

## nao-sec.org

By nao\_sec

Published: 2021-01-04 · Archived: 2026-04-05 16:53:27 UTC

### Royal Road! Re:Dive

2021-01-04



### Abstract

We introduced the “Royal Road RTF Weaponizer” in our previous blog [1] (and presented at Japan Security Analyst Conference 2020 and CPX 360 CPRCon 2020). Royal Road is a tool shared by many targeted attack groups believed to belong to China. It’s been a year since our previous blog, and Royal Road is still in use. Here, we will introduce the Royal Road-related attacks observed during 2020.

### Previous Blog

Let’s briefly review the previous blog. Royal Road is a tool that generates RTF files that exploit the Microsoft Office Equation Editor vulnerabilities (CVE-2017-11882, CVE-2018-0798, CVE-2018-0802). The details of the tool are unknown, but the RTF file generated by it has various characteristics. The definition of “RTF file generated by Royal Road” may vary from researcher to researcher. Therefore, we define a file that meets the following conditions as an “RTF file generated by Royal Road”.

1. Exploiting a vulnerability in Microsoft Office Equation Editor
2. Containing an object named "8.t"

However, some RTF files are likely to be related to Royal Road, even though they don't meet the second condition. For classification purposes, we refer to this as "Related Samples". In reality, this may also be an RTF file generated by Royal Road, but the truth is only known to the attacker. Due to our research, we have divided these into "Royal Road Samples" and "Related Samples". However, they are treated the same in the specific case studies below.

And Royal Road is shared among various attack groups believed to belong to China. Specifically, it is believed to be used by the following attack groups. The attack group alias is written for reference. Strictly speaking, these can be different. For example, TA428 and Pirate Panda are not exactly equivalent.

1. Temp.Tick (BRONZE BUTLER, RedBaldKnight)
2. Temp.Conimes (Goblin Panda, Cycldek)
3. Temp.Periscope (Leviathan, APT40)
4. Temp.Trident (Dagger Panda, IceFog)
5. Tonto (Karma Panda, CactusPete, LoneRanger)
6. TA428 (Pirate Panda)
7. Rancor

Also, we categorized the various characteristics of the RTF files used by these groups and showed what they have in common.

Group-A	Group-B	
Temp.Conimes	Temp.Trident	TA428
Temp.Periscope		
Rancor	Tick	Tonto

## Updates

It's been a year since we introduced Royal Road. In the meantime, the RTF file, believed to have been generated by Royal Road, has been used many times in targeted attacks, and several updates have been observed. First of all,

we will introduce the updates.

The RTF file generated by Royal Road contains encoded malware. It is decoded by Shellcode after exploit. In our previous blog, we introduced the following 5 encodings.

1. 4D 5A 90 00 (not encoded)
2. F2 A3 20 72
3. B2 A6 6D FF
4. B0 74 77 46
5. B2 5A 6F 00

Many of the RTF files we observed in 2020 used the 3rd and 4th encodings. However, a few samples used the new encodings. The following 2 encodings.

1. A9 A4 6E FE

The image displays two hex dump screenshots. The top screenshot shows a hex dump with a yellow header row: ADDRESS 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F 0123456789ABCDEF. The first few lines of data are: 00000000 A9 A4 6E FE F3 FE FE FE F2 FE FE FE FF FF FE FE, 00000010 46 FE FE FE FE FE FE FE BE FE FE FE FE FE FE FE, 00000020 FE FE FE FE FE FE FE FE FE FE FE FE FE FE FE FE, 00000030 FE FE FE FE FE FE FE FE FE FE FE FE FE 1E FE FE FE, 00000040 E8 DF 44 E8 FE 42 F5 29 DD 46 FD AA 29 DD A2 96, 00000050 95 83 DE 8E 8C 8F 97 8C 9D 89 DE 93 9D 88 88 8F, 00000060 82 DE 9C 91 DE 8C 81 88 DE 95 88 DE B2 AF A3 DE, 00000070 89 8F 92 91 C8 E9 E9 F4 D2 FE FE FE FE FE FE FE, 00000080 A9 17 C6 7E F5 70 A0 23 F5 70 A0 23 F5 70 A0 23, 00000090 1D 65 AC 23 EB 70 A0 23 1D 65 9A 23 E9 70 A0 23, 000000A0 C8 BE C9 23 E8 70 A0 23 F5 70 A7 23 DC 70 A0 23, 000000B0 FE F8 1A 23 EB 70 A0 23 FE F8 32 23 F6 70 A0 23, 000000C0 FE F8 37 23 F6 70 A0 23 AC 95 93 96 F5 70 A0 23, 000000D0 FE FE FE FE FE FE FE FE FE FE FE FE FE FE FE, 000000E0 AE B1 FE FE AA FD F3 FE 74 EA A0 98 FE FE FE FE, 000000F0 FE FE FE FE 1E FE FC DD EB FD F5 FE FE CE FE FE. The decoded text on the right includes: 0123456789ABCDEF, 00000000 0x, 00000010 F., 00000020, 00000030, 00000040 轄D..B.)ンF.ェ)ン抹, 00000050 ・詞女拳迦統・盾, 00000060 '恆'載菱譜'イ', 00000070 縁耐視, 00000080 ウ.ニ. #. #. #, 00000090 .eヤ#. #.e.#駱.#, 000000A0 社/#鐘.#.ア#7p.#, 000000B0 ...#. #..2#. #, 000000C0 ..7#. #.ヤ無靖p.#, 000000D0, 000000E0 ヲア..エ...t槓..., 000000F0 ..... \* .....ホ..

The bottom screenshot shows the same hex dump but with the 'A9 A4 6E FE' bytes decoded to '4D 5A 90 00' at address 00000000. The decoded text on the right includes: 0123456789ABCDEF, 00000000 MZ., 00000010 ク.....@., 00000020, 00000030, 00000040 ..コ..エ.ハ!ク.L!Th, 00000050 is program cannot be run in DOS, 00000060 mode....\$. ....., 00000070 M.8.. \* 毛. \* 毛. \* 毛, 00000080 瘡R毛. \* 毛瘡毛. \* 毛, 00000090 .@-毛. \* 毛. \* 毛" \* 毛, 000000A0 ..7毛. \* 毛..ト毛. \* 毛, 000000B0 ..又毛. \* 毛Rich. \* 毛, 000000C0, 000000D0, 000000E0 PE..L.....V^...., 000000F0 .....!.....0..

