(https://www.welivesecurity.com/) (https://www.eset.com)

Registers as “Default Print Monitor”, but is a malicious downloader. Meet DePriMon

ESET researchers have discovered a new downloader with a novel, not previously seen in the wild installation technique

DePriMon is a malicious downloader, with several stages and using many non-traditional techniques. To achieve persistence, the malware registers a new local port monitor – a trick falling under the “Port Monitors (https://attack.mitre.org/techniques/T1013/)” technique in the MITRE ATT&CK knowledgebase (https://attack.mitre.org/). For that, the malware uses the “Windows Default Print Monitor” name; that’s why we have named it DePriMon. Due to its complexity and modular architecture, we consider it to be a framework.

According to our telemetry, DePriMon has been active since at least March 2017. DePriMon was detected in a private company, based in Central Europe, and at dozens of computers in the Middle East.

Some of the domain names used as C&C servers contain Arabic words, which gives an indication of a region-specific campaign. However, DePriMon deserves attention beyond its targets’ geographical distribution: it is carefully written malware, with lots of encryption that is used properly.

To help defenders stay safe from this threat, we've thoroughly analyzed this newly discovered malware, focusing on the downloader itself. Because we're missing initial stage(s), which we will refer to here as “the first stage”, we don't know the initial distribution and compromise vector. What kind of final payload is used in the attacks is another question that remains to be answered.

However, it should be noted that, in a few cases, DePriMon was detected with ColoredLambert malware on the same computers within a short time frame. ColoredLambert is used by the Lamberts (aka Longhorn) cyberespionage group and linked to the Vault 7 leak of CIA capabilities. Our colleagues from Symantec (https://www.symantec.com/connect/blogs/longhorn-tools-used-cyberespionage-group-linked-vault-7) and Kaspersky (https://securelist.com/unraveling-the-lamberts-toolkit/77990/) published their analyses in April 2017.

Technical analysis

Stage two

Both DePriMon's second and third stages are delivered to the victim's disk in the first stage. The second stage installs itself and loads the third stage using an encrypted, hardcoded path. One of the possible explanations is that it was configured after the first stage of the attack occurred.

The described installation technique is unique. In principle, it is described in the MITRE ATT&CK taxonomy as “Port Monitors”, under both Persistence and Privilege Escalation tactics. We believe DePriMon is the first example of malware using this technique ever publicly described.

The second stage registers the third-stage DLL as a port monitor by creating the following registry key and value:


```
HKLM\SYSTEM\CurrentControlSet\Control\Print\Monitors\Wi
Default Print Monitor
Driver = %PathToThirdStageDLL%
```

Administrator rights are required for creating this registry key.

At system startup, the registered DLL will be loaded by `spoolsv.exe` with `SYSTEM` privileges, which, combined with the uniqueness of this method, makes this technique very effective for attackers.

The second stage checks regularly whether there is a file in the `%system32%` folder with the same name as the third stage DLL file but without the “.dll” extension. This file serves as an uninstallation trigger – should DePriMon find it, it removes both this file and its own components in a secure way by overwriting the binaries and then deleting them.

Stage three

The third stage, responsible for downloading the main payload(s) from DePriMon’s operators, also implements some interesting techniques.

For C&C communication, it uses the Microsoft implementation of SSL/TLS, Secure Channel, instead of common APIs like `WinHTTP` or `WinInet`. Its configuration is very complex, as is the way the malware

handles it. Finally, the authors have put significant effort into encryption, making the DePriMon malware more difficult to analyze.

(https://www.welivesecurity.com/) (https://www.eset.com)

C&C communication

DePriMon communicates securely over TLS, however, not on a high level as is a typical scenario in malware. The connection is initialized with a Windows socket and can continue with initialization of an authenticated Security Support Provider Interface (SSPI) session with the Negotiate / NTLM SSP. After that, DePriMon uses Schannel.

SSPI is used/not used according to a particular flag in the configuration file and can utilize the local proxy settings of the machine. The implementation is similar to this example provided by Microsoft (https://docs.microsoft.com/en-us/windows/desktop/secauthn/using-sspi-with-a-windows-sockets-client).

