I Know You Want Me - Unplugging PlugX

Takahiro Haruyama / Hiroshi Suzuki
Internet Initiative Japan Inc.

Who are we?

- Takahiro Haruyama (@cci_forensics)
  - Forensic Investigator & Malware Analyst
  - Presenter & Hands-on Trainer
    - SANS DFIR Summit, BlackHat USA/EU, CEIC, FIRST, RSA Conference JP, etc..
  - EnCase Certified Examiner since 2009
  - http://takahiroharuyama.github.io/
- Hiroshi Suzuki (@herosi_t)
  - Malware Analyst & Forensic Investigator
  - Presenter & Hands-on Trainer
    - Co-author of Haruyama's presentation of BlackHat USA/EU, FIRST, domestic conference in Japan (MWS, and others)

Overview

- Motivation
- Evolution of PlugX
- Parsing PlugX Config
- Associating PlugX samples with known targeted attack groups
  - Classifying PlugX samples
  - Finding relationships between PlugX groups and known attacker groups
- Wrap-up

MOTIVATION

What's PlugX?

- RAT used for targeted attacks
  - Also known as Korplug/Gulpix/Sogu/Thoper/Destory RAT
  - Acknowledged in earlier 2012
    - Trend Micro pointed out it was part of a campaign that has been since 2008 [1]
  - The author extends its implementation aggressively
    - AlienVault identified the author? [2] ☺
    - AhnLab acquired PlugX builder/controller [3]

Motivation

- PlugX includes config data like PoisonIvy
  - e.g., C2 hostname/IP/domain, installed service name/registry value
- The config information can be used for identifying attacker groups
  - FireEye published the analysis result about PoisonIvy configs [4]
  - So, what about PlugX?
- The author's continual update makes it difficult
  - 10 config size versions
  - frequently-changed encryption algorithms
- We tried to categorize PlugX samples in terms of attackers based on intelligence extracted from config data and code

EVOLUTION OF PLUGX

Type I: Behavior Summary

Dropper (exe)

create&run

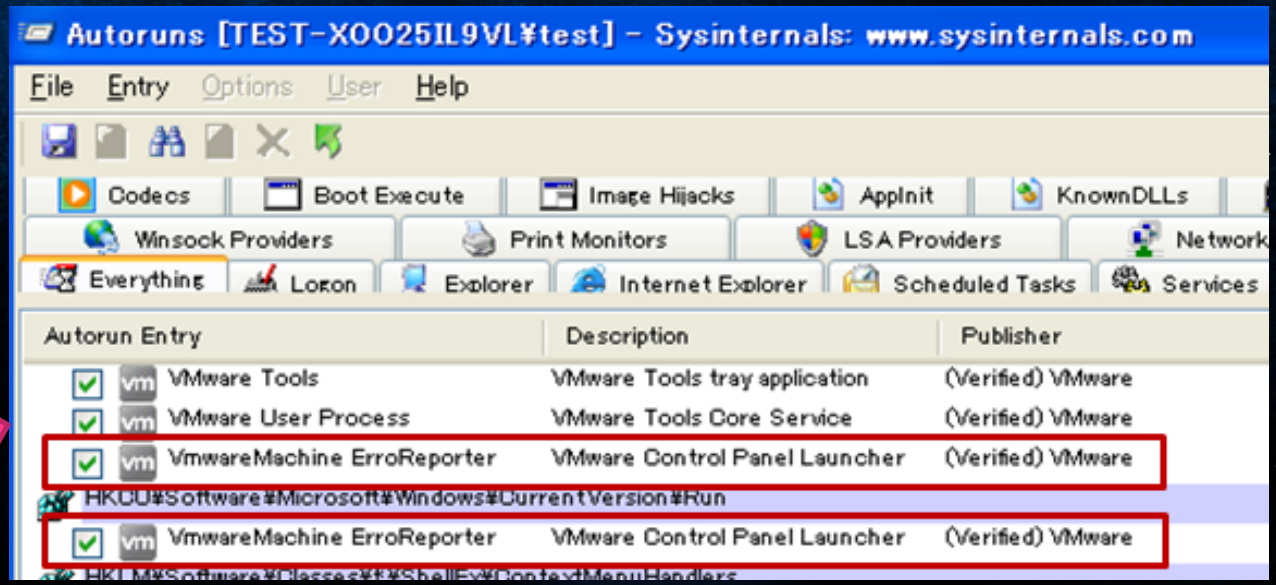
create

load
(DLL load order abuse)

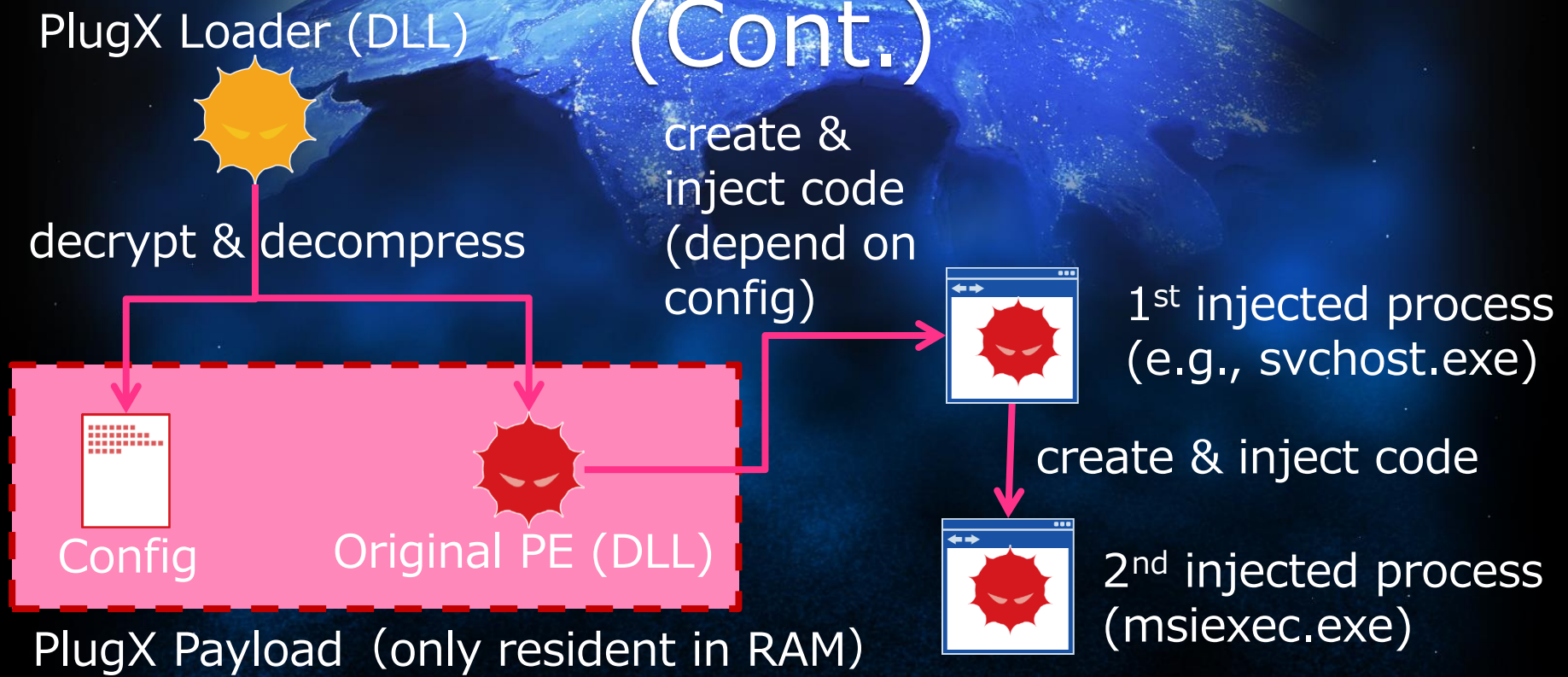
signed executable (exe)

PlugX Loader (DLL)

Cont.



Type I: Behavior Summary (Cont.)



Type I: Encrypted Data Table

Address	Value	ASCI
00990000	504C5547	GULP
00990004	10027DCC	
00990008	0001D03A	
0099000C	10029B01	
00990010	0001AC35	
00990014	10027DDF	
00990018	0001D18	
0099001C	00991490	

```
EncryptedDataTable struc ; (sizeof=0x20)
                                        ; XREF:
                                        ; sub_81
MAGIC_PLUG      dd ?                    ; XREF:
                                        ; func_X
PICPointer      dd ?                    ; XREF:
                                        ; XPlugS
TotalPayloadLength dd ?                 ; XREF:
                                        ; func_X
encryptedRealPlugXExecutable dd ?       ; XREF:
                                        ; func_X
encryptedRealPlugXExecutableLength dd ? ; XREF:
                                        ; func_X
ecnryptedConfig dd ?                    ; XREF:
                                        ; func_r
encryptedConfigLength dd ?              ; XREF:
                                        ; func_X
decrypted_and_decompressed_plugx_EntryPoint dd ?
                                        ; LdrLoa
EncryptedDataTable ends
```

Type I: Config (Xsetting)

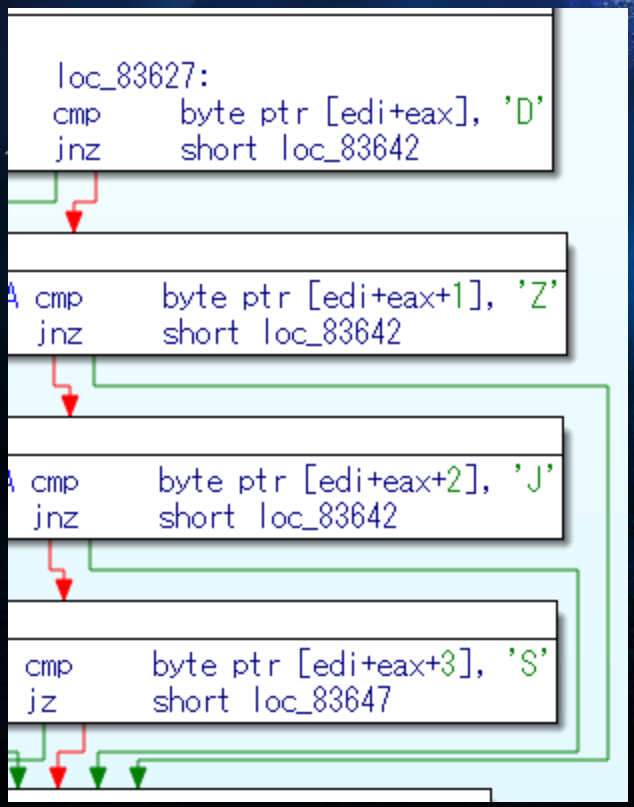
- C2 Setting
  - protocol, port, hostname
- C2 Setting URL
  - download C2 Setting from the URL
- Install Options
  - type (service, registry, registered_by_loader)
  - install folder path
  - service name, registry value
- Others
  - Proxy Setting
  - stand alone flag (not inject code)
  - code injection target
  - mutex name

```
XSetting        struc ; (sizeof=0x1D18) ;
                                        ;
field_0_xheader XHeader ?               ;
                                        ;
```

```
field_2EC_C2_hostname1 C2Setting ?

field_330_C2_hostname2 C2Setting ?

field_374_C2_hostname3 C2Setting ?

field_3B8_C2_hostname4 C2Setting ?

field_3FC_C2Setting_URL1 db 128 dup(?)
field_47C_C2Setting_URL2 db 128 dup(?)
```

```
field_90C_install_option dd ?           ; XRE
                                        ; fun
field_910_install_folder_path dw 256 dup(?)
                                        ; XRE
                                        ; fun
field_B10_install_ServiceName dw 256 dup(?)
```

Type I: Downloading C2 Setting



- C2 Setting URL examples
  - http://dl.dropboxusercontent.com/s/eg3qusm8pl4iz49/index.txt
  - HTTP://TIEBA.BAIDU.COM/F?KZ=866965377
- Download data from the URL and decode a string between “DZKS” and "DZJS"

Type I: Debug Strings

- Some samples include debug string
  - file name
  - function name
  - version path
  - etc…
- Useful information for analysis

```
push    0Dh
push    31h
mov     eax, offset aXplgloader_cpp ; "XPlgLoader.cpp"
call    func_DebugPrintAndLogging ; STR: "file: %s, lin
```

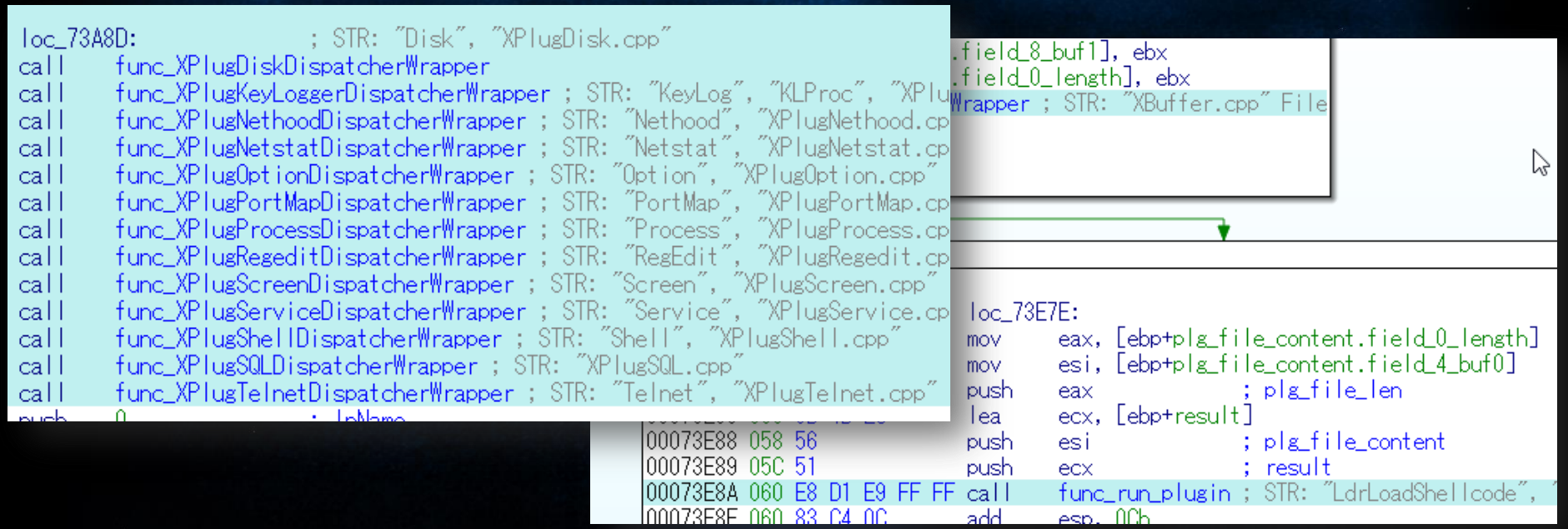
```
push    offset bootProc ; func_ptr
push    offset aBootproc ; "bootProc"
mov     ebx, offset hHandle ; a2
call    func_CreateThread_wrapper ; STR
```

Type I: Debug Strings (Version Path)

```
d:¥work¥plug3.0(gf)¥shellcode¥shellcode¥XSetting.h
d:¥work¥plug3.0(gf)udp¥shellcode¥shellcode¥XSetting.h
d:¥work¥plug3.0(lyt)¥shellcode¥shellcode¥XSetting.h
d:¥work¥plug3.0¥shellcode¥shellcode¥XSetting.h
d:¥work¥plug3.1(icesword)¥shellcode¥shellcode¥XSetting.h
d:¥work¥plug4.0(nvsmart)(sxl)¥shellcode¥shellcode¥XSetting.h
d:¥work¥plug4.0(nvsmart)(烟灰)(7.0)¥shellcode¥shellcode¥XSetting.h
d:¥work¥plug4.0(shellcode)(hrb)(gf)¥shellcode¥shellcode¥XSetting.h
d:¥work¥plug4.0(shellcode)(马甲)¥shellcode¥shellcode¥XSetting.h
d:¥work¥plug4.0(马甲)(定制版)¥shellcode¥shellcode¥XSetting.h
d:¥work¥plug6.0(360)(gadget)(小宏)¥shellcode¥shellcode¥XSetting.h
d:¥work¥plug6.0(360)(gadget)(非洲来客)(正式)¥shellcode¥shellcode¥XSetting.h
d:¥work¥plug6.0(360)(hkcmd)(xts)(scldr3.0)¥shellcode¥shellcode¥XSetting.h
d:¥work¥plug6.0(360)(mcinsupd)(封初)¥shellcode¥shellcode¥XSetting.h
d:¥work¥plug6.0(360)(mcoemcpy)(hhhtwy)(scldr3.0)¥shellcode¥shellcode¥XSetting.h
d:¥work¥plug6.0¥plug6.0(minidownloader)¥shellcode¥shellcode¥XSetting.h
d:¥work¥plug7.0(mcvsmap)(fking)(前锋)¥shellcode¥shellcode¥XSetting.h
d:¥work¥plug7.0(老罗)(3套)¥plug7.0(oleview)(老罗3)(协议变体)¥shellcode¥shellcode¥XSetting.h
d:¥work¥plug8.0(mcoemcpy)(lyt)¥shellcode¥shellcode¥XSetting.h
d:¥work¥scontroller(3.0)(超哥)(demo)(马甲)¥shellcode¥shellcode¥XSetting.h
h:¥fast¥plug(dnsbench)¥shellcode¥shellcode¥XSetting.h
```

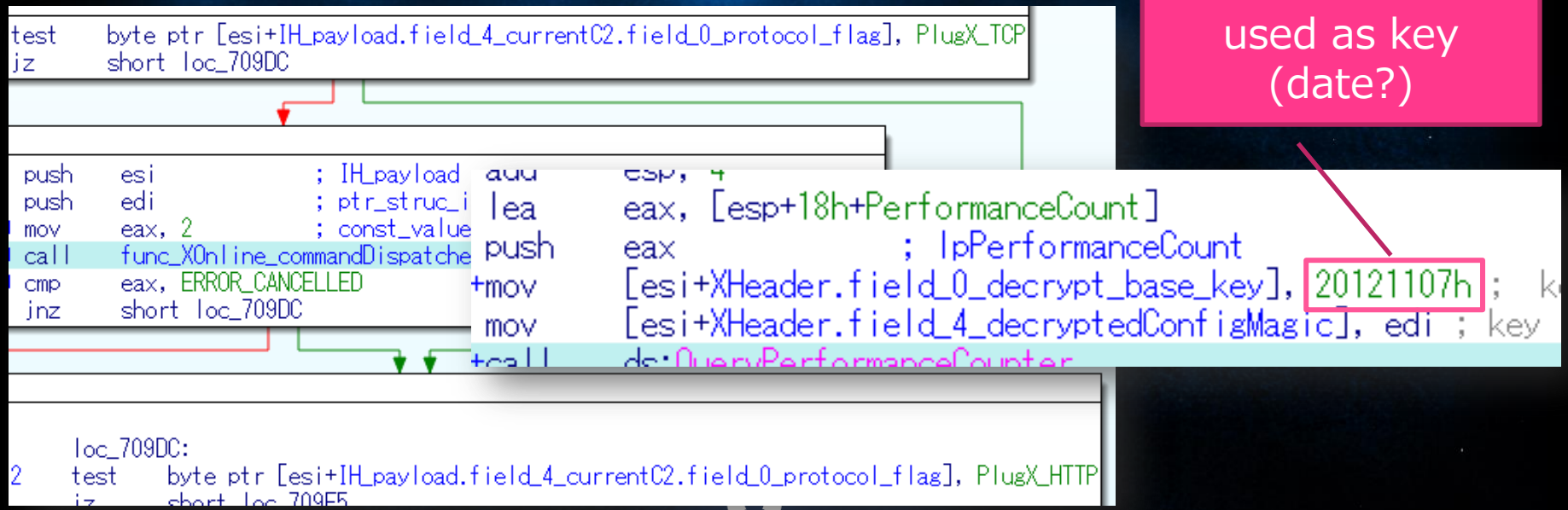
Type I: Functions and Plugins

- Functions
  - File/Registry operations
  - Keylogging, screenshot, remote shell
  - portscan, SQL command, etc…
- Plugins
  - install path + %d.plg (use LdrLoadShellcode to load)



Type I: Protocols

- Supported Protocols
  - TCP, HTTP, UDP, ICMP (not implemented)
- Traffic data is encrypted
  - the same algorithm as PlugX Payload



dword value used as key (date?)