This encoding can be decoded with code like the following:

```
dec_data = []
```

```
for i in range(len(enc_data)):
```

```
    dec_data.append(((int.from_bytes(enc_data[i], "little") ^ 0x7b) + 0x7b) % 256)
```

1. 94 5F DA D8

ADDRESS	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF
00000000	94	5F	DA	D8	F9	B8	BA	9E	D2	C8	E9	74	3F	04	17	1C	農ノ・コ樽袴?... [アH].>耀.ヌ.・Pε. .セ^慈ニ..ヨfオtヲトヨ 蘇ハ..ミ.・.-3Q槻ブ 絮ヤ.9轉~E@ニ*Y禿+ j)エpヤ・->-bD満=.0 ..意堡クノ朝L.7. Q 烈ttto.zヤ.マ.イ. .ホ.<.e2WIK7ヨカ` ウチ姉.2:.;.L州・ .;CA.Zヒオ.;vR)PL d匡.?マ2-ヌウ・ル:セ. .飢Cムヌヤ・价o.;ヒ ..・.0呎w:ヲ辟2詢 .>3.x恒9せ°6ヌ..Q ..・.府`ソe0が`アP.
00000010	7C	B1	48	D8	FD	3E	97	73	02	C7	2C	F5	6D	50	AA	E7	
00000020	18	BE	CD	5E	FA	F4	C6	E3	1E	AE	DC	B5	74	CF	74	AE	
00000030	91	68	CA	F1	12	D0	88	84	E5	2D	33	51	92	CE	83	76	
00000040	E3	50	D4	E9	39	E7	75	7E	45	40	C6	2A	59	93	C3	2B	
00000050	6A	29	AD	70	D4	F1	DE	3E	B0	62	44	9F	DE	3D	13	4F	
00000060	FD	0E	88	D3	9A	C6	B8	2E	C9	11	92	A9	4C	FD	DC	09	
00000070	20	51	DA	97	F3	74	74	6F	0F	7A	AC	80	CF	1E	B4	A5	
00000080	A0	CE	02	02	3C	FF	65	D5	57	69	4B	CC	D6	B9	3E	60	
00000090	B3	83	60	8E	6F	09	32	3B	08	3B	13	4C	C5	48	F0	41	
000000A0	1A	3B	43	CA	01	5A	CB	B5	F6	00	3B	76	52	29	50	4C	
000000B0	6F	8B	A7	F2	3F	CF	32	2D	BD	A9	EA	FC	D9	3A	BE	01	
000000C0	09	8B	51	43	83	80	BD	AC	88	8E	98	C1	6F	EF	3B	CB	
000000D0	97	18	F7	62	07	30	99	E6	77	3A	CF	E7	84	32	E6	6D	
000000E0	FF	3E	33	A4	78	8D	50	39	BE	DF	36	83	58	7F	0B	51	
000000F0	06	F2	4F	12	95	7B	DE	BF	65	30	B6	D4	22	B1	50	97	

ADDRESS	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	0123456789ABCDEF
00000000	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00	MZ.....
00000010	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00	ク.....@.....
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000030	00	00	00	00	00	00	00	00	00	00	00	00	F0	00	00	00	.....
00000040	0E	1F	BA	0E	00	B4	09	CD	21	B8	01	4C	CD	21	54	68	..コ..I..^!ク.L^!Th
00000050	69	73	20	70	72	6F	67	72	61	6D	20	63	61	6E	6E	6F	is program canno
00000060	74	20	62	65	20	72	75	6E	20	69	6E	20	44	4F	53	20	t be run in DOS
00000070	6D	6F	64	65	2E	0D	0D	0A	24	00	00	00	00	00	00	00	mode....\$......
00000080	F3	A4	81	5E	B7	C5	EF	0D	B7	C5	EF	0D	B7	C5	EF	0D	・/村..村..村..
00000090	D2	A3	EC	0C	BD	C5	EF	0D	D2	A3	EA	0C	3F	C5	EF	0D	村..村..村..?村..
000000A0	D2	A3	EB	0C	A5	C5	EF	0D	62	A8	EA	0C	AA	C5	EF	0D	村..ナ..bi..エナ..
000000B0	62	A8	EB	0C	B8	C5	EF	0D	62	A8	EC	0C	A6	C5	EF	0D	bi..村..bi..ナ..
000000C0	D2	A3	EE	0C	B2	C5	EF	0D	B7	C5	EE	0D	E6	C5	EF	0D	村..イ..村..賢..
000000D0	2D	AB	E6	0C	B5	C5	EF	0D	2D	AB	ED	0C	B6	C5	EF	0D	-オ..村..-オ..村..
000000E0	52	69	63	68	B7	C5	EF	0D	00	00	00	00	00	00	00	00	Rich村.....
000000F0	50	45	00	00	4C	01	04	00	16	0B	FE	5D	00	00	00	00	PE..L.....]

This encoding can be decoded with code like the following:

```
dec_data = []
```

```
xor_key = 1387678300
```

```
for i in range(len(enc_data)):
```

```
    for _ in range(7):
```

```
        x0 = (xor_key & 0x20000000) == 0x20000000
```

```
        x1 = (xor_key & 8) == 8
```

```
x2 = xor_key & 1
x3 = 1 + (x0 ^ x1 ^ x2)
xor_key = (xor_key + xor_key) + x3
dec_data.append(int.from_bytes(enc_data[i], "little") ^ (xor_key % 256))
```

Our tool for decrypting Royal Road encoded object is already available on GitHub. It also supports the above new encodings.

[https://github.com/nao-sec/rr\\_decoder](https://github.com/nao-sec/rr_decoder)

## New Attack Groups

As we mentioned earlier, several attack groups use Royal Road. The following eight attack groups have been observed to use Royal Road (including both Royal Road Samples and Related Samples) during 2020.