The malware’s implementation of TLS via Schannel is similar to this example by Coast Research & Development (http://www.coastrd.com/c-schannel-smtp). It includes creating credentials, performing the client handshake and verifying the server certificate.

```
108           if ( !CreateCredentials(&establishConnection->phCreds) )
109           {
110             pCertContext = establishConnection->pCertContext;
111             establishConnection->pCertContext.dwCertEncodingType = 0;
112             v18 = establishConnection->socket;
113             establishConnection->field_224 = 1;
114             if ( !PerformClientHandshake(
115                     v18,
116                     &establishConnection->phCreds,
117                     &establishConnection->SchannelCredStruct,
118                     establishConnection->CnC,
119                     &establishConnection->phContext,
120                     &pExtraData) )
121             {
122               v19 = lpMem;
123               v20 = GetProcessHeap();
124               HeapFree(v20, 0, v19);
125               if ( !SSPIdispatchTable->sspicli_QueryContextAttributesW(
126                       &establishConnection->phContext,
127                       SECPKG_ATTR_REMOTE_CERT_CONTEXT,
128                       &establishConnection->pCertContext) )
129               {
130                 if ( !VerifyServerCertificate(*pCertContext, establishConnection->CnC) )
131                 {
132                   CertFreeCertificateContext(*pCertContext);
133                   *pCertContext = 0;
134                   goto LABEL_26;
135                 }
136                 goto LABEL_27;
137               }
138             }
139           }
```

(https://www.welivesecurity.com/wp-content/uploads/2019/11/Figure-1.png)

Figure 1. Part of the SSPI implementation as output by the Hex-Rays decompiler

After the communication is established, the third stage encrypts and decrypts messages manually each time.

Configuration

The configuration data for DePriMon’s third stage has 27 members, which is an unusually large number for a downloader. It is encrypted with AES-256 and embedded in the binary.

During the first run, DePriMon’s third stage (the downloader itself) decrypts the configuration data with Key 2 (see the IoCs section), encrypts it with Key 3 and stores the encrypted configuration file in a temporary folder. The filename for the configuration file is created via the following process: Starting with the second byte, the value of Key 2

is transformed into a number in base 36 but encoded using custom alphabet “abc...xyz012...789”. The extension of the configuration file is .tmp.

An example of a configuration file path: `%temp%\rb1us0wm99sslpa1vx.tmp`.

During the second run, the downloader reads the configuration data from the file, not from itself – this way, the attacker can easily update the configuration.

Thanks to its secure design, the configuration is not left in memory in unencrypted form. Every time the downloader needs to use some element of the configuration file, it decrypts the configuration file, retrieves the member and encrypts the file again.

This design protects the malware’s primary function – C&C communication – against memory forensics.

```
87      decryptConfig();
88      v3 = useSecondCnCForSecond_ime ? wprintf(&CnC, 0x104u, L"%S", config.CnC2_A) : wprintf(
                                                                                        &CnC,
                                                                                        0x104u,
                                                                                        L"%S",
                                                                                        config.CnC1_A);
92      v4 = v3;
94      *port = config.port;
95      arg10_halfMinuteFromConfButTimeOut = config.timeOutInSec;
96      if ( config.flag_SSPI_or_customTLS )
97      {
98        SSPI_or_customTLS = 1;
99        v5 = 4960;
100       v6 = &proxyPort;
101       do
102       {
103         *v6 = 0;
104         v6 = (v6 + 1);
105         --v5;
106       }
107       while ( v5 );
108       wstrcat(Dst, 260, proxyIp);
109       proxyPort = ::proxyPort;
110       wstrcat(a1, 260, &proxyUsername);
111       wstrcat(v26, 260, proxyPassword);
112       v35 = serverSync_;
113       wprintf(proxyCnC1, 0x104u, L"%S", config.ProxyCnC1);
114       port2 = config.port2;
115       wstrcat(userName, 260, config.UserNameW_1);
116       wstrcat(password, 260, config.PasswordW_1);
117       wprintf(&proxyCnC2, 0x104u, L"%S", config.ProxyCnC2);
118       port3 = config.port3;
119       wstrcat(&v33, 260, config.UserNameW_2);
120       v4 = wstrcat(&v34, 260, config.PasswordW_2);
121     }
122     else
123     {
124       SSPI_or_customTLS = 0;
125     }
126     v36 = config.flag != 0;
127     encryptConfig();
```