Type I: Bypassing UAC Dialog

- For default UAC setting Win7 machines (ConsentPromptBehaviorAdmin != 2)
  - PlugX: create msiexec.exe process and inject code to it
  - msiexec.exe: abuse DLL load order to execute malicious code in sysprep.exe process
  - sysprep.exe (privileged): run PlugX again

```
mov     eax, [esp+0A8h+IFileOperation_ppv]
mov     ecx, [eax]
mov     edx, [ecx+IFileOperationVtbl.SetOperationFlags] ; (This,dwOperationFlags)
push    10840014h       ; (FOF_NOCONFIRMATION|FOF_SILENT|FOFX_SHOWELEVATIONPROMPT|FOFX_NOCOPYHOOKS|FOFX_REQUIREELEVATION|F
push    eax
call    edx             ; SetOperationFlags
```

Type II: Summary

- Observed since 2013/3Q
- The differences
  - Anti-reversing/forensic techniques
  - Encryption algorithm changed
  - Config extended
  - More protocols
    - ICMP
    - DNS [5] (not sure)
  - More functions
    - e.g., network sniffing to steal proxy auth info
  - Debug strings suppressed

Type II: Anti-reversing/forensic techniques

- PlugX payload MZ/PE signatures replaced by “XV”
- magic “GULP” in encrypted data table eliminated
- string obfuscation (decode/clear)
- massive API wrappers added for anti code diffing

similarity	confidence	change	EA primary	name primary	EA secondary	name secondary
0.40	0.76	GI--E-C	009A26B1	func_GetTokenInformati···	00081740	sub_81740_1240
0.40	0.59	GI--E-C	009A6E3A	sub_9A6E3A_1044	00087880	func_CreateThread_for_key_function
0.39	0.45	-I--E--	0099C6C8	func_RegEnumKeyExW_···	0007A520	Reg_sub_1000FA50
0.39	0.57	GI--ELC	00995955	func_C2_multiple_protoc···	00070980	func_C2_multiple_protocol_communi···

Type I

```
call    ds:[EnterCriticalSection]
lea     eax, ds:[edi+0x18]
mov     ds:[esi], eax
mov     edx, ds:[eax+4]
mov     ds:[esi+4], edx
mov     ecx, ds:[eax+4]
mov     ds:[ecx], esi
mov     ds:[eax+4], esi
inc     ds:[eax+8]
push    edi
call    ds:[LeaveCriticalSection]
mov     edx, ds:[ebx]
push    edx
call    ds:[ResumeThread]
test    eax, eax
```

Type II

```
call    func_EnterCriticalSection_Wrapper
lea     eax, ds:[edi+0x18]
mov     ds:[esi], eax
mov     ecx, ds:[eax+4]
mov     ds:[esi+4], ecx
mov     ecx, ds:[eax+4]
mov     ds:[ecx], esi
mov     ds:[eax+4], esi
inc     ds:[eax+8]
push    edi
call    func_LeaveCriticalSection_Wrapper
push    ds:[ebx]
call    func_ResumeThread_Wrapper
```

Type II: Encryption algorithm changed

Type I

```
if ( buf_len > 0 )
{
  dwBufferOffset = lpEncryptBuf - v4;
  while ( 1 )
  {
    a1 = a1 + (a1 >> 3) - 0x11111111;
    v5 = v5 + (v5 >> 5) - 0x22222222;
    decrypt_key3 += 0x44444444 - (decrypt_key3 << 9);
    lpEncryptBufa = 0x33333333 - (v6 << 7) + v6;
    v7 = *(_BYTE *)(dwBufferOffset + v4++) ^ (decrypt_key3 + lpEncryptBufa + v5 + a1);
    v8 = buf_len-- == 1;
    *(_BYTE *)(v4 - 1) = v7;
    if ( v8 )
      break;
    v6 = lpEncryptBufa;
  }
}
```

Type II

```
if ( bufsize > 0 )
{
  offset = ptr_encrypted_buf - ptr_decrypted_buf;
  do
  {
    xor_key = (xor_key << 7) - (xor_key >> 3) + i + 0x713A8FC1;
    ptr_current_decrypted_config_array = &ptr_decrypted_buf[i];
    decrypted_byte = xor_key ^ BYTE1(xor_key) ^ (xor_key >> 16) ^ *(&ptr_decrypted_buf[i++] + offset) ^ BYTE3(xor_key);
    *ptr_current_decrypted_config_array = decrypted_byte;
  }
  while ( i < bufsize );
}
```

Type II: Config Extended

- magic word for message authentication
- multiple code injection targets (1 -> 4)
- configuration for screenshots

```
field_1D18_receive_magic_word dw 256 dup(?)
                                        ; XREF: func_XOnline_Command0_CollenctMac
                                        ; func_initConfig+262↑o ... ; "fuck"
field_1F18_send_magic_word dw 256 dup(?)
                                        ; XREF: func_XOnline_Command0_CollenctMac
                                        ; func_initConfig+28A↑o ... ; "taipei"
```

```
injection target: %windir%¥system32¥winlogon.exe↓
injection target2: %windir%¥explorer.exe↓
injection target3: %windir%¥system32¥svchost.exe↓
injection target4: %ProgramFiles%¥Internet Explorer¥iexplore.exe↓
```

```
                                        ; func_initConfig+1E9↑w ...
field_232C_SC_ExpireDays dd ?           ; XREF: func_XScreen_SC+A7↑r
                                        ; func_XScreen_SC+F2↑r ...
field_2330_screenshots_folder_path dw 256 dup(?) ; XREF: func_XScreen_SC+9C↑o
                                        ; func_initConfig+232↑o ... ; %AUTO%¥screen
```

Type III: Summary

- Observed since mid 2012
  - It may be ancestor of Type I & II, but still evolving
- The differences
  - Very different implementation from Type I & II
    - no PlugX Payload, no encrypted data table
  - More advanced anti-reversing
    - string obfuscation (200->1000)
    - code obfuscation
  - Config shrunk
    - C2 Setting/C2 Setting URL/Proxy Setting merged
    - Recently Type II members (e.g., magic word) added
  - TCP & HTTP supported

Type III: Code Obfuscation

```
int __thiscall func_copyString(struc_string *struc_string, int offset, unsigned
{
  *ptrSize = size;
  func_strcpy(offset + struc_string->field_0_ptr_string, srcString, *ptrSize);
  return 0;
}
```

junk code inserted

```
 push    ebp
 mov     ebp, esp
 sub     esp, 8
+mov     edx, ds:GetShortPathNameA
 mov     eax, 0B50Ah
 mov     [ebp+var_4], ax
 mov     [ebp+var_8], edx
 movzx   eax, [ebp+var_4]
 mov     edx, [ebp+size]
+imul    eax, 1530h
 mov     [ebp+var_4], ax
 shr     [ebp+var_4], 3
 push    esi
 mov     esi, [ebp+ptrSize]
 mov     [esi], edx
 movzx   edx, [ebp+var_4]
 mov     eax, 49A86EABh
 imul    edx
 sar     edx, 0Eh
 mov     eax, edx
 shr     eax, 1Fh
 add     eax, edx
 mov     [ebp+var_4], ax
 movzx   edx, [ebp+var_4]
+and     edx, 2633h
 mov     [ebp+var_4], dx
 movzx   eax, [ebp+var_4]
 mov     edx, [esi]
+imul    eax, 0FA2Ch
 mov     ecx, [ecx+struc_string.field_0_ptr_string]
 add     ecx, [ebp+offset]
 mov     [ebp+var_4], ax
 mov     eax, [ebp+srcString]
 push    edx             ; size
 push    eax             ; srcString
 push    ecx             ; dstString
 call    func_strcpy
 movzx   edx, [ebp+var_4]
 add     esp, 0Ch
+or      edx, 68C2h
 mov     [ebp+var_4], dx
 xor     eax, eax
 pop     esi
 mov     esp, ebp
 pop     ebp
 retn    10h
```

Type III: “DESTORY”?

- Type III checks the consistency of its decrypted config
  - If not successful, pop up a message
  - The correct spelling seems to be “DESTROY” ☺

```
push    1
push    offset asc_1008DCC8 ; "CONFIG-DESTORY!"
push    0A61CA3C6h          ; key
push    10h                 ; data_len
push    offset aSojo7X3CIs ; "sO¥t7ｵｼｼ邊甥ﾘ"
+lea    ecx, [ebp+var_84] ; struc_decoded_string
call    func_decode_string ; decode "CONFIG-DESTORY!"
mov     ecx, [eax]
push    200040h             ; uType
push    0                   ; lpCaption
push    ecx                 ; lpText
push    0                   ; hWnd
call    MessageBoxA_0       ; <"MessageBoxA">
+lea    ecx, [ebp+var_84] ; this
call    func_erase_string
```

	Type I	Type II	Type III
executable loading PlugX	signed exe	signed exe	rundll32
encrypted data table	Yes	Yes (no “GULP”)	No
protocols	TCP,HTTP,UDP	TCP,HTTP,UDP, ICMP,DNS?	TCP,HTTP
Functions compared with Type I	-	more	much more
Debug Strings (Version Path)	sometimes	rarely	not at all
string obfuscation	No	Yes	Yes (much more)
code obfuscation	No	No	Yes
date dword value	20120123, 20121016, 20121107,	20130810	20100921, 20111011

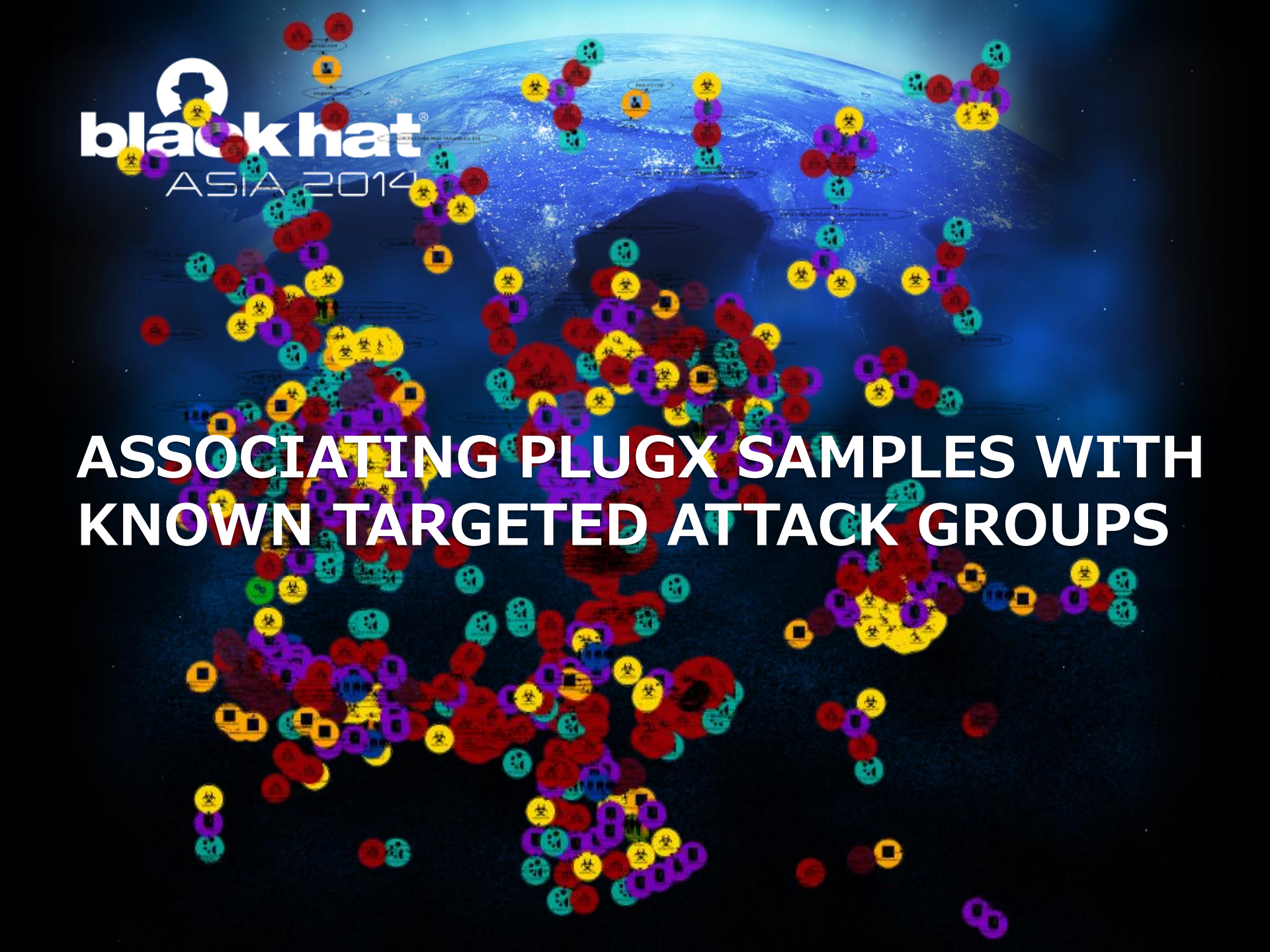
PARSING PLUGX CONFIG

Related Works

- "PlugXExtract" for Type I [6]
  - decrypt, but not parse
- "plugxdecoder" for Type I [7]
  - decode PlugX traffic then extract encrypted artifacts (not parse)
- Volatility Plugin for Type I & II [5]
  - decrypt then parse
    - supported 6 config size versions
      - 0xbe4, 0x150c, 0x1510, 0x1b18, 0x1d18, 0x2540

Implementation

- Type I & II
  - Immunity Debugger script
    - Search encrypted data table in injected process
      - "GULP" signature for Type I
      - PIC (Position Independent Code) for Type II
    - Decrypt (&decompress if needed) config/original PE
    - Parse config
      - supported: 0x150C, 0x1510, 0x1B18, 0x1D18, 0x2540
- Type III
  - IDAPython script
    - Deobfuscate strings and identify config decryption routine
      - You need to specify the deobfuscation function ☹
    - Decrypt and parse config
      - supported: 0x72C, 0x76C, 0x7AC, 0x840, 0xDF0



ASSOCIATING PLUGX SAMPLES WITH KNOWN TARGETED ATTACK GROUPS

Summary

- We analyzed over 150 samples
  - About 30 samples do not contain configurations.
    - These are “DEMO” version.
  - So we classified other 123 samples at this time.
- Procedure
  - Categorize samples into some groups
    1. Make groups based on service name (registry value)
       - except default service name (SxS, XXX, TVT)
       - 1 group includes at least 4 samples
    2. Merge groups by using following information
       - C2 (IP address, hostname, domain, owner email)
       - debug string (version path), date dword
       - etc…
  - Check relations between the groups and published targeted attack reports
    - Are they connected to famous targeted attack groups?

The Main Source of the Intelligence for the C2 Connection

- VirusTotal passive DNS
  - https://www.virustotal.com/en/documentation/searching/
- AlienVault Open Threat Exchange
  - http://www.alienvault.com/open-threat-exchange
- TEAM CYMRU IP to ASN Mapping
  - https://www.team-cymru.org/Services/ip-to-asn.html
- Many targeted attack reports [8]-[20]
- Our internal Passive DNS and IP2DNS system
- dig, nslookup, whois command
  - And other whois sites

CLASSIFYING PLUGX SAMPLES

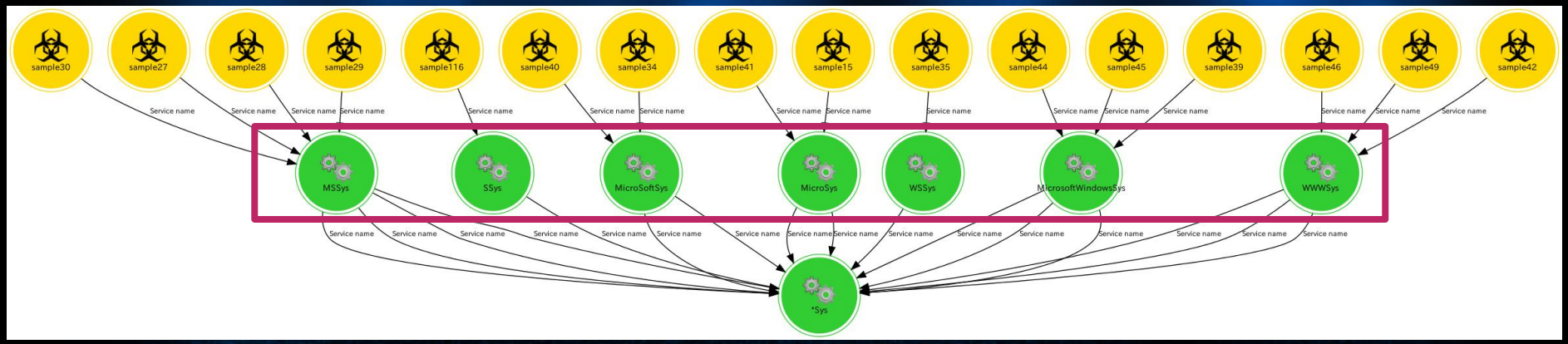
Summary of PlugX Groups

- We found seven groups.
  - *Sys group
  - *Http group
  - Starter group
  - graphedt group
  - WS group
  - 360 group
  - cochin group


black hat
ASIA 2014
*SYS GROUP

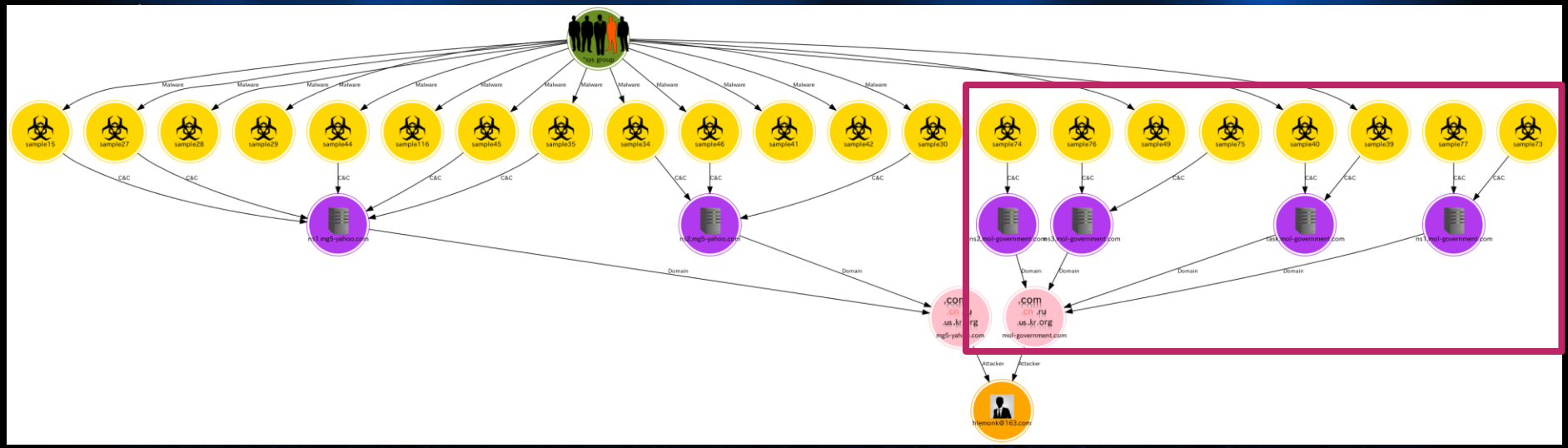
*Sys Group

- Some samples use "*Sys" as the service name.