1. Temp.Conimes
2. Tonto
3. TA428
4. Naikon
5. Higaisa
6. Vicious Panda
7. FunnyDream
8. TA410

Of these, we have already reported on 1-3 attack groups in our previous blog. Temp.Conimes used NewCore RAT to attack Vietnamese organizations. Tonto used Bisonal to attack organizations in East Asia such as Russia.

And the TA428 was also particularly active, using PoisonIvy, Cotx RAT, Tmanger, and nccTrojan to attack East Asian organizations such as Mongolia. We will not cover these individual cases here, but if you are interested, see the IOC chapter. For TA428, the paper [2] and blogs [3][4][5] are available from NSJ (NTT Security Japan). Please refer to that.

For Naikon, CheckPoint Research reported [6], but unfortunately, we could not observe this. Therefore, in the following, we will introduce attack cases related to Royal Road for four groups (5-8).

### Higaisa

Higaisa is an attack group that seems to have been active since at least around 2016. It is primarily targeted at North Korean-related organizations and is believed to be aimed at stealing information using AttackBot, PIZ Stealer, and Gh0st RAT.

The blogs have been written by Tencent and Positive Technologies so far [7][8][9], and are attributed to (South) Korea. However, NSJ's paper [10] showed a connection with Ghost Dragon [11] and PKPLUG [12], and it was reported that it might belong to China.

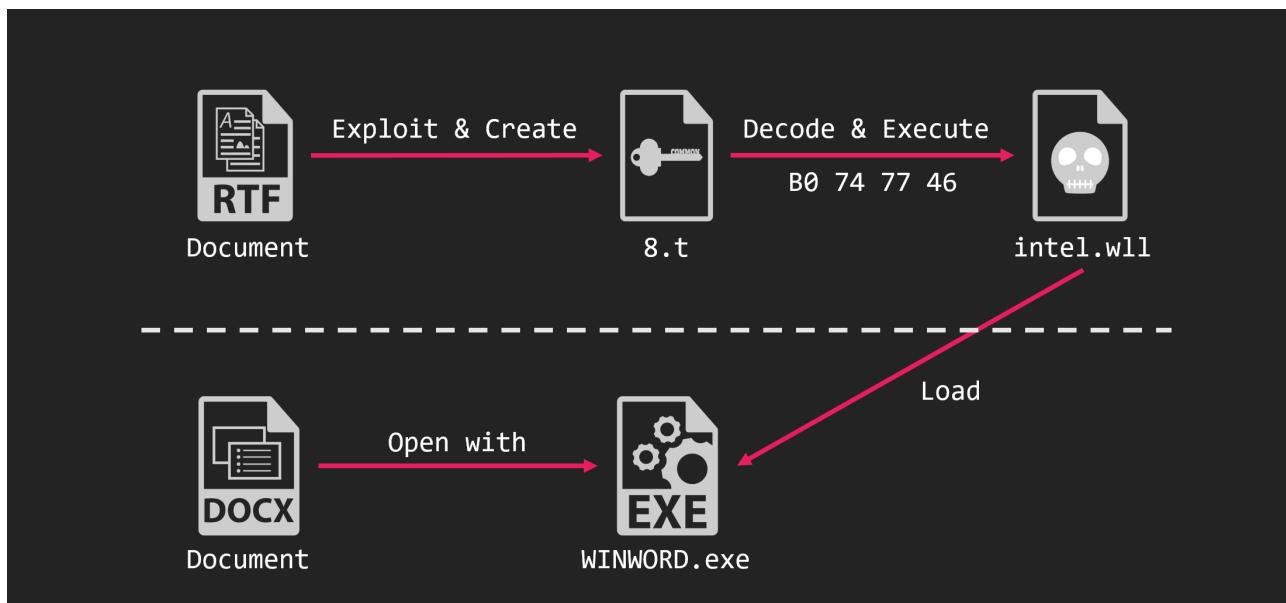
We observed an attack by Higaisa on Royal Road in March 2020.

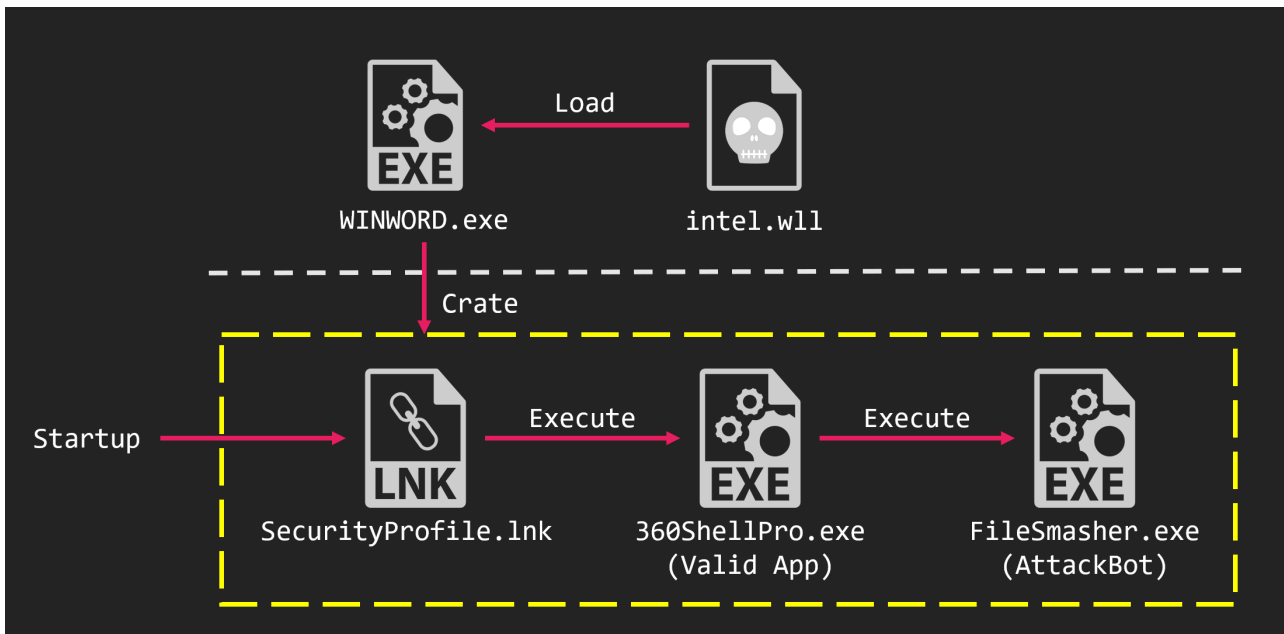
Merry Christmas And Happy New Year!