(https://www.welivesecurity.com/) (https://www.eset.com)

(https://www.welivesecurity.com/wp-content/uploads/2019/11/Figure-2.png)

Figure 2. Part of the code as seen by the Hex-Rays decompiler, which illustrates how the DePriMon malware decrypts the configuration file, saves a few members to local variables and encrypts it again

Of interest in the configuration file are:

Two entries for usernames and two members for passwords – for the proxy server if it is set on the machine. It means attackers are preparing to further their attack via a proxy with credentials. However, we haven’t seen functionality for stealing these details, so it appears that it is done in another phase of the attack.

Three entries for three C&C servers – each of them used on a different occasion.

Three entries for three ports – each of them used on a different occasion.

Flags indicating whether the downloader initializes a connection through Security Support Provider Interface (SSPI) with a possible proxy or only with a socket (described later).

It should be noted that besides C&C servers extracted from malware samples, we identified additional domains and servers likely related to this malware.

Encryption

The malware uses the AES encryption algorithm with three different 256-bit keys for different purposes (these keys are listed in the IoCs section).

Key 1: For decryption of various sensitive strings in the malware.

Key 2: For encryption and decryption of the configuration data in memory (as described earlier). This key is also used to generate the third key.

Key 3: For encryption and decryption of the configuration file on disk.

This key is not hardcoded but derived using a 32-byte array which is then encrypted. The array is generated as follows: the first 4 bytes are the volume serial number of the system drive, and the remaining 28 bytes contain the values 5 – 32. This array is encrypted with Key 2, resulting in Key 3.

Conclusion

DePriMon is an unusually advanced downloader whose developers have put extra effort into setting up the architecture and crafting the critical components.

DePriMon is downloaded to memory and executed directly from there as a DLL using the reflective DLL loading technique. It is never stored on disk. It has a surprisingly extensive configuration file with several interesting elements, its encryption is properly implemented and protects the C&C communication effectively.

As a result, DePriMon is a powerful, flexible and persistent tool designed to download a payload and execute it, and to collect some basic information about the system and its user along the way.

Indicators of Compromise (IoCs)

ESET detection names

```
Win32/DePriMon
Win64/DePriMon
```

SHA-1 hashes

```
02B38F6E8B54885FA967851A5580F61C14A0AAB6
03E047DD4CECB16F513C44599BF9B8BA82D0B7CB
0996C280AB704E95C9043C5A250CCE077DF9C8B2
15EBE328A501B1D603E66762FBB4583D73E109F7
1911F6E8B05E38A3C994048C759C5EA2B95CE5F7
2B30BE3F39DEF1F404264D8858B89769E6C032D9
2D80B235CDF41E09D055DD1B01FD690E13BE0AC7
6DB79671A3F31F7A9BB870151792A56276619DC1
6FAB7AA0479D41700981983A39F962F28CCFBE29
7D0B08654B47329AD6AE44B8FF158105EA736BC3
```

```
7E8A7273C5A0D49DF56DA04FEF963E30D5258814
8B4F3A06BA41F859E4CC394985BB788D5F76C85C
94C0BE25077D9A76F14A63CBF7A774A96E8006B8
968B52550062848A717027C512AFEDED19254F58
9C4BADE47865E8111DD3EEE6C5C4BC83F2489F5B
AA59CB6715CFFF545579861E5E77308F6CAEAC36
C2388C2B2ED6063EACBA8A4021CE32EB0929FAD2
CA34050771678C650400658227 29F44B35C87B0C
D38045B42C7E87C199993AB929AD92ADE4F82398
E272FDA0E9BA1A1B8EF444FF5F2E8EE419746384
E2D39E290201010F49652EE6116FD9B35C9AD882
F413EEE3CFD85A60D7AFC4D4ECC4445BB1F0B8BC
```

Domains

Domain	IP address
img.dealscienters[.]net	138.59.32.72
teknikgorus[.]com	88.119.179.17
wnupdnew[.]com	190.0.226.147
babmaftuh[.]com	185.56.89.196
alwatantrade[.]com	188.241.60.109
shayalyawm[.]com	5.226.168.124
elehenishing[.]com	185.225.17.77
almawaddrial[.]com	46.151.212.202
mdeastserv[.]com	46.151.212.201

Keys – example

Key 1:
C097CF17DC3303BC81553435046E50176ACA63842B0973831D8C6

Key 2:
8D35913F80A23E820C23B3125ABF57901BC9A7B83283FB2B240193A
Key 3: Derived as described earlier.