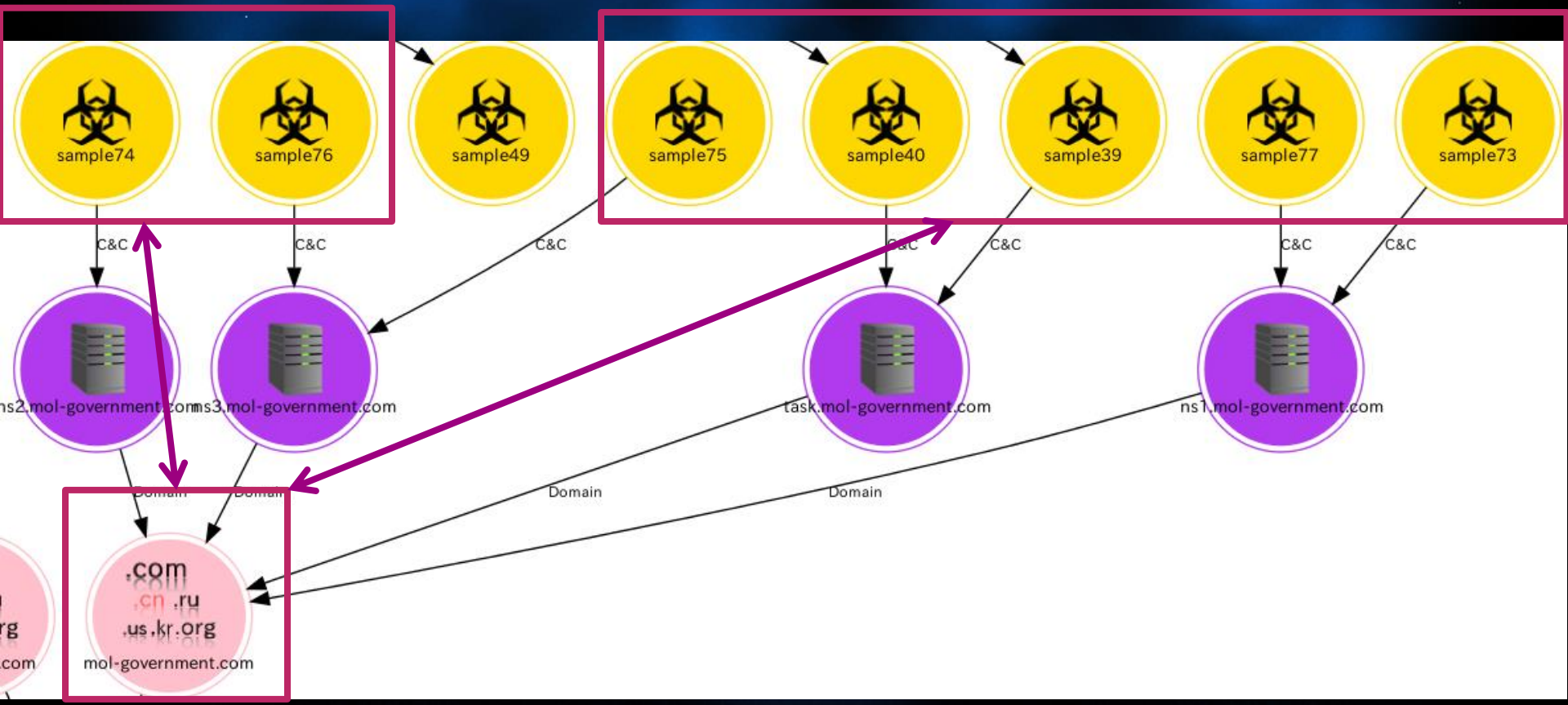
*Sys Group

- Some another samples share the domain of "*Sys" group samples.



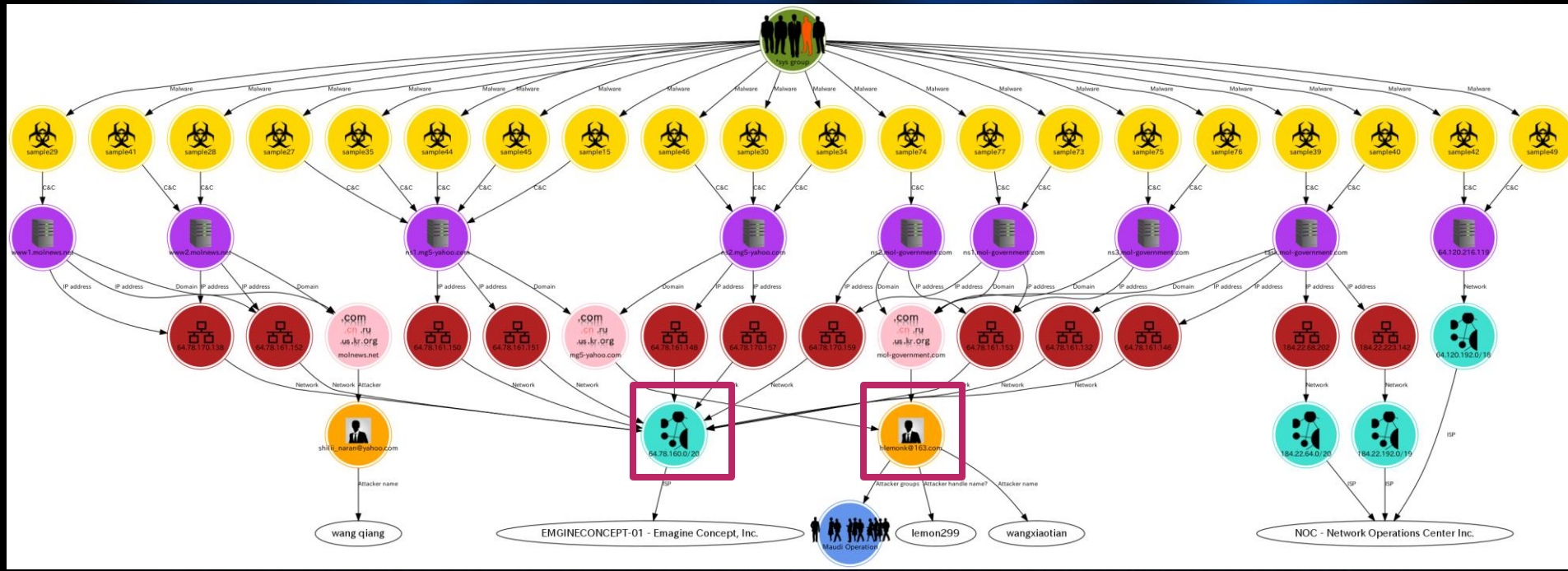
*Sys Group

- Some another samples share the domain of "*Sys" group samples.



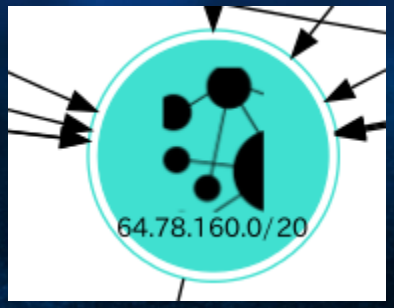
*Sys Group

- This is the overall "*Sys" group.



*Sys Group

- Most samples of "*Sys" group …
  - use the same network.

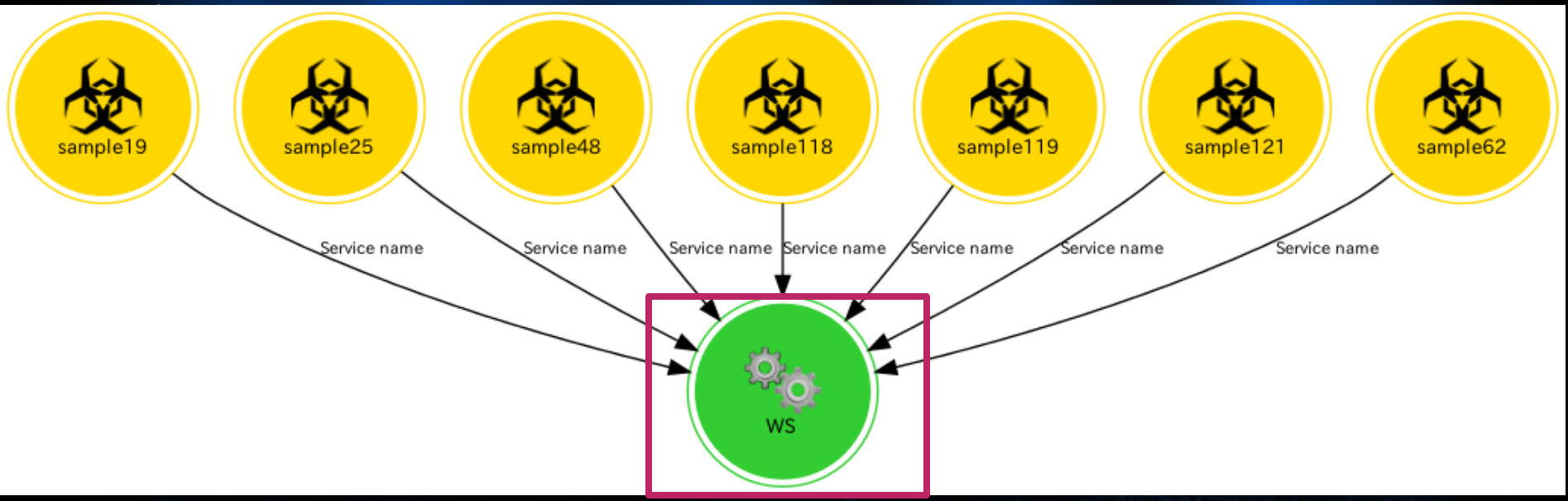


  - C2s are managed by the same domain owner.

WS GROUP

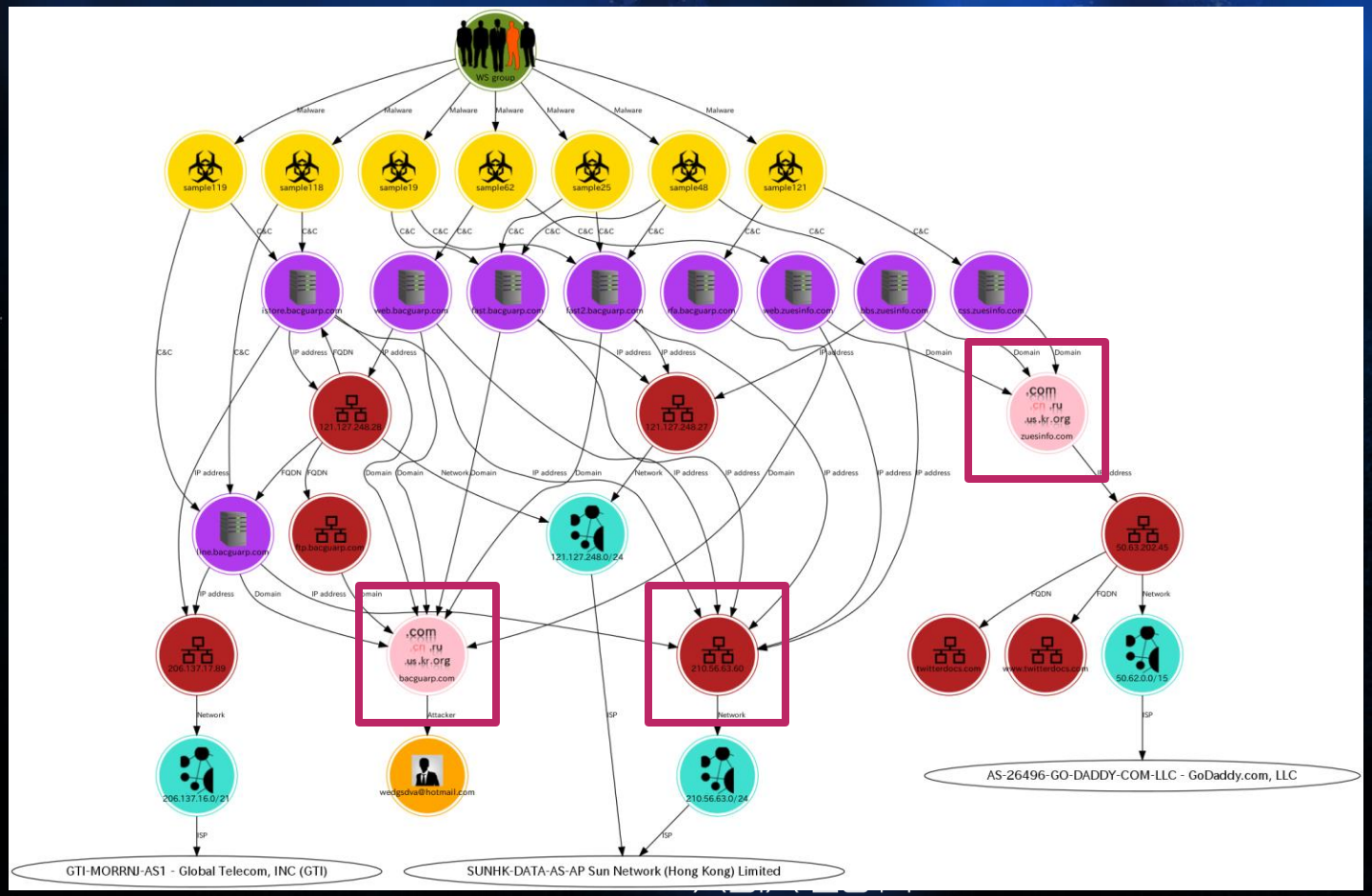
WS Group

- Some samples use “WS” as the service name.



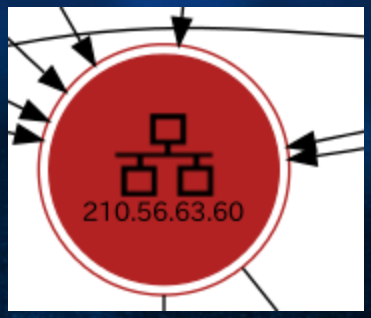
WS Group

- This is the overall “WS” group.



WS Group

- Most samples of “WS” group use the same IP address.

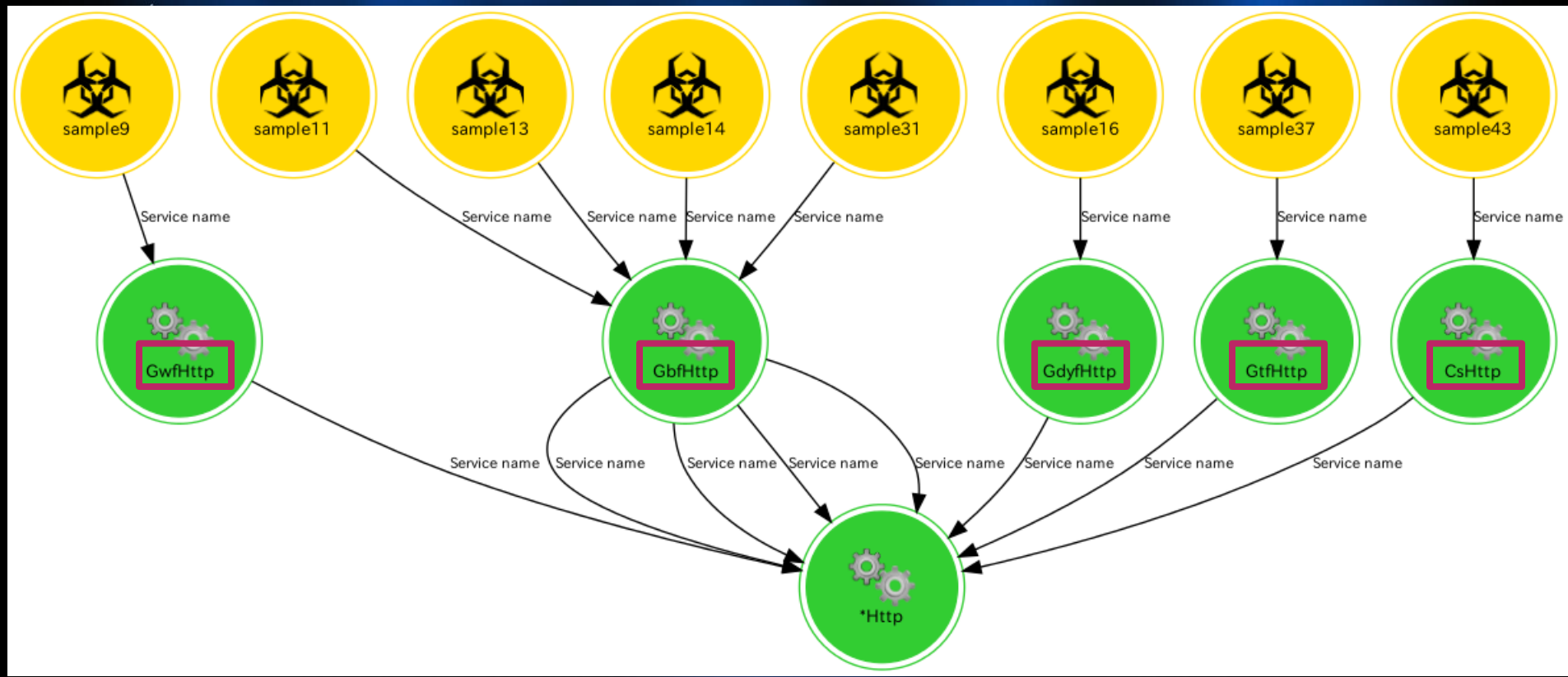


  - All samples of the “WS” group share these domains as the C2s.