Hope all your Christmas dreams come true!

I wish you a merry Christmas. All affection and best wishes to you and yours.

..  
..  
..  
..





The malware executed by the Royal Road RTF was AttackBot. AttackBot is a downloader that has been used by Higaisa since at least April 2018.

```
0x00401063    mov     ecx, 5
0x00401068    mov     eax, 8
0x0040106d    xor     esi, esi
0x0040106f    nop
0x00401070    add     eax, ecx
0x00401072    cdq
0x00401073    mov     edi, 0xff ; 255
0x00401078    idiv   edi
0x0040107a    inc     esi
0x0040107b    mov     eax, ecx
0x0040107d    mov     byte [ebp + esi - 0x45], dl
0x00401081    movzx  ecx, dl
0x00401084    cmp     esi, 0x40 ; 64
0x00401087    jb     0x401070
```

### Vicious Panda

Vicious Panda is an attack group reported by CheckPoint Research in March 2020 [13]. It is said to belong to China and targets East Asia such as Russia, Mongolia, and Ukraine.

We observed an attack on the Royal Road by Vicious Panda in March 2020.

БНХАУ-ын Төрийн зөвлөлийн гишүүн, Гадаад хэргийн сайд Ван И “2020 он бол Хятад Үндэстний агуу их сэргэн мандалтын явц дахь түүхэн ач холбогдол бүхий жил бөгөөд гадаад харилцаанд 6 чухал үүргийг анхаарах ёстой”-г илэрхийлсэн.

Энэ хүрээнд 2020 онд гадаад харилцааны чиглэлээр дараах зорилтуудыг хэрэгжүүлэхээр төлөвлөөд байна.

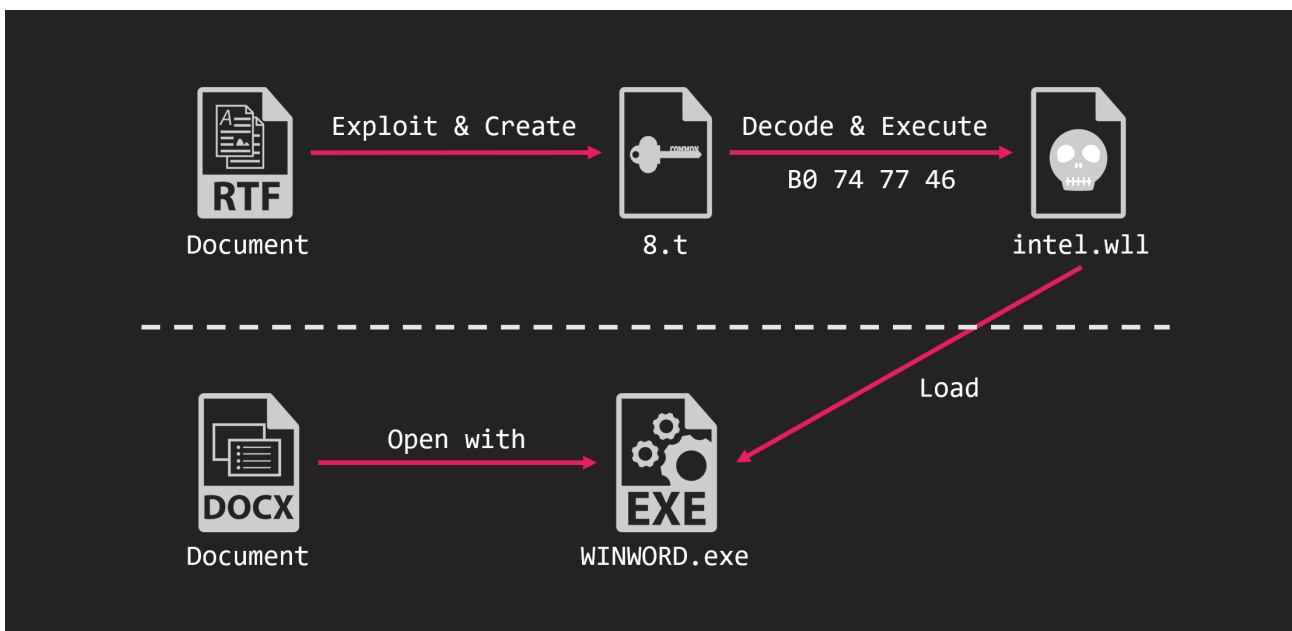
Нэгдүгээрт. *Дотоодын хөгжилд бүх чадлаараа үйлчлэх.*

