[v13 - byte_41AB26 + v1] = byte_41AB5D[v1] ^ (byte_41AB25[v1] + ((v1 + 1) & 0x7F));
    byte_41AB29[v13 - byte_41AB27 + v1] = byte_41AB5E[v1] ^ (byte_41AB26[v1] + ((v1 + 2) & 0x7F));
    v2 = byte_41AB27[v1] + ((v1 + 3) & 0x7F);
    v1 += 4;
    v13[v1 - 1] = byte_41AB5B[v1] ^ v2;
  }
  while ( v1 < 52 );
}
```

Figure 26. Defray777 string decryption example.

Creating Mutexes

Defray777 uses the same Mutex routine as the updated PyXie sample we analyzed, including the DEFAULTCOMPNAME fallback. One thing Defray777 does differently is that it omits the step where the computed MD5 hash is XOR'd with 0x2.

```
v1 = mw_get_data_length(*Computer_Name_or_DEFAULTCOMPNAME);// Gets the length of the ComputerName/default value
if ( mw_create_MD5(v2, v1, &rguid) )          // Generates MD5 hash
{
  LOBYTE(rguid.Data1) = 120;
  if ( StringFromGUID2(&rguid, sz, 260) )     // Converts GUID to string
  {
    CreateMutexW(0, 0, sz);                   // Generates mutex (derived from computer name)
```

Figure 27. The Defray777 mutex creation process.

Conclusion

Since 2018, a financially-motivated threat group has been using a *combination* of Vatet loader, PyXie RAT and Defray777 ransomware to target organizations in the healthcare, education, government and technology industries without drawing attention to themselves. They've only been a blip on the radar.

We have exposed this group's desire to use open source tools as a means for the Vatet loader. We have shown how PyXie is used to conduct reconnaissance and find and exfiltrate data. We have also uncovered how this group uses Cobalt Strike to deliver Defray777 into memory to encrypt files, causing catastrophic damage to their victims.

We hope that by shining more light on this group of threat actors, we can help disrupt their ability to conduct ransomware operations. Now that they are on the radar, we must aim to keep them there.

Palo Alto Networks customers are protected from this threat in the following ways:

- All samples in this report have a malicious verdict in WildFire.
- Cortex XDR detects these threats.
- Command and Control infrastructure has been classified as malicious in URL Filtering.

Continue reading: Indicators of Compromise