*HTTP GROUP

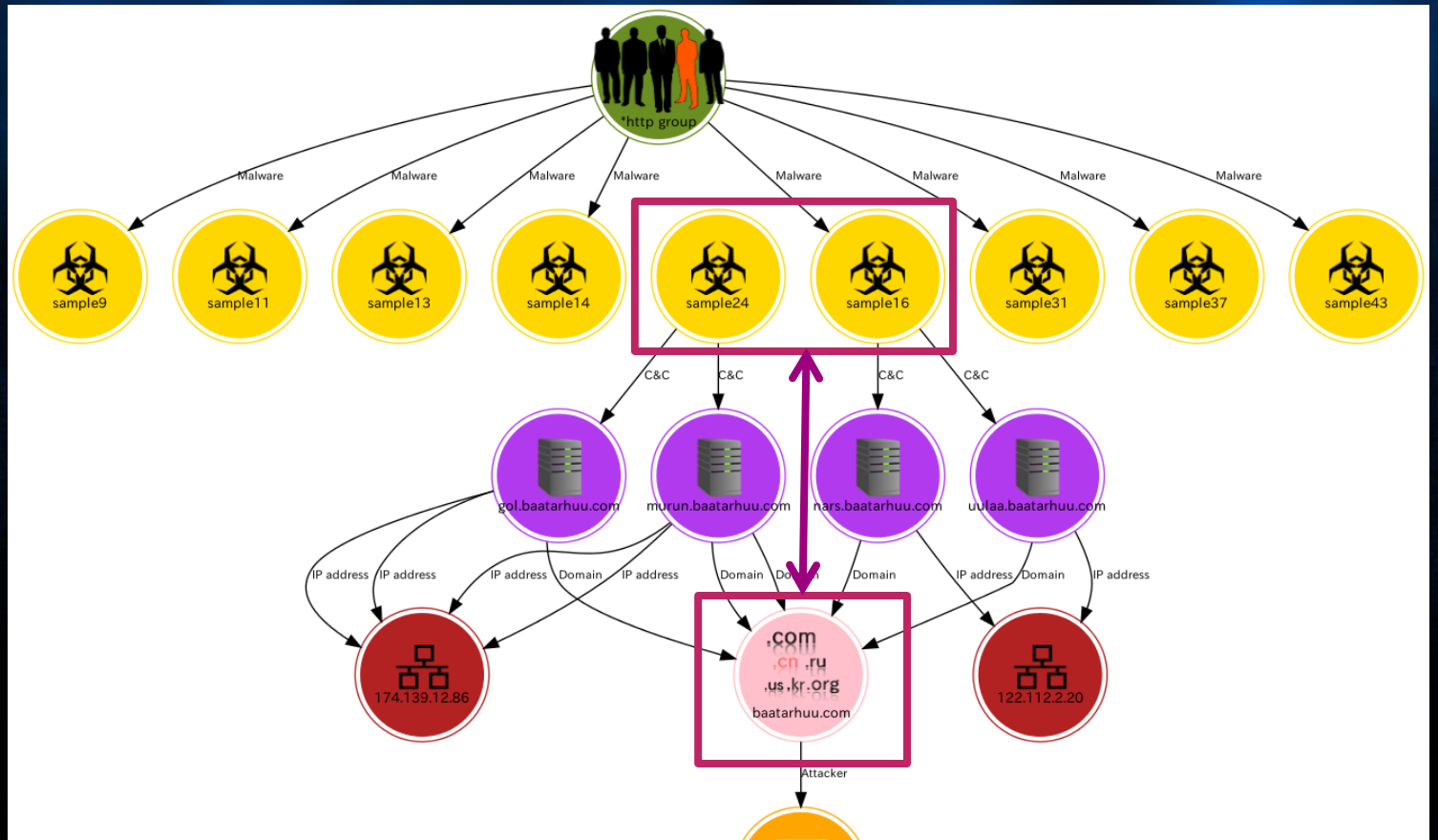
*Http Group

- Some samples use “*Http” as the service name.



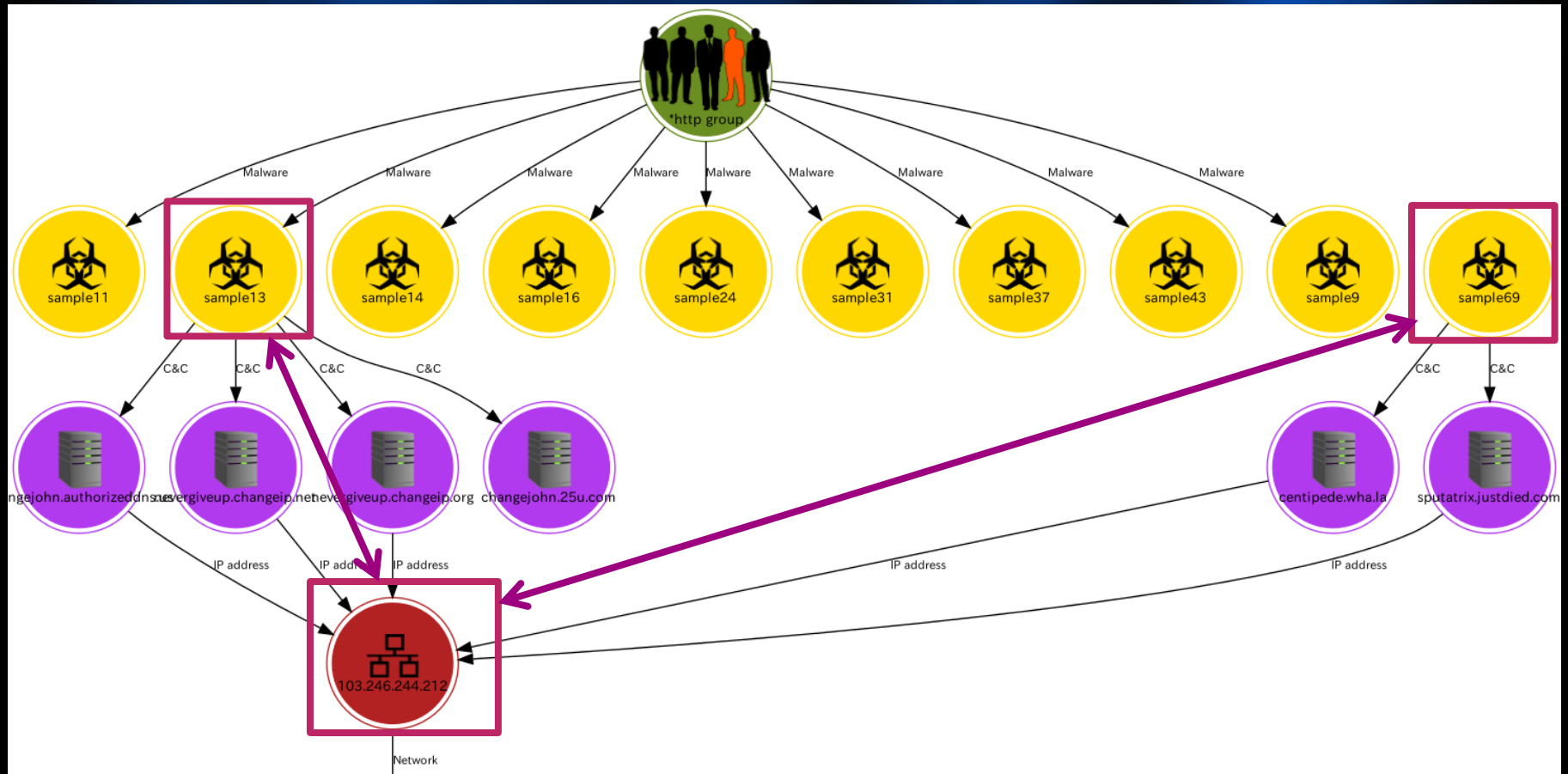
*Http Group

- Sample 16 and sample 24 use the same domain.



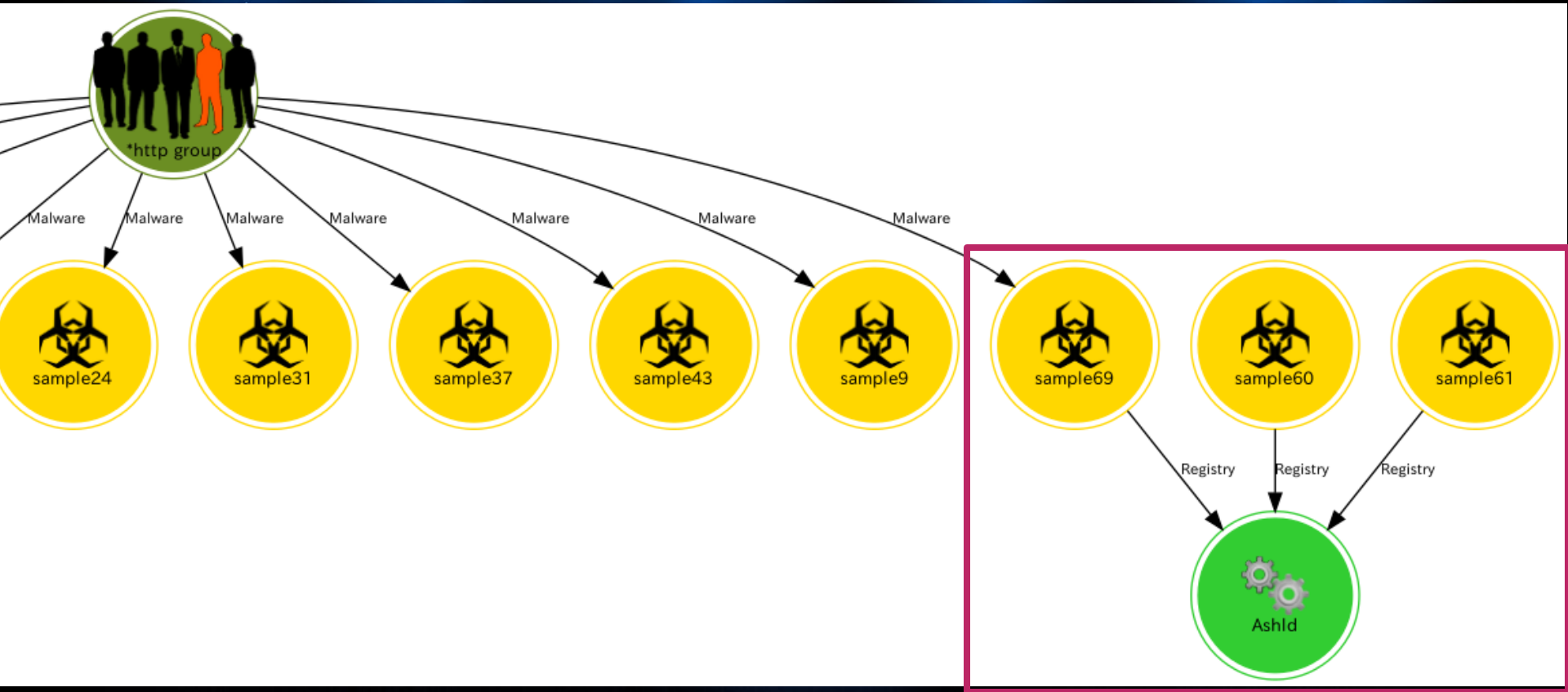
*Http Group

- Sample 13 and sample 69 use the same IP address.



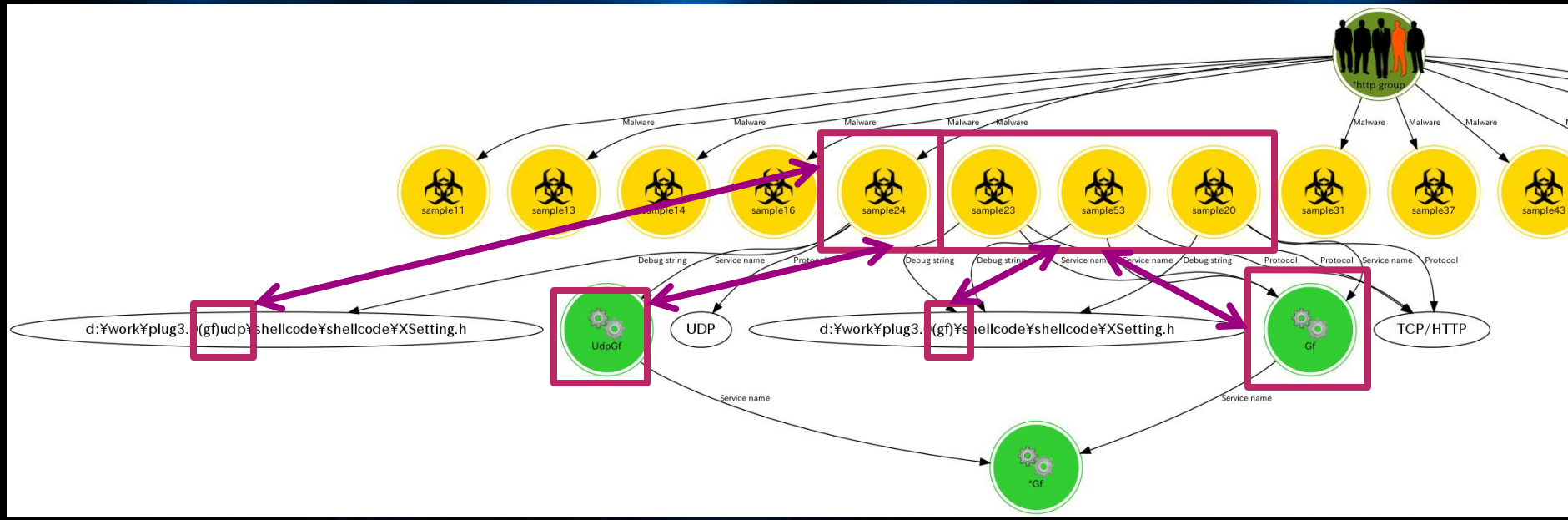
*Http Group

- Sample 69 and sample 60, 61 have the same registry value.



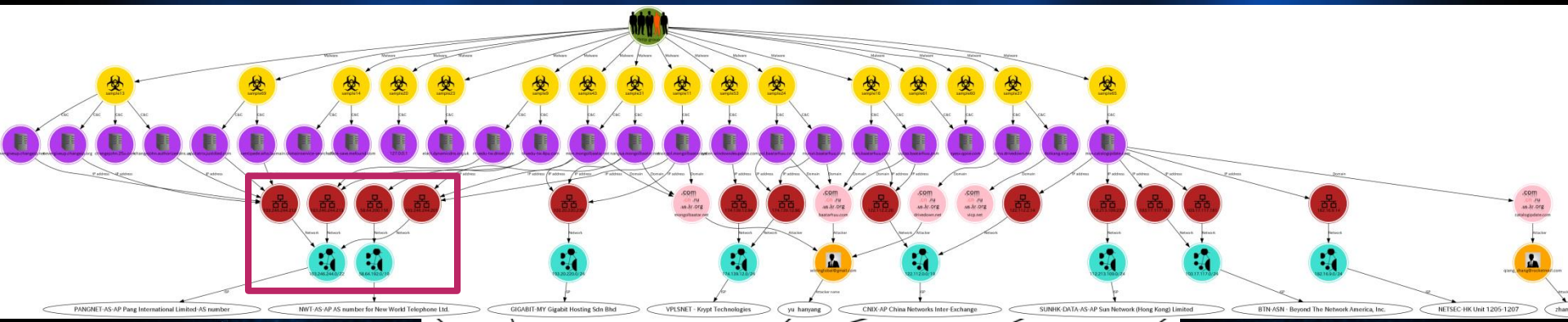
*Http Group

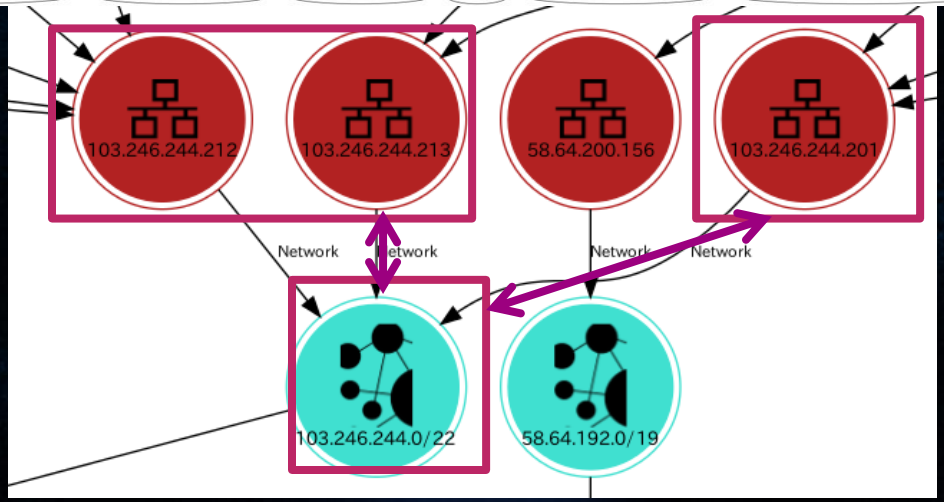
- Sample 24 and sample 23, 53, 20 use "*Gf" value as the service name.
  - And similar debug strings.



*Http Group

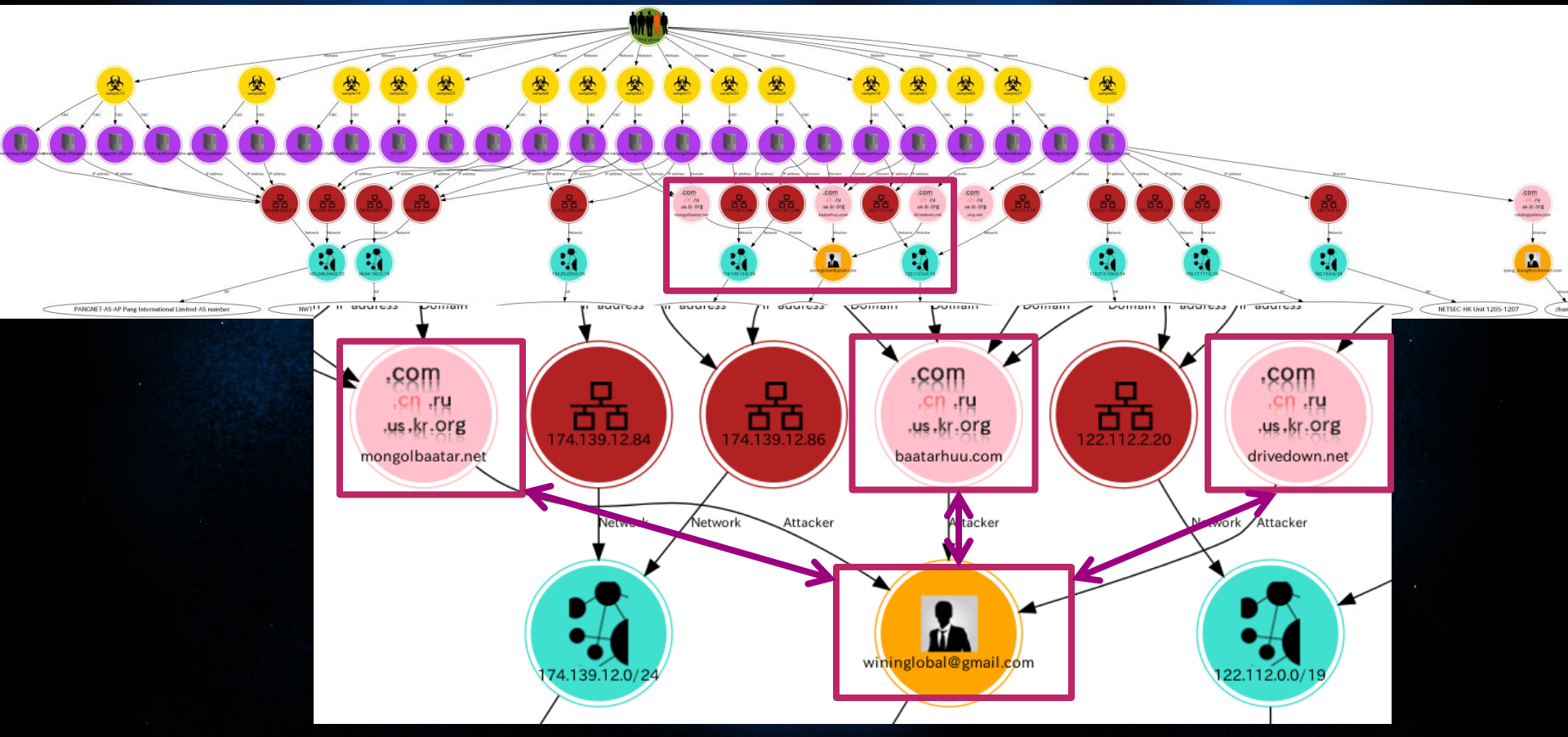
- Characteristics of the "*Http" group
  - Some C2s are connected to the same network.