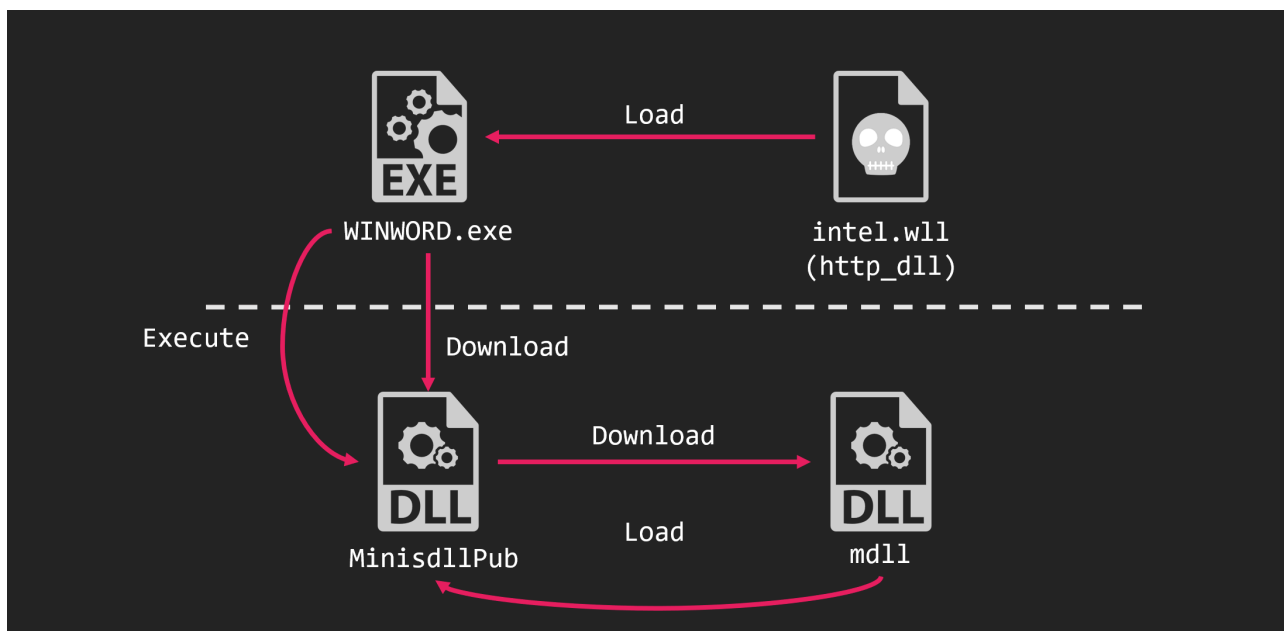
Чинээлэг нийгмийг бүх талаар бий болгох нь Хятад улсын “хос зуун” жилийн чухал зорилт бөгөөд үүний тулд олон улс бүс нутгийн таатай гадаад орчинг бүрдүүлэхийн төлөө чармайлт гаргаж ирсэн. Ван И хэлэхдээ, 2020 онд дотоод, гадаад нөхцөл байдлыг харгалзан, гадаад харилцааны нөөц боломжийг бүрэн дүүрэн ашиглаж, улсын хөгжлийн томоохон стратегийг хэрэгжүүлэхийн зэрэгцээ гадаад сурталчилгаа, танилцуулах арга хэмжээг улам идэвхтэй зохион байгуулж, орон нутаг дэлхий нийттэй харилцан ашигтай хамтын ажиллах өргөн индэрийг бий болгохыг зорьж буйгаа илэрхийлсэн.

Хоёрдугаарт. *Улсын эрх ашгийг эрс шийдэмгий хамгаална.*

Хятал улс галаал харилцааны нээлттэй боллогыг төлөвшүүлэхийн зэрэгцээ улсын

It has been reported to execute malware similar to Enfal and BYEBY.





## FunnyDream

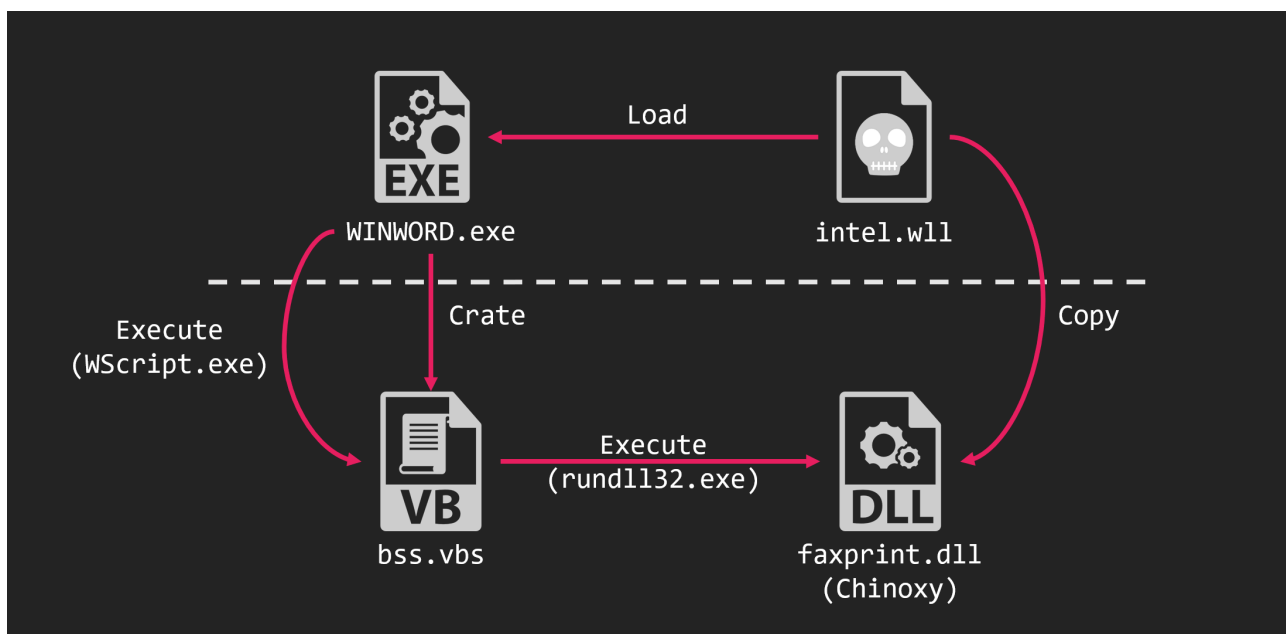
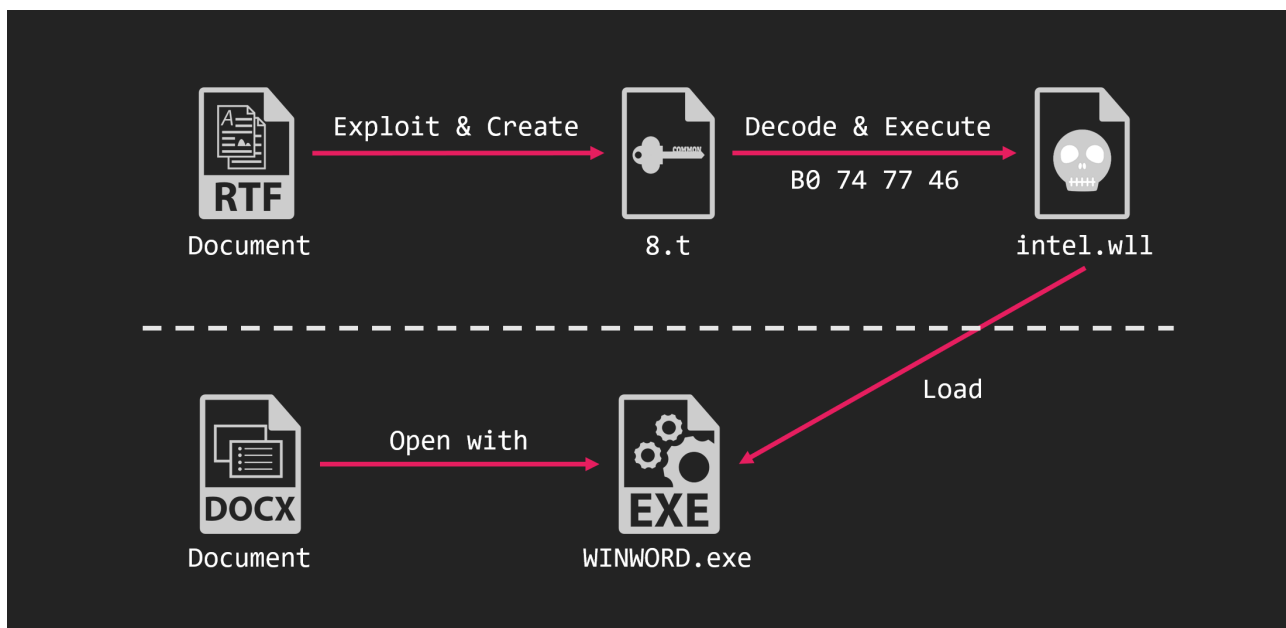
FunnyDream is an attack group that is said to have been active since around 2018. It is said to belong to China and targets Southeast Asia such as Vietnam and Malaysia. FunnyDream uses Chinoxy and FunnyDream Backdoor. BitDefender has published a detailed report [14] on FunnyDream.