Filenames

```
dpnvmrs.dll
hp3mlnv.dll
hp4mlnv.dll
hp5nhd.dll
hp6nhd.dll
hpjdnb64.dll
hpmdnel3b.dll
ifssvc.dll
ifssvcmgr.dll
msprtmon64.dll
msptromn.dll
plamgr.dll
ppcrlchk.dll
ppcrlupd.dll
prntapt.dll
prntqdl64.dll
pscript6f.dll
pscript6s.dll
shprn64.dll
stprn32.dll
winmnprt.dll
```

MITRE ATT&CK techniques

Tactic	ID	Name	Descri
Persistence	T1013 (https://attack.mitre.org/techniques/T1013/)	Port Monitors	DePriM achievir
Defense Evasion	T1036 (https://attack.mitre.org/techniques/T1036/)	Masquerading	DePriM names r
	T1107 (https://attack.mitre.org/techniques/T1107/)	File Deletion	DePriM random
	T1112 (https://attack.mitre.org/techniques/T1112/)	Modify Registry	DePriM `HKLM\S` to achie
	T1134 (https://attack.mitre.org/techniques/T1134/)	Access Token Manipulation	DePriM proxy se
	T1140 (https://attack.mitre.org/techniques/T1140/)	Deobfuscate/Decode Files or Information	DePriM using A
Discovery	T1007 (https://attack.mitre.org/techniques/T1007/)	System Service Discovery	DePriM
	T1057 (https://attack.mitre.org/techniques/T1057/)	Process Discovery	DePriM
	T1082 (https://attack.mitre.org/techniques/T1082/)	System Information Discovery	DePriM
	T1124 (https://attack.mitre.org/techniques/T1124/)	System Time Discovery	DePriM actions
Command And Control	T1043 (https://attack.mitre.org/techniques/T1043/)	Commonly Used Port	DePriM
	T1071 (https://attack.mitre.org/techniques/T1071/)	Standard Application Layer Protocol	DePriM
	T1090 (https://attack.mitre.org/techniques/T1090/)	Connection Proxy	DePriM suspicic

ESET Research (https://www.welivesecurity.com/author/esetresearch/) 21 Nov 2019 - 11:30AM

Similar Articles

(https://www.welivesecurity.com/) (https://www.eset.com)

(https://www.welivesecurity.com/2019/11/botnet-adds-cryptomining-criminal-activities/)

Stantinko botnet adds cryptomining to its pool of criminal activities (https://www.welivesecurity.com/2019/11/26/stantinko-botnet-adds-cryptomining-criminal-activities/)

(https://www.welivesecurity.com/2019/11/advertisement-discounted-unhappy-meal/)

Mispadu: Advertisement for a discounted Unhappy Meal (https://www.welivesecurity.com/2019/11/19/advertisement-discounted-unhappy-meal/)

Discussion

welivesecurity™ BY eset®

(https://www.welivesecurity.com/) (https://www.eset.com)

Home (/)

About Us (https://www.welivesecurity.com/about-us/)

Research (https://www.welivesecurity.com/research/)

Contact Us
(https://www.welivesecurity.com/contact-us/)

How To
https://www.welivesecurity.com/category/how-to/)

(https://www.welivesecurity.com/) (https://www.eset.com)

Sitemap
(https://www.welivesecurity.com/sitemap/)

Categories
(https://www.welivesecurity.com/categories/)

Our Experts
(https://www.welivesecurity.com/our-experts/)

RSS Configurator
(https://www.welivesecurity.com/rss-configurator/)

ESET (https://eset.com)

News Widget
(https://www.welivesecurity.com/news-widget-generator/)

()

Privacy policy (https://www.welivesecurity.com/privacy/)

Legal Information (https://www.welivesecurity.com/legal-information/)