*Http Group

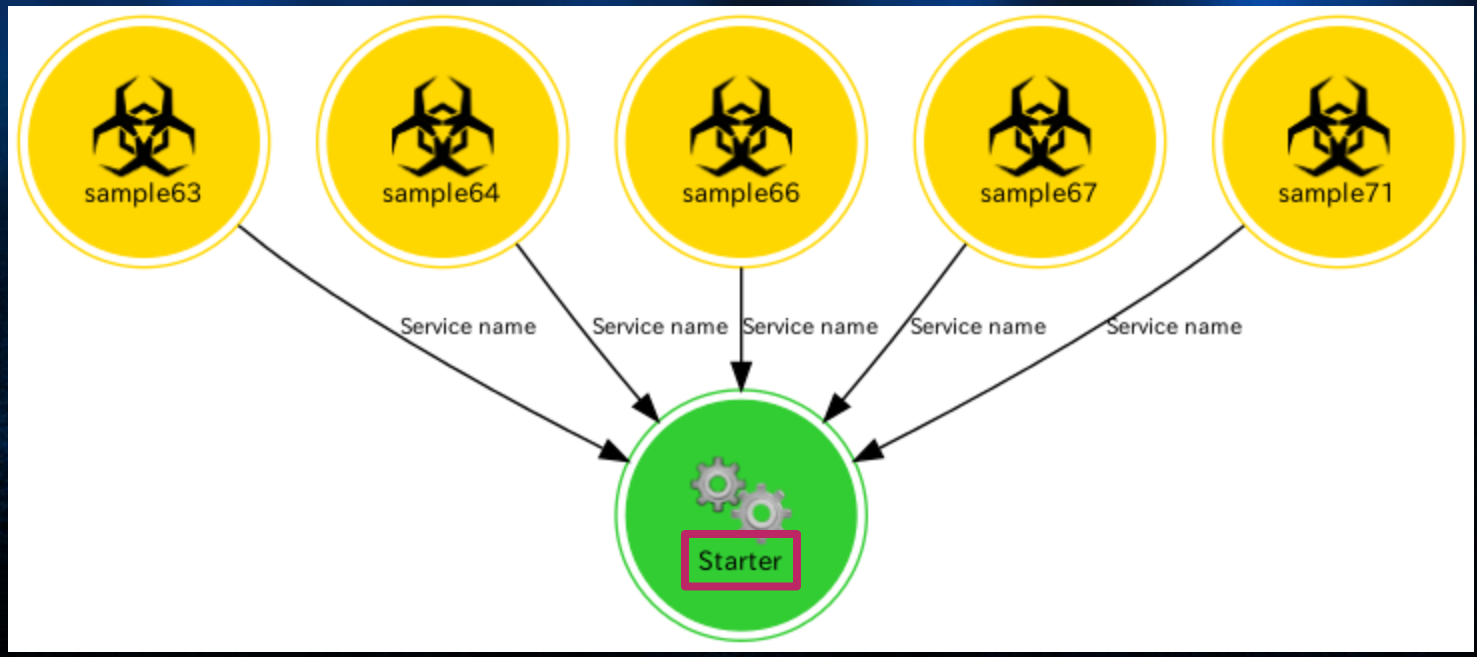
- Characteristics of the “*Http” group
  - Some domains are managed by the same owner.



STARTER GROUP

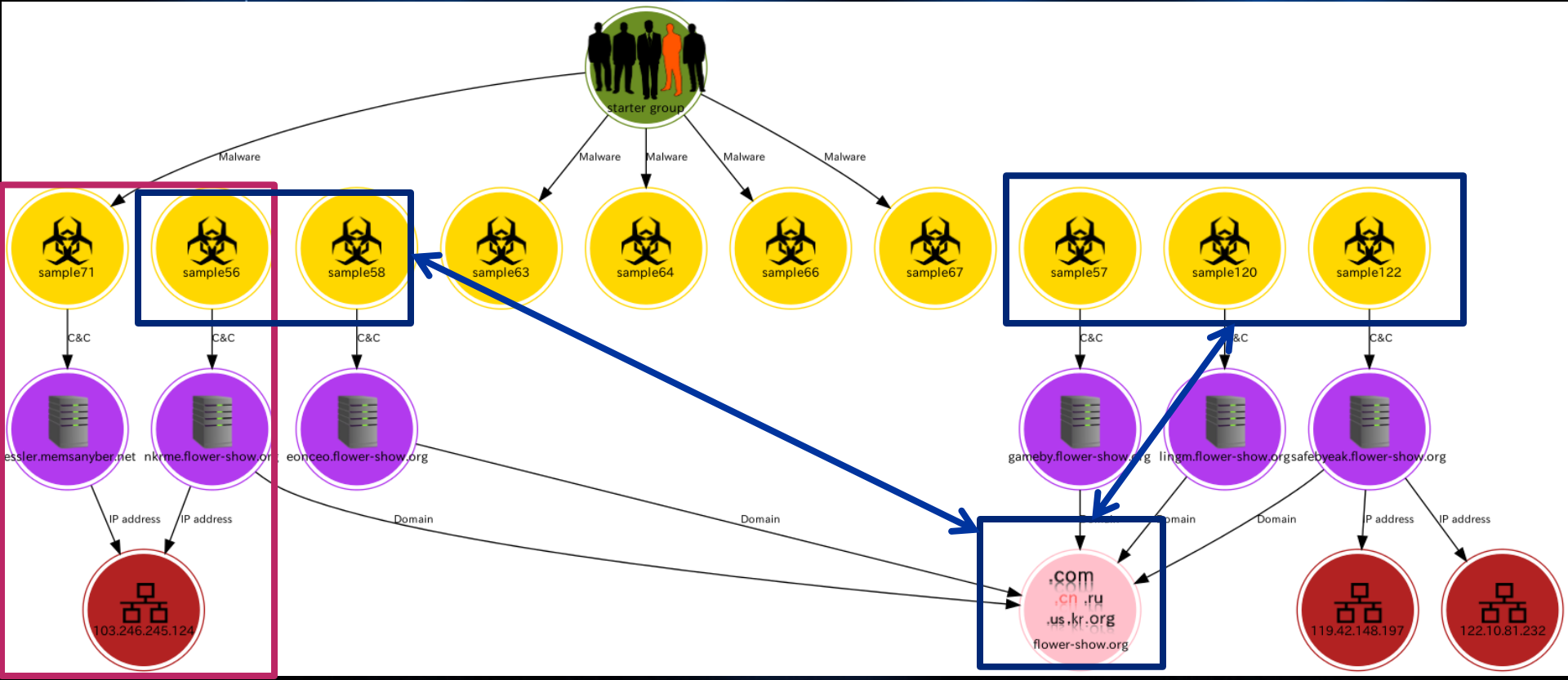
Starter Group

- Some samples use “Starter” as the service name.



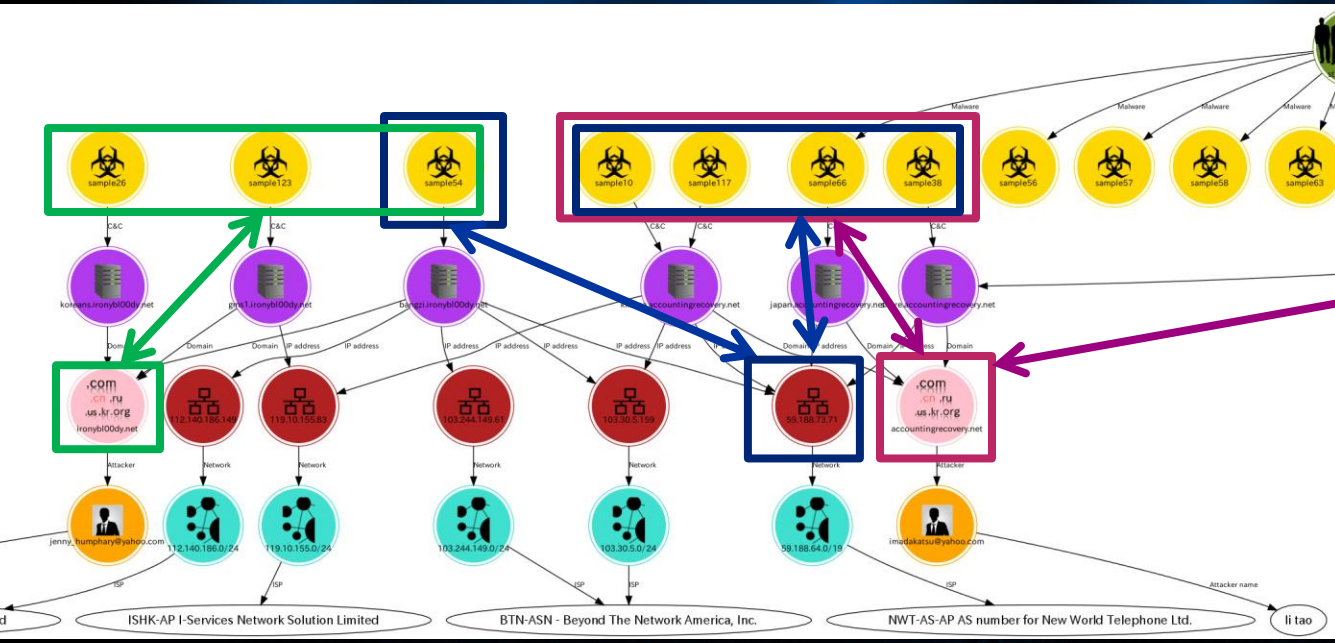
Starter Group

- Sample 71 and 56 use the same IP address.
- Sample 56 and several samples use the same domain.



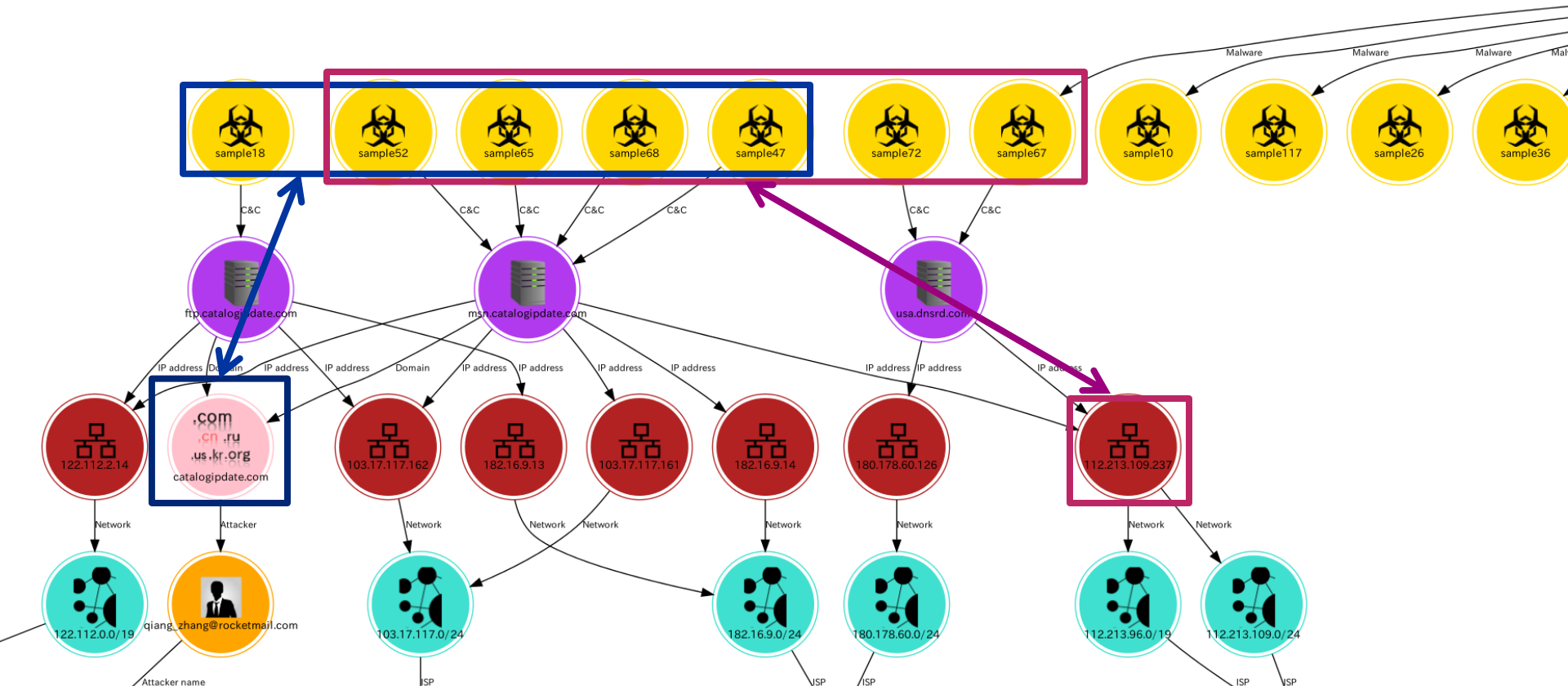
Starter Group

- Sample 66 and several samples use the same domain.
- Sample 66 and other ones use the same IP address.
- Sample 26, 123 and 54 use the same domain.



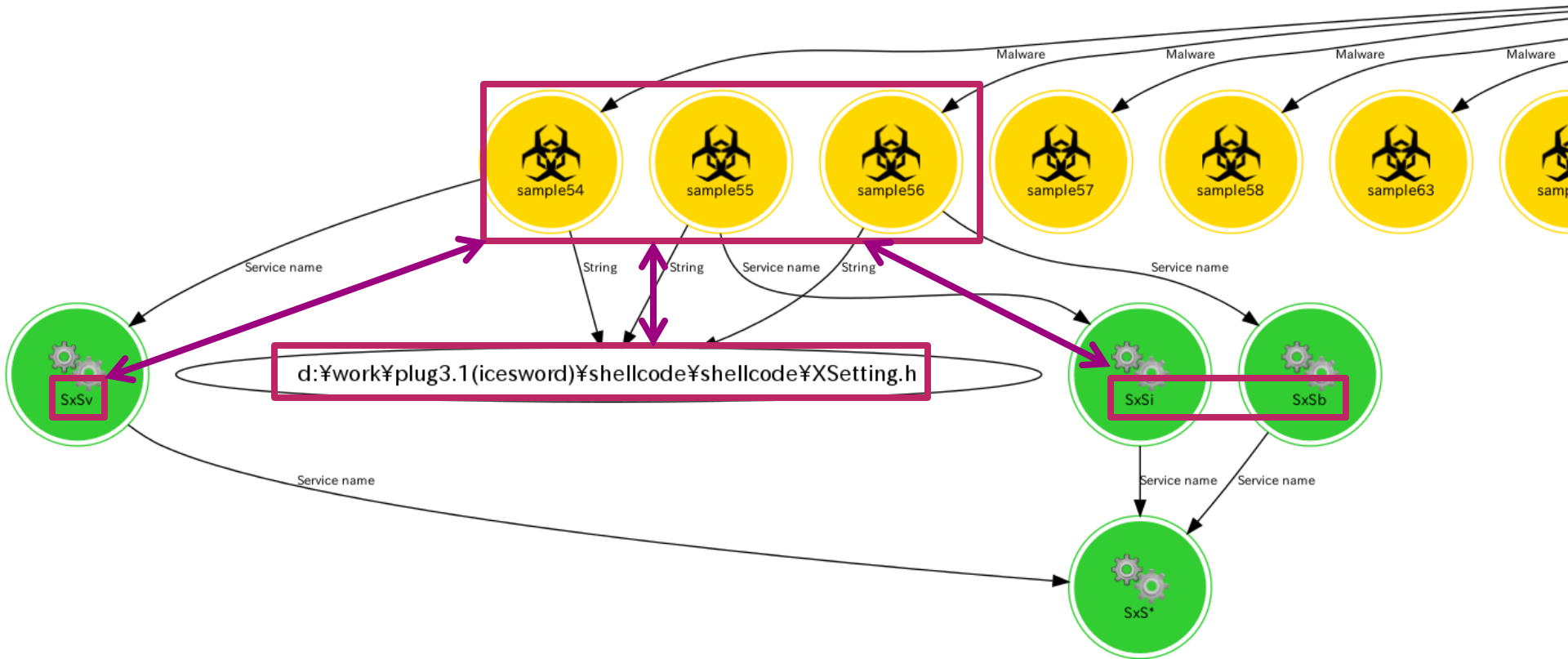
Starter Group

- Sample 67 and several samples use the same IP address.
- Sample 52 and 18 use the same domain.



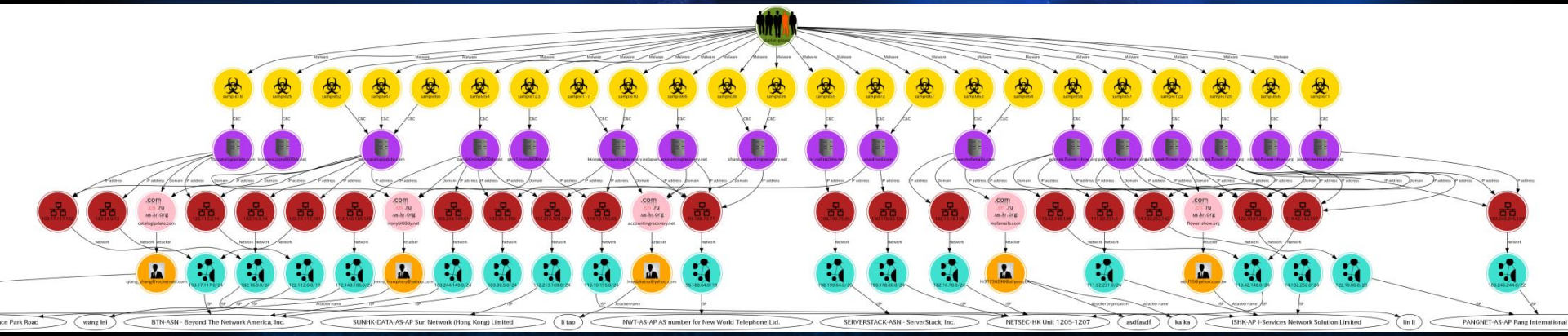
Starter Group

- Sample54, 55 and 56 use “SxS[a-z]” as the service name.
  - And these samples use the same debug string.