We observed an attack by FunnyDream from March to May 2020.

**Thủ trưởng Vịnh giao cho Công tiếp thu nội dung của CDN, chỉnh sửa, hoàn chỉnh bài viết của TT, hoàn thành trước 15.00, ngày 20/5/2020.**  
**Bài viết để đăng tải trên Đặc san “VN-HK: Dấu ấn 25 năm” của Báo TG&VN/Bộ Ngoại giao. Y/c bài viết khoảng 1.000 chữ.**  
**Đề nghị các đồng chí nghiên cứu bài viết này, đầu sáng mai (19/) cho ý kiến/hướng xử lý hoàn chỉnh.**

**QUÂN SỰ-QUỐC PHÒNG – NHÂN TỐ THỨC ĐÃY**  
**QUAN HỆ VIỆT NAM – HOA KỲ SAU CHIẾN TRANH**

**(Bài viết của đồng chí Thượng tướng Nguyễn Chí Vịnh,**  
**Thủ trưởng Bộ Quốc phòng đăng trên Báo Thế giới & Việt Nam.**



Chinoxy is a RAT that has been used by FunnyDream since around 2018. It decoded the config using two numeric data and communicates with the C&C server using its original protocol using Blowfish.

## TA410

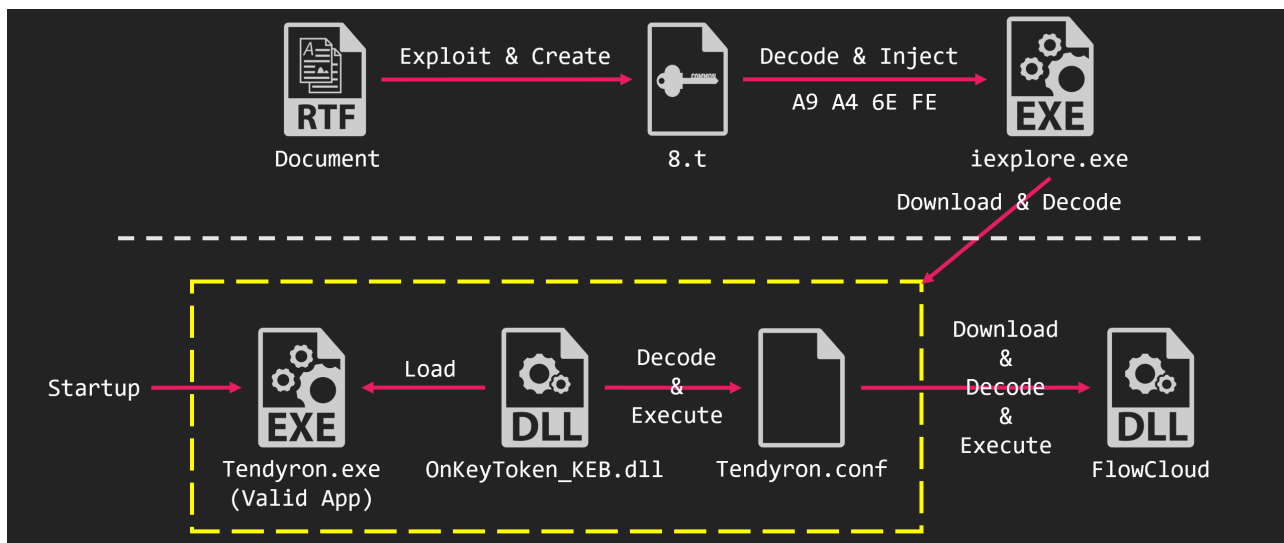
TA410 is an attack group that is said to have been active since around 2016. It is said to belong to China and is suspected to be related to APT10. The report has been published by Proofpoint [15][16][17] and is mainly targeted at public sector in the US. It uses malware called LockBack and FlowCloud.

We observed an attack by TA410 in October 2020.

### 怎样看待蔡双十讲话的内容和风格。

总的评价，她的讲话是在放烟雾弹，是暗度陈仓，掩人耳目。

第一，蔡讲话通篇不同于以往，几乎完全不触及涉及两岸政治定位和台湾政治前途的敏感语汇，没有回应“九二共识”、没有出现“中国”或“中华民国”，也没有提“维持现状”或“独立”与否，连随时挂在嘴边的“自主”“尊严”或“主权”等都较少浮现。显然是要有意回避，在双十这样的重要“节日”，不提“国家”，只讲社会，显得毫不切题、不伦不类。这与她惯有的风格和一段时间以来“台独”



FlowCloud is a RAT reported by Proofpoint in June 2020. FlowCloud has been reported to be v4 and v5, but the FlowCloud we observed at this time was similar to v4.