Starter Group

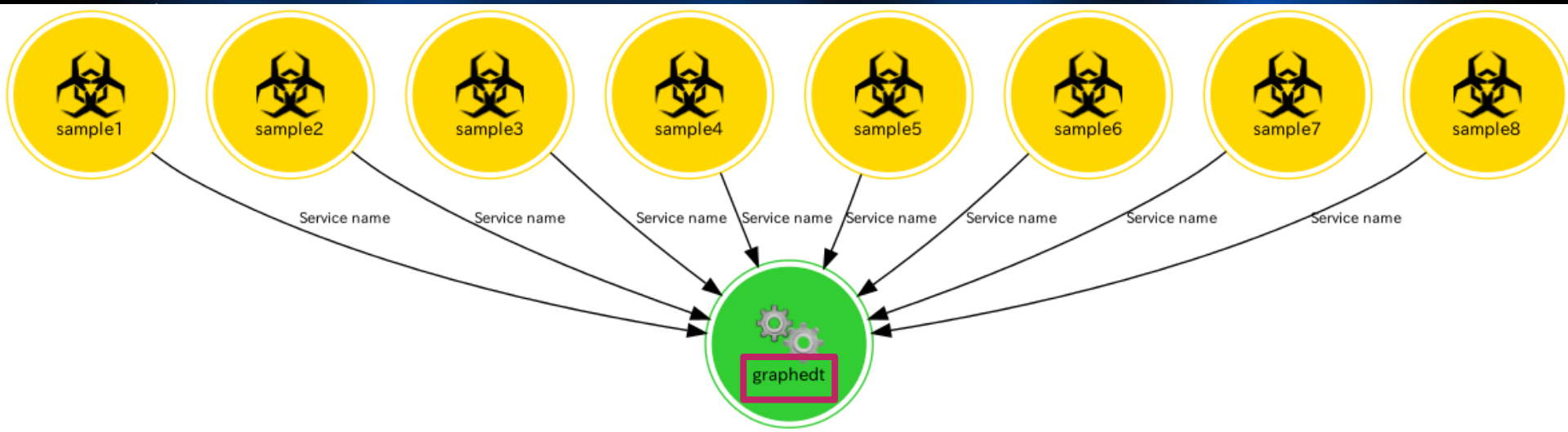
- This is the overall “Starter” group.



GRAPHEDT GROUP

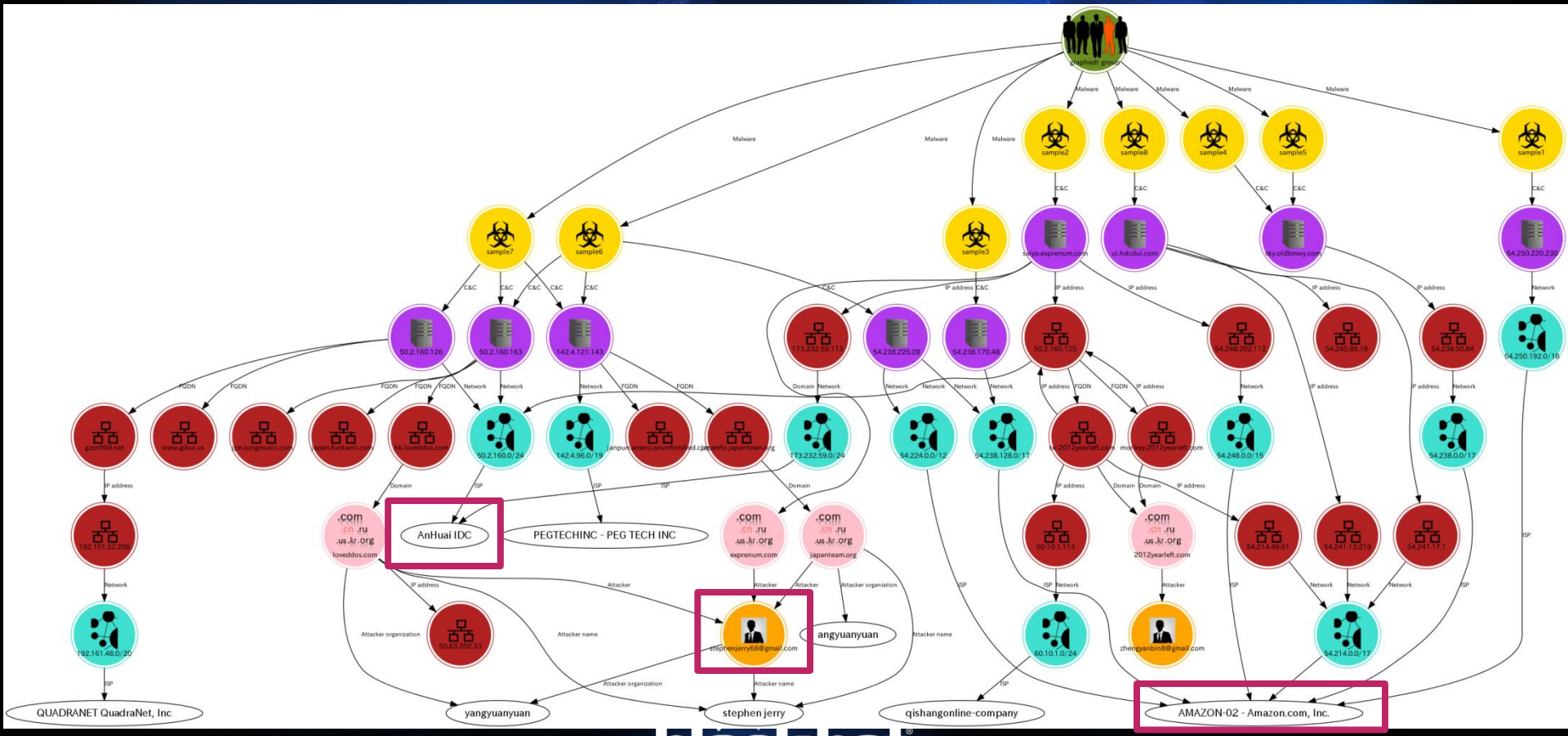
Graphedt Group

- Some samples use “graphedt” as the service name.



Graphedt Group

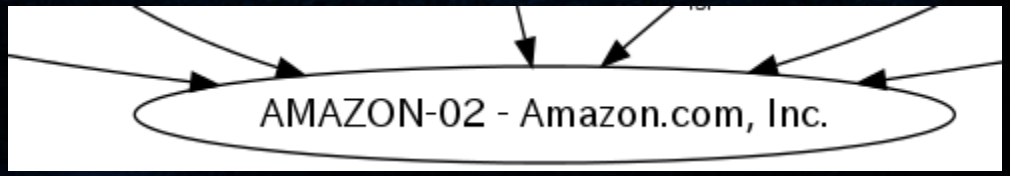
- This is the overall “graphedt” group.



Graphedt Group

- Graphedt group characteristics
  - Most samples of the Graphedt group …
    - Some domains are managed by the same owner.

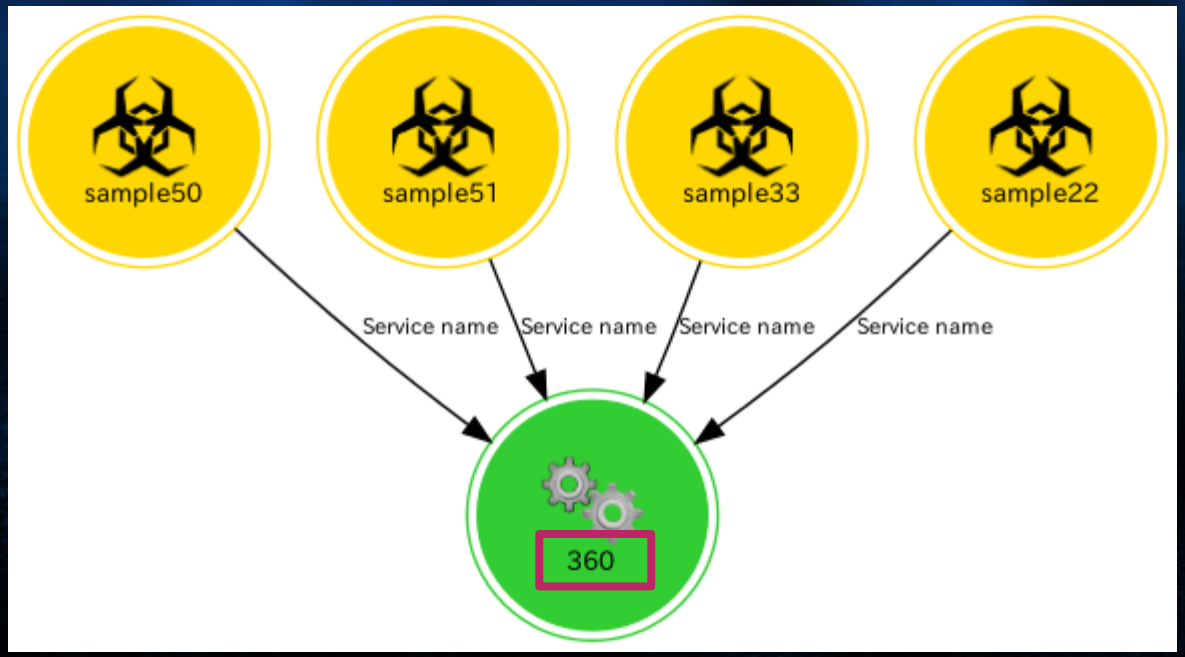
    - The C2s belong to these networks.




black hat
ASIA 2014
360 GROUP

360 Group

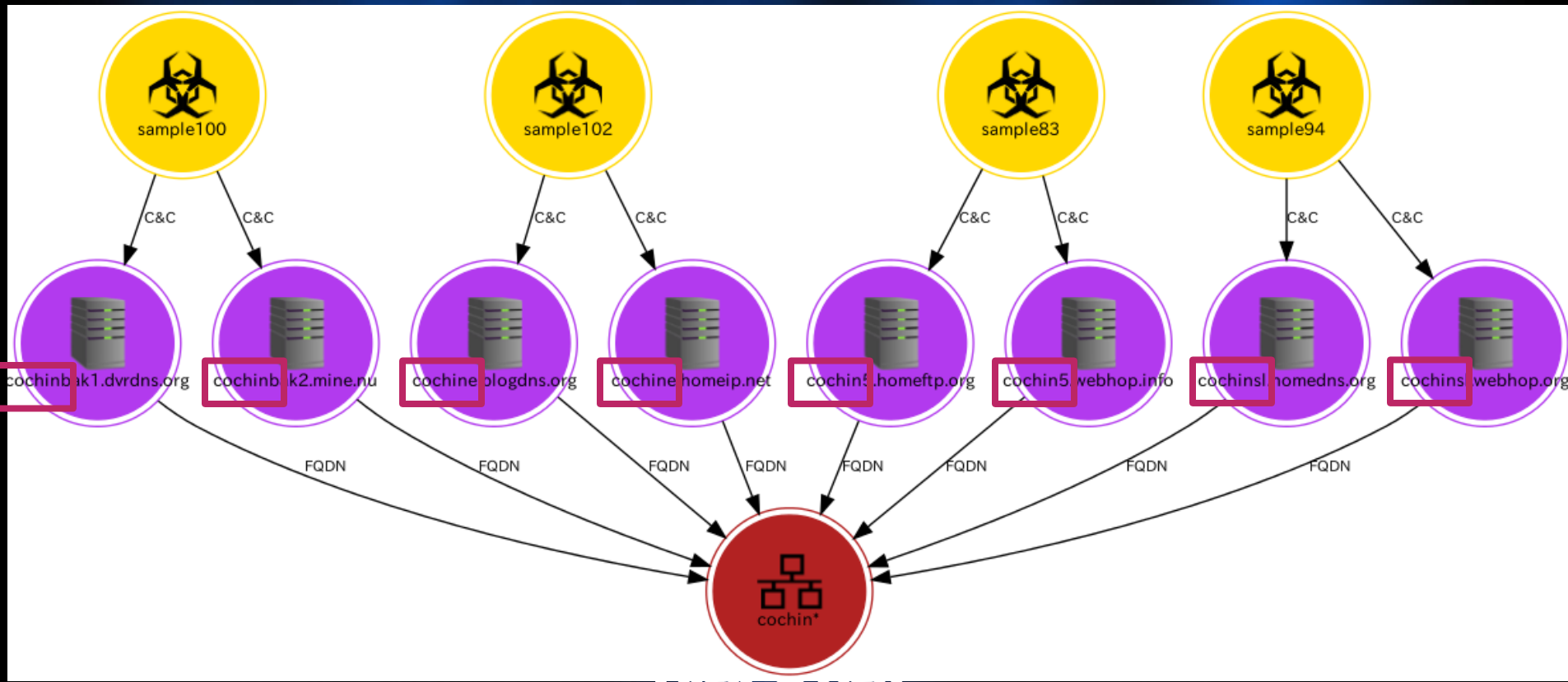
- Some samples use “360” as the service name.



COCHIN GROUP

Cochin Group

- The subdomain of some samples includes “cochin*”.

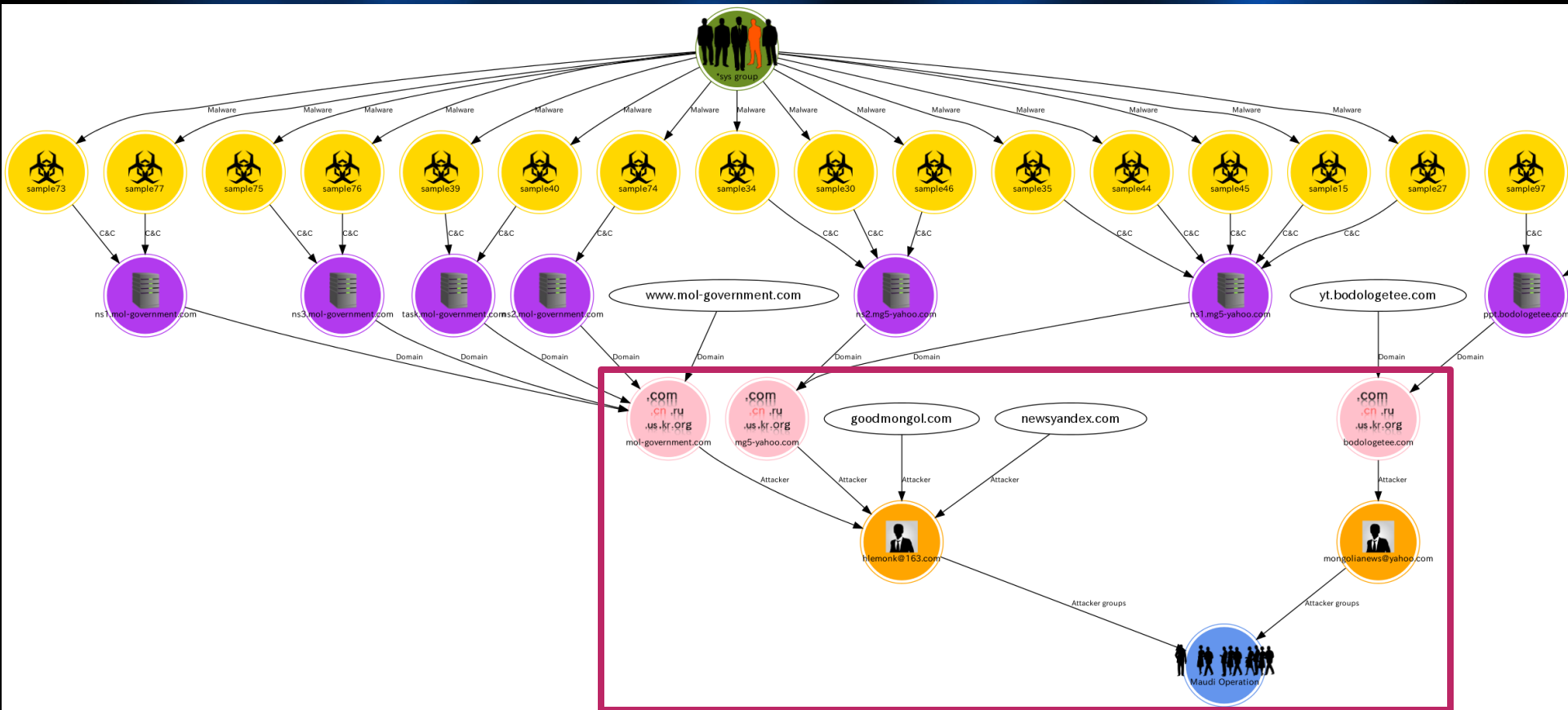


FINDING RELATIONSHIPS BETWEEN PLUGX GROUPS AND KNOWN ATTACKER GROUPS

MAUDI OPERATION

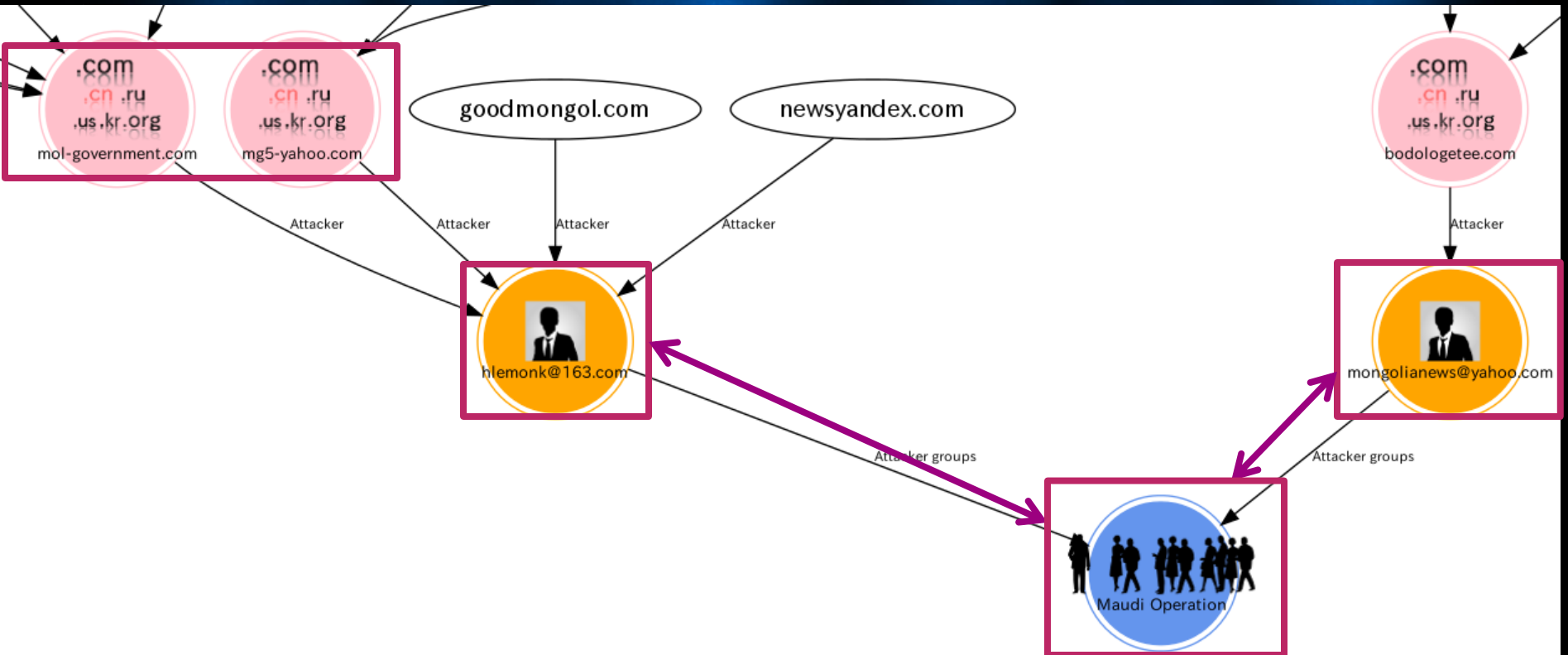
Maudi Operation

- "*Sys" group and sample 97 connect to Maudi operation [8].



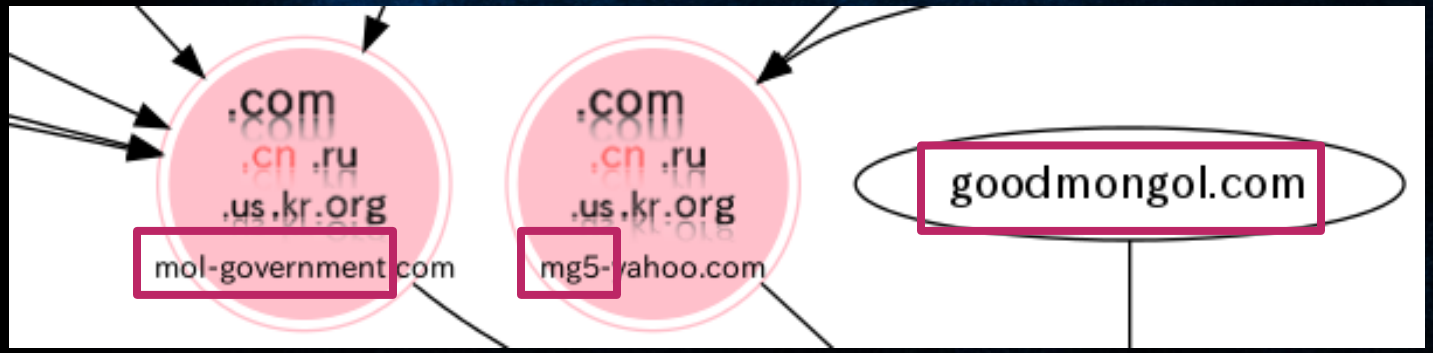
Maudi Operation

- The domain owners of “*Sys” group and sample 97 match this operation.



Maudi Operation

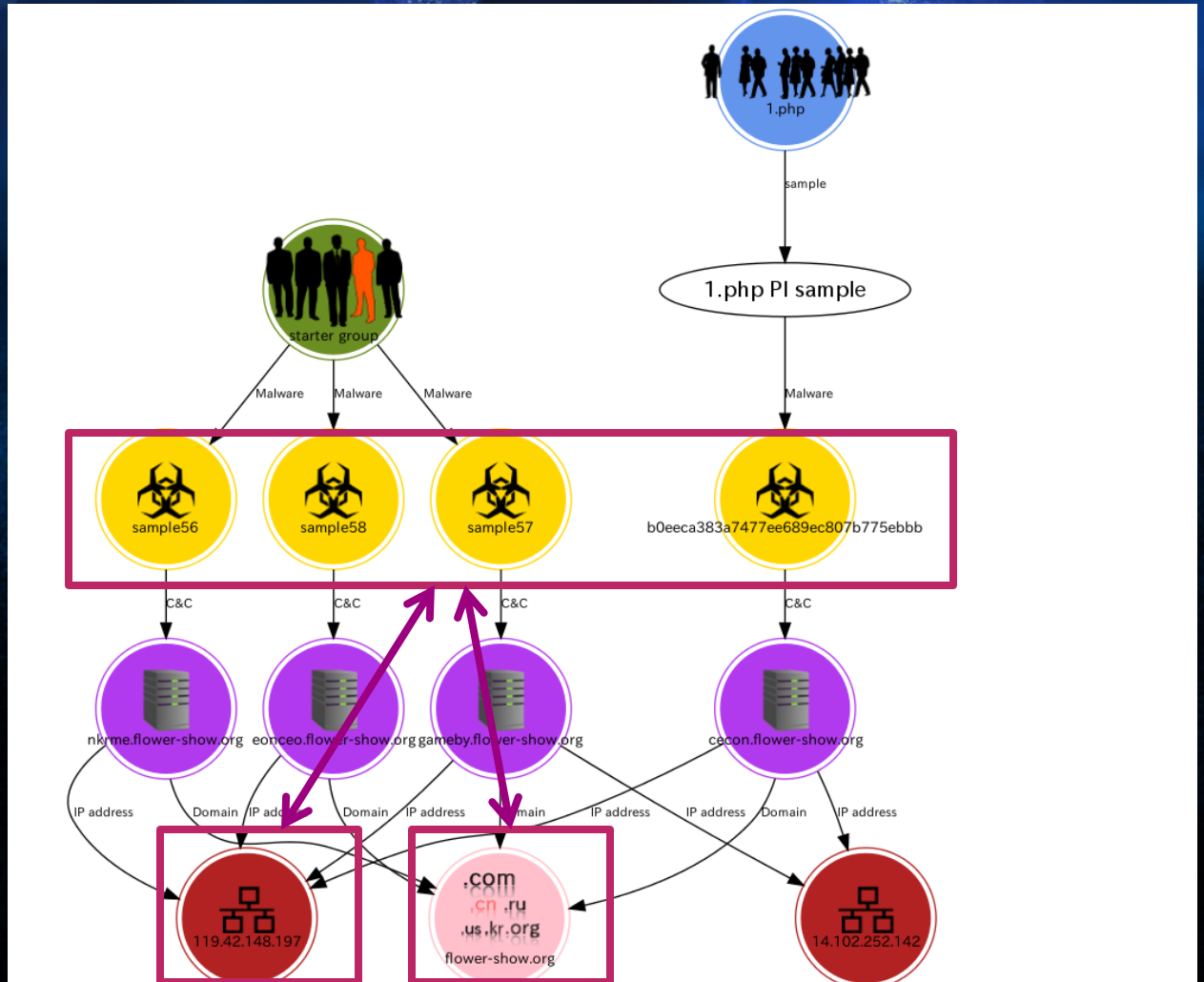
- Norman Shark said, Maudi operation targets
  - Mongolian
  - Korean
  - Local Chinese interests and human rights activists



1.PHP

1.php

- A poison ivy sample of 1.php [9] connects to “Starter” group.



1.php

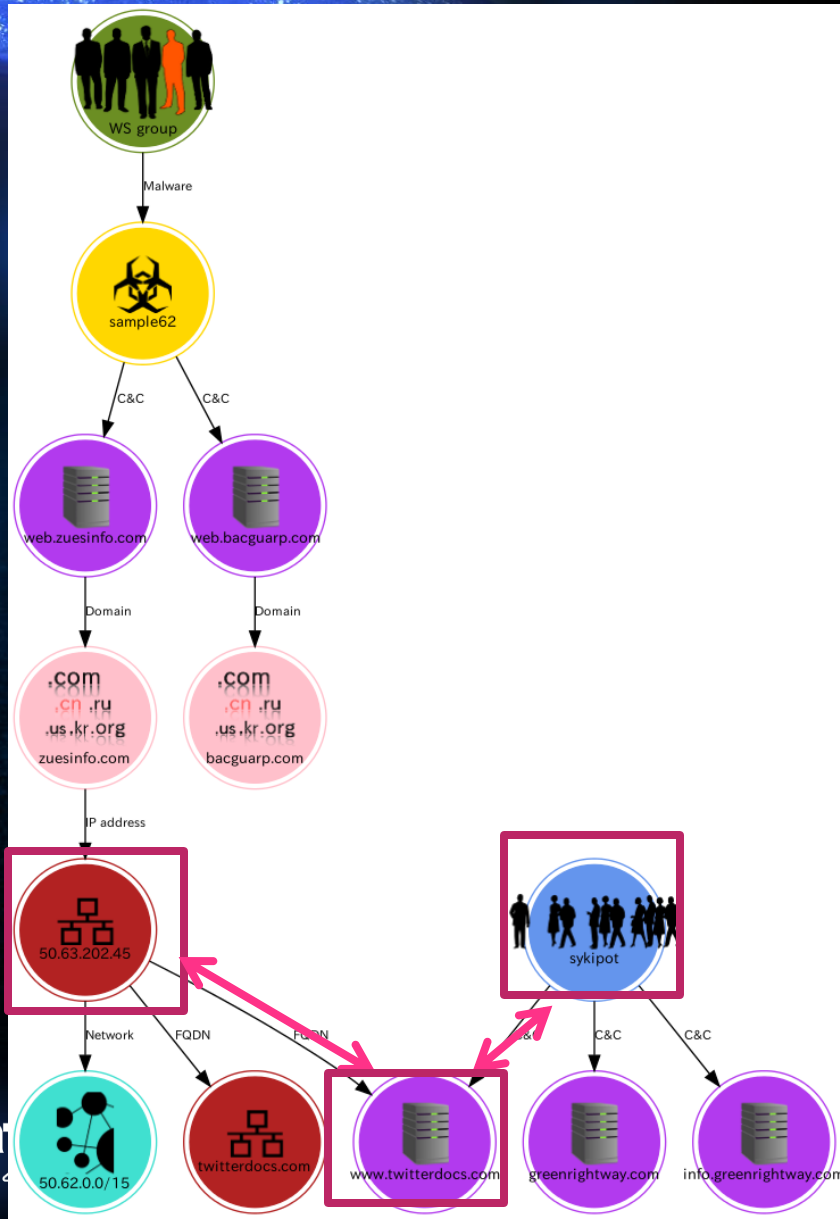
- 1.php may be related to APT1, zscaler said.
- flower-show.org targets Japan, China, Taiwan / USA relationship [10].

http://contagiodump.blogspot.jp/2011/07/message-targeting-experts-on-japan.html

SYKIPOT

Sykipot

- The IP address of the C2 used by sample 62 belonging to “WS” group is the same as the one used by Sykipot [11].

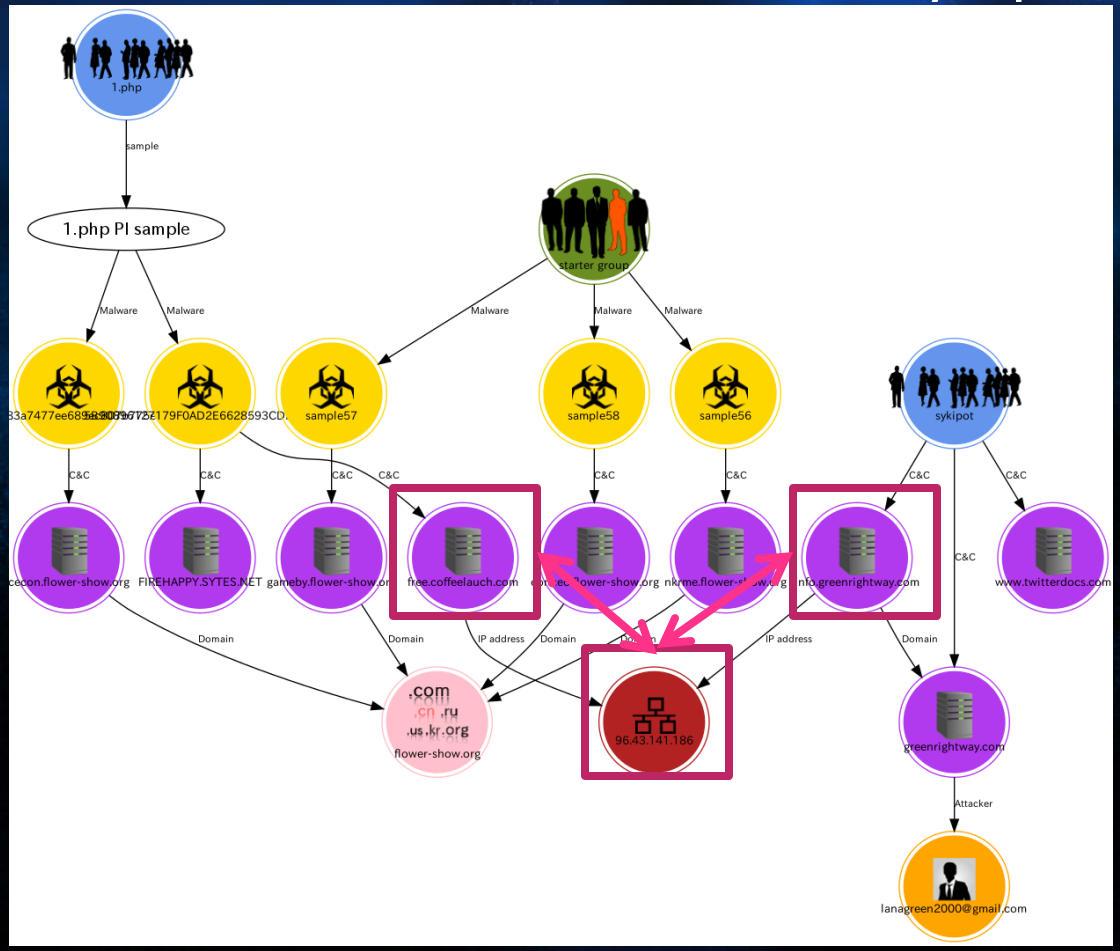


Sykipot

- The companies attacked by this latest wave of Sykipot include, but aren't limited to, organizations in the following market sectors, primarily based in the US or UK:[11]
  - Defense contractors
  - Telecommunications
  - Computer Hardware
  - Chemical
  - Energy
  - Government Departments

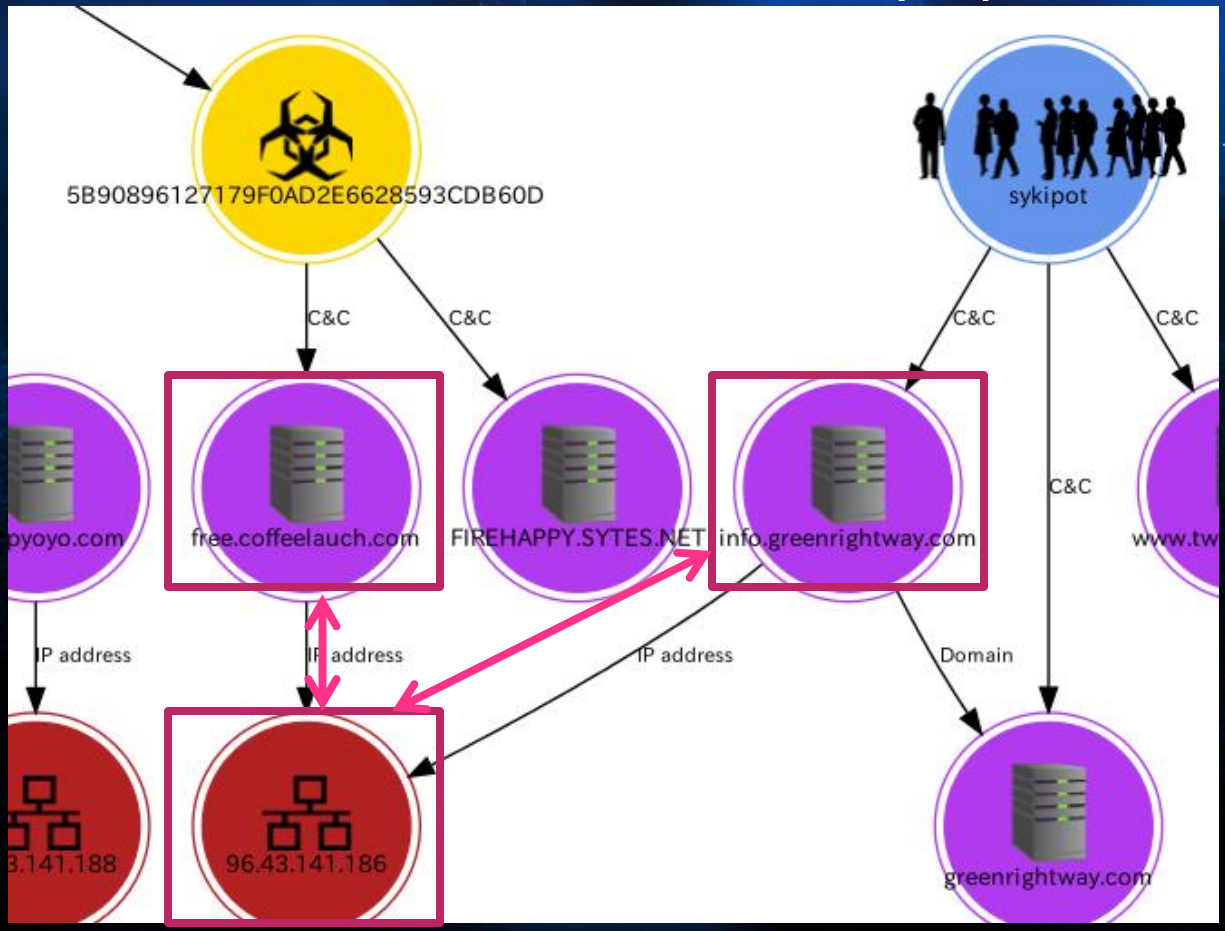
Sykipot

- Another C2 of 1.php PosinIvy sample connects to 96.43.141.186. This connects to Sykipot C2.



Sykipot

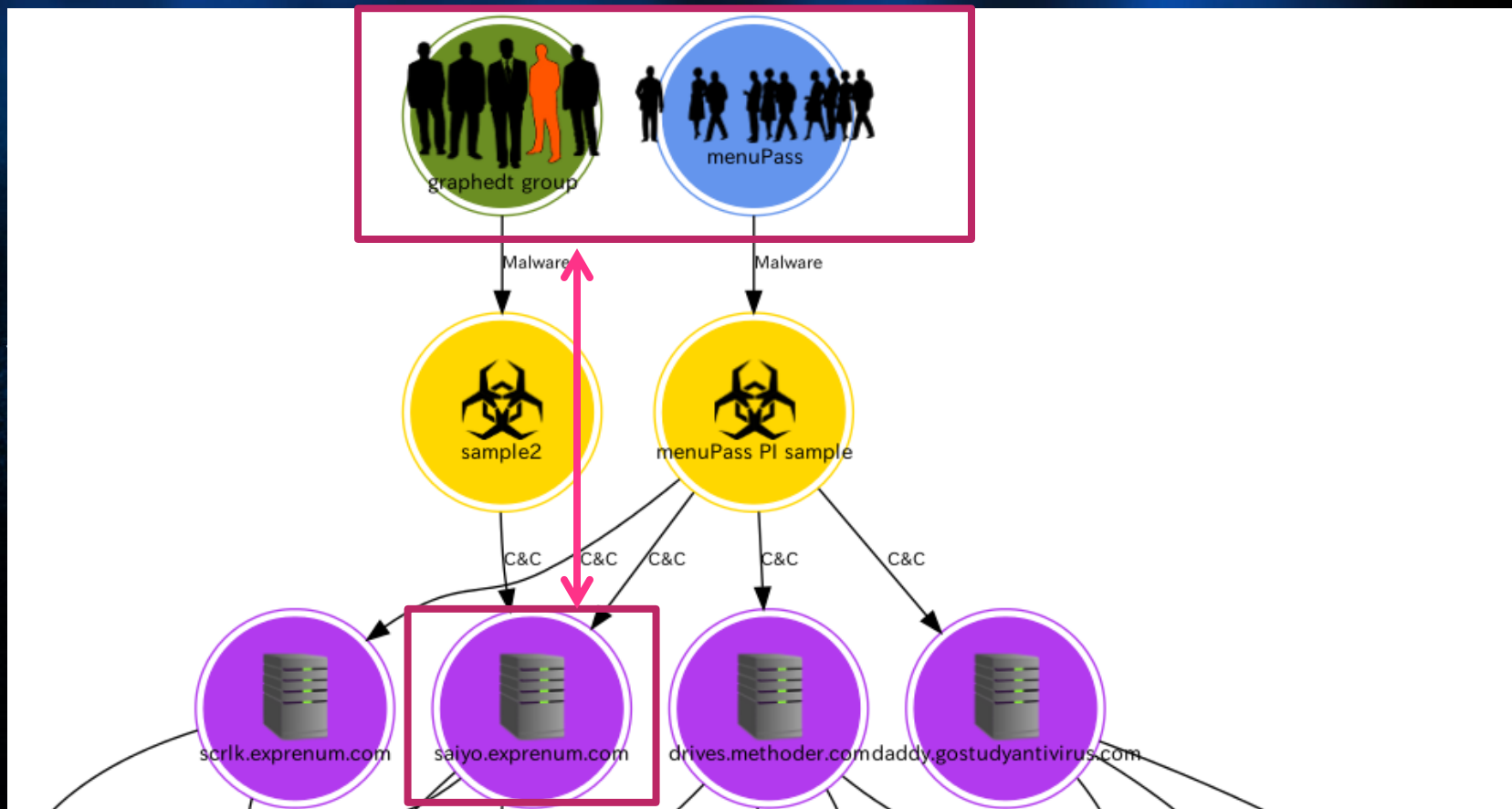
- Another C2 of 1.php PosinIvy sample connects to 96.43.141.186. This connects to Sykipot C2.



MENUPASS

menuPass

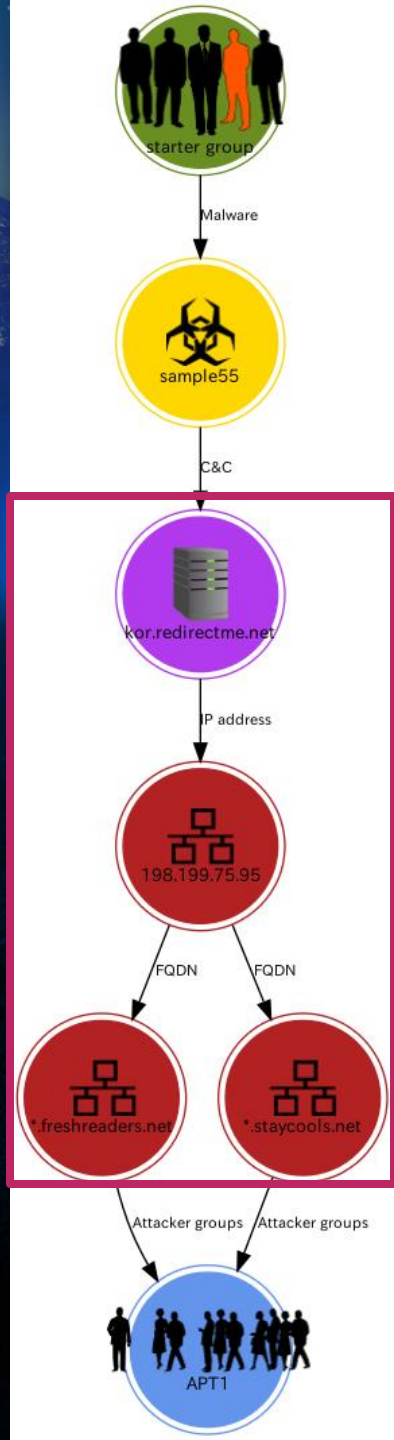
- Graphedt group is actually the same as menuPass [12].
- Some of our customers were targeted from menuPass in Aug to Oct, 2013 [13].



APT1

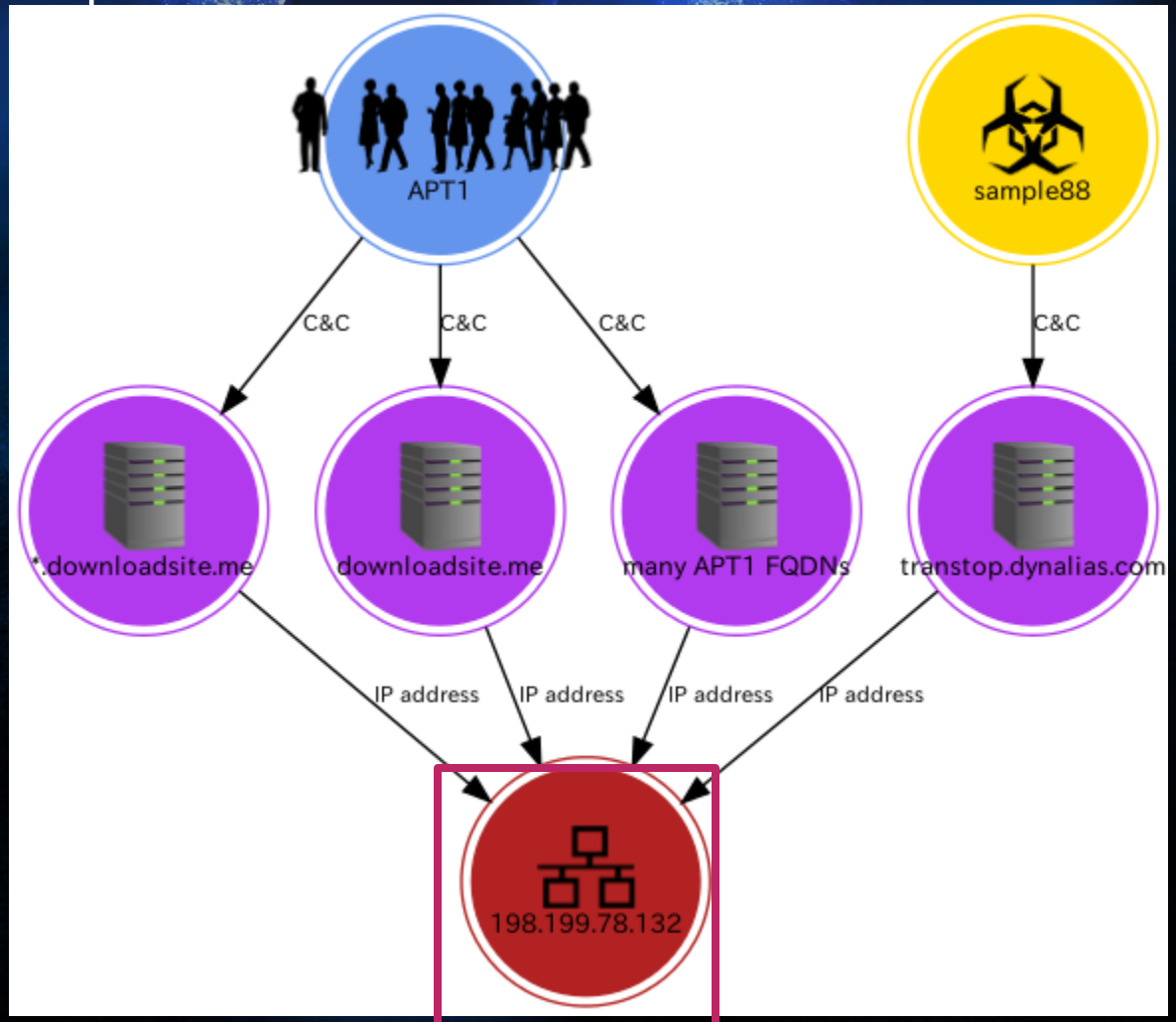
APT1

- The C2 of sample 55 belonging to “Starter” group is the same IP address as the one used by APT1 [14].



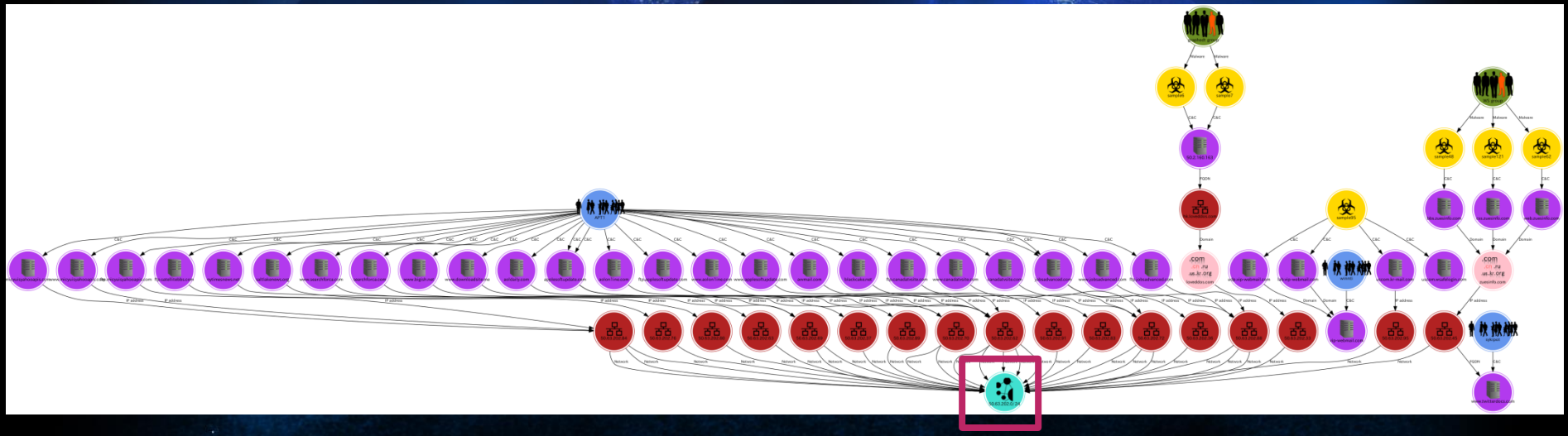
APT1

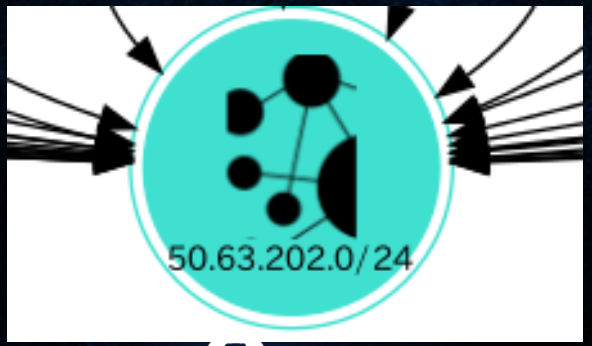
- The C2 of the sample 88 has the same IP address as the one used by APT1.



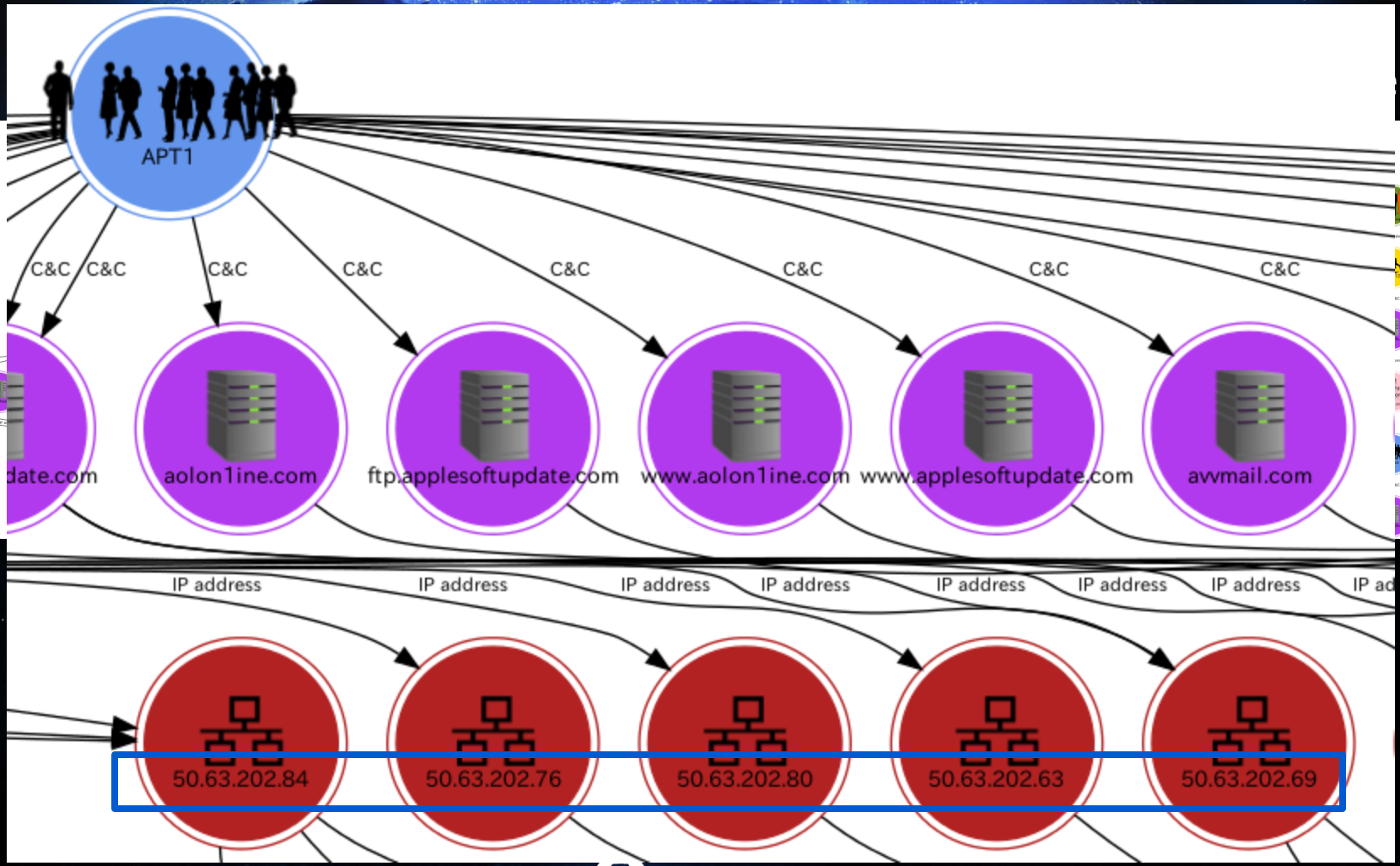
APT1

- Many C2s of APT1 samples and “graphedt” / “WS” group belong to the same network range.

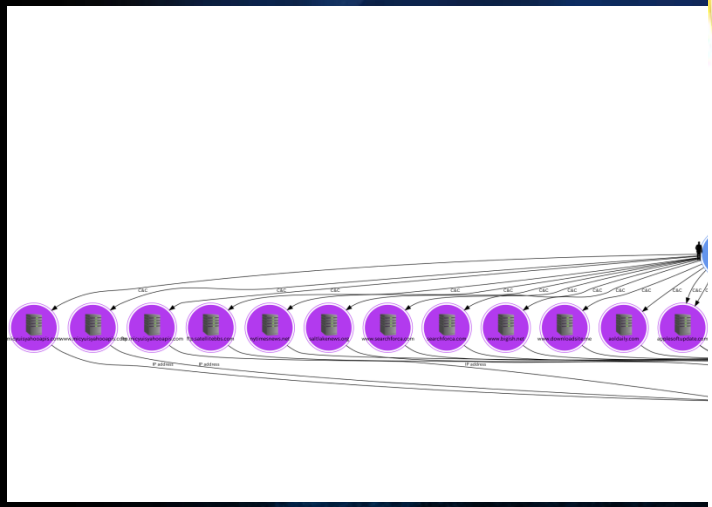


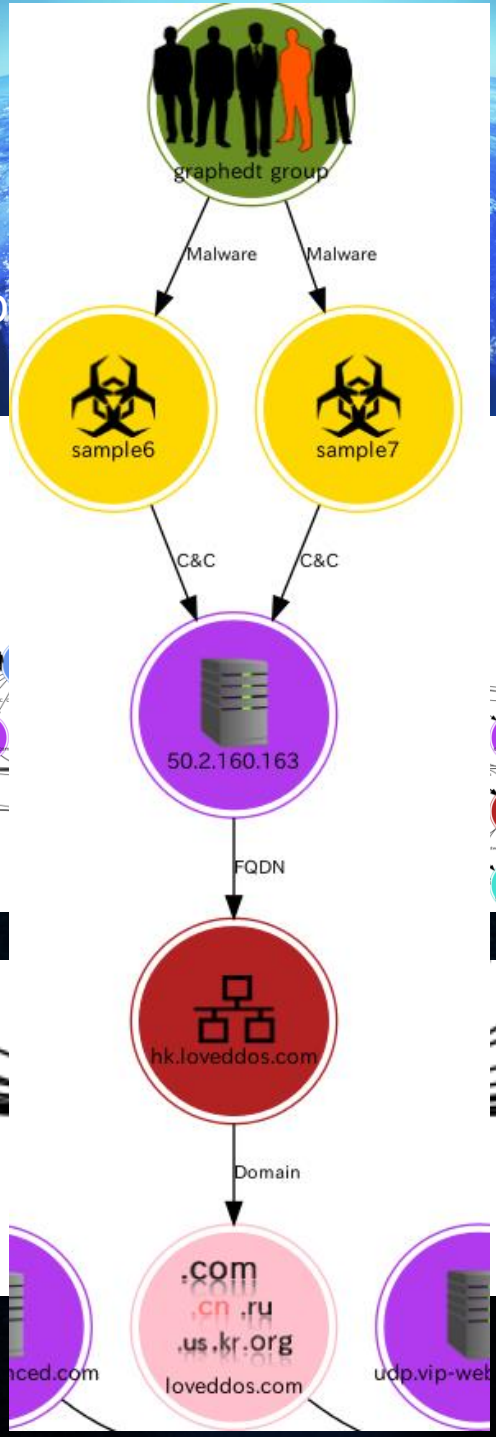