## Attack case against Japan

In addition to the four attack groups shown so far (Higaisa, Vicious Panda, FunnyDream, TA410), attacks that appear to be related to Royal Road have been observed. Among them, we will introduce an example of attacks on Japan. We are not able to identify which attack group made this attack. If you have any knowledge about it, please share it with us...

The attack on Japan took place in November 2020. The attack began with 2 RTF files attached to the email.

経費支出明細							
<経費明細総括表>							
(単位: 円)							
申請者名	予算額 (交付決定額または変更申請額)			実績額			
	A	B	B×2/3以内	A	B	B×2/3以内	
補助事業に要する経費	補助対象経費	補助金交付決定額	補助事業に要した経費	補助対象経費	補助金の額		
	(税込み)	(税抜き)	(税抜き)	(税込み)	(税抜き)	(税抜き)	(税抜き)
<代表者> 補助事業者名							
合計	0	0	0	0	0	0	0
<経費明細表>							
(事業者名: )							
(単位: 円)							
	予算額 (交付決定額または変更申請額)			実績額			
	A	B	B×2/3以内	A	B	B×2/3以内	

精算請求書									
事業者名:									
管理No.	支払	支払先	内容および仕様等詳細	数量	単位	単価	補助事業に要した経費<支払額>	補助対象経費	
	年 月 日					(税込み)	(税込み)	(税抜き)	
1									
2									
3									
4									
5									
6									
7									
合計							0	0	0

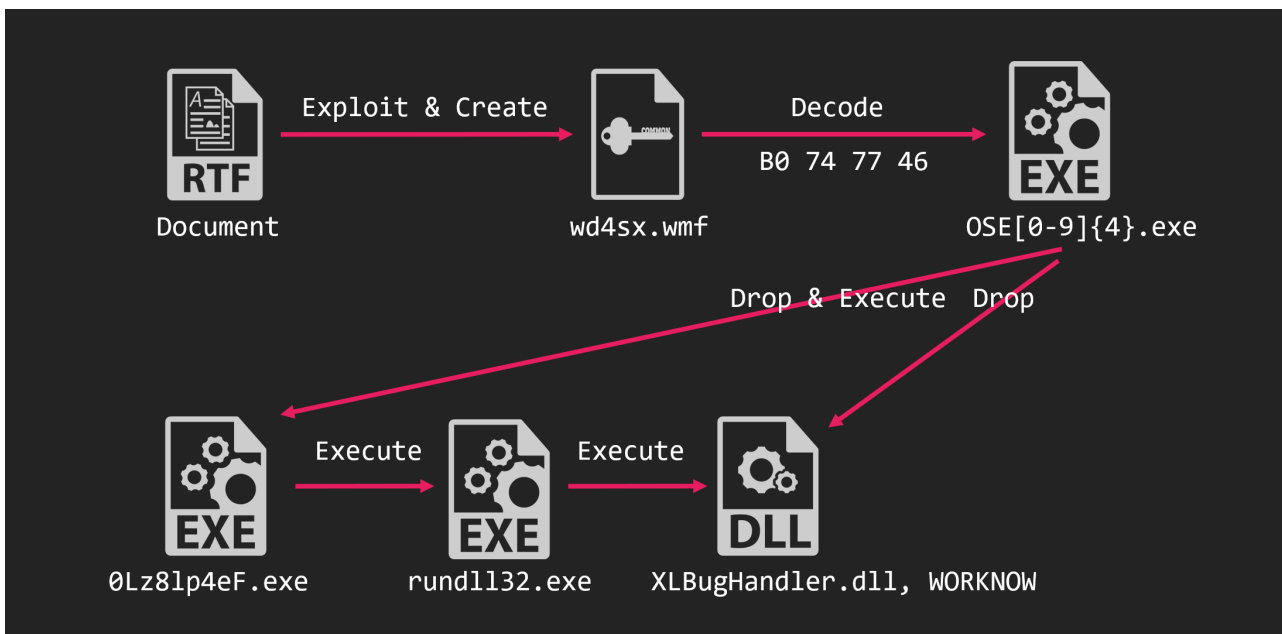
These RTF files did not contain an 8.t object, however did contain an associated object. This is the malware encoded by the 4th (B0 74 77 46) encoding shown above.

```

-----+-----+-----
id | index      | OLE Object
-----+-----+-----
0  | 00036596h | format_id: 2 (Embedded)
   |           | class name: 'Package'
   |           | data size: 553136
   |           | OLE Package object:
   |           | Filename: u'wd4sx.wmf'
   |           | Source path: u'C:¥¥Windows¥¥wd4sx.wmf'
   |           | Temp path = u'C:¥¥Windows¥¥wd4sx.wmf'
   |           | MD5 = '9a67a7ccace7d8f788663077d57d6811'
-----+-----+-----
1  | 00144786h | format_id: 2 (Embedded)
   |           | class name: 'Equation.2¥x00¥x124Vx¥x90¥x124VxvT2'
   |           | data size: 6436
   |           | MD5 = '00b38c4c02dcda01f8abea580c95a919'
-----+-----+-----
2  | 0014476Ch | Not a well-formed OLE object
-----+-----+-----

```

The overall picture of the attack is as follows.



The malware executed was an unknown RAT. We call this XLBug RAT because of the characteristics left in this RAT. The RAT held information such as C&C server encoded by Base64 and XOR.

1002588c	4d 59 41 ...	ds	"MYAAAArq+wvbKlvbKwva6xtamnr58="
100258ab	4d 59 42 ...	ds	"MYBBBB5PH88u+unw=="
100258be	4d 59 43 ...	ds	"MYCCCC0Lz8lp4eF"
100258ce	4d 59 44 ...	ds	"MYDDDD/f76Aq6f"

```
size = base64_decode(param_1, &buf);
i = 0;
if (0 < (int)size) {
    do {
        *(char *) (i + (int)buf) = *(char *) (i + (int)buf) + 'z';
        *(byte *) (i + (int)buf) = *(byte *) (i + (int)buf) ^ 0x19;
        i = i + 1;
    } while (i < (int)size);
}
return buf;
}
```



103.56.53.126:80  
Group1  
name1

The following commands are implemented in XLBug RAT.

- Get directory information
- Get file information
- Get computer information
- Execute file
- Upload file
- Download file
- Rename file
- Delete file
- Delete itself

The naming convention and encoding of the encoded object contained in the RTF are similar to those of the TA428. However, we could not say that this was a TA428 attack.

## Relationship

In the previous blog, we summarized the characteristics of attack groups that use Royal Road. We used it to divide the attack groups into two groups. However, by 2020, those characteristics are almost meaningless. It has been standardized or deleted. It's not as easy to group as it used to be. In the first place, the groups sharing Royal Road should be close. We do not classify further, but if you have any comments please let us know.

## Yara Rule

The GitHub repository we shared in the previous blog is still being updated.

[https://github.com/nao-sec/yara\\_rules](https://github.com/nao-sec/yara_rules)

## IOC

The IOC sheet shared in the previous blog is still being updated.

[https://nao-sec.org/jsac2020\\_ioc.html](https://nao-sec.org/jsac2020_ioc.html)

The tool used by Royal Road to decrypt encoded object is still being updated.

[https://github.com/nao-sec/rr\\_decoder](https://github.com/nao-sec/rr_decoder)

## Wrap-Up

The attacks using Royal Road have decreased compared to 2019, but are still ongoing. There are many cases of attacks by TA428 and Tonto, but other attacks by different attack groups (Higaisa, Vicious Panda, FunnyDream, TA410) have also been observed.

The attacks on Japan have also been observed and we were unable to identify this with a known attack group. The use of Royal Road by these unknown attack groups is expected to continue.

In addition to Royal Road, there are other cases, such as the Tmanger family, that appear to share tools among multiple targeted attack groups. We should continue to pay close attention to these tool sharing cases.

## Acknowledgments

“nao\_sec” is an independent research team that does not belong to any company. Individuals belong to each company and engage in research, but the activities of nao\_sec still maintain their independence from each company. We are grateful to all of you who cooperated with our research activities every day.

---

## References

- [1] nao\_sec, “An Overhead View of the Royal Road”, <https://nao-sec.org/2020/01/an-overhead-view-of-the-royal-road.html>
- [2] NTT Security Japan, “Operation LagTime IT: colourful Panda footprint”, <https://vbllocalhost.com/uploads/VB2020-Ozawa-etal.pdf>
- [3] NTT Security Japan, “Panda’s New Arsenal: Part 1 Tmanger”, <https://insight-jp.nttsecurity.com/post/102gi9b/pandas-new-arsenal-part-1-tmanger>
- [4] NTT Security Japan, “Panda’s New Arsenal: Part 2 Albaniutas”, <https://insight-jp.nttsecurity.com/post/102gkfp/pandas-new-arsenal-part-2-albaniutas>
- [5] NTT Security Japan, “Panda’s New Arsenal: Part 3 Smanager”, <https://insight-jp.nttsecurity.com/post/102glv5/pandas-new-arsenal-part-3-smanager>
- [6] CheckPoint Research, “Naikon APT: Cyber Espionage Reloaded”, <https://research.checkpoint.com/2020/naikon-apt-cyber-espionage-reloaded/>
- [7] Tencent, “APT攻击组织”黑格莎 (Higaisa) ”攻击活动披露”, <https://s.tencent.com/research/report/836.html>
- [8] Tencent, ““Higaisa (黑格莎) ”组织近期攻击活动报告”, <https://s.tencent.com/research/report/895.html>
- [9] Positive Technologies, “COVID-19 и новогодние поздравления: исследуем инструменты группировки Higaisa”, <https://www.ptsecurity.com/ru-ru/research/pt-esc-threat-intelligence/covid-19-i-novogodnie-pozdravleniya-issleduem-instrumenty-gruppirovki-higaisa/>
- [10] NTT Security Japan, “Crafty Panda 標的型攻撃解析レポート”, <https://www.nttsecurity.com/docs/librariesprovider3/default-document-library/craftypanda-analysis-report>
- [11] Cylance (BlackBerry), “The Ghost Dragon”, <https://blogs.blackberry.com/en/2016/04/the-ghost-dragon>
- [12] Palo Alto Networks, “PKPLUG: Chinese Cyber Espionage Group Attacking Southeast Asia”, [https://unit42.paloaltonetworks.com/pkplug\\_chinese\\_cyber\\_espionage\\_group\\_attacking\\_asia/](https://unit42.paloaltonetworks.com/pkplug_chinese_cyber_espionage_group_attacking_asia/)
- [13] CheckPoint Research, “Vicious Panda: The COVID Campaign”, <https://research.checkpoint.com/2020/vicious-panda-the-covid-campaign/>
- [14] BitDefender, “A Detailed Timeline of a Chinese APT Espionage Attack Targeting South Eastern Asian Government Institutions”, <https://labs.bitdefender.com/2020/11/a-detailed-timeline-of-a-chinese-apt-espionage-attack-targeting-south-eastern-asian-government-institutions/>
- [15] Proofpoint, “LookBack Malware Targets the United States Utilities Sector with Phishing Attacks Impersonating Engineering Licensing Boards”, <https://www.proofpoint.com/us/threat-insight/post/lookback-malware-targets-united-states-utilities-sector-phishing-attacks>
- [16] Proofpoint, “LookBack Forges Ahead: Continued Targeting of the United States’ Utilities Sector Reveals Additional Adversary TTPs”, <https://www.proofpoint.com/us/threat-insight/post/lookback-forges-ahead-continued-targeting-united-states-utilities-sector-reveals>
- [17] Proofpoint, “TA410: The Group Behind LookBack Attacks Against U.S. Utilities Sector Returns with New Malware”, <https://www.proofpoint.com/us/blog/threat-insight/ta410-group-behind-lookback-attacks-against-us-utilities-sector-returns-new>

---

Source: <https://nao-sec.org/2021/01/royal-road-redive.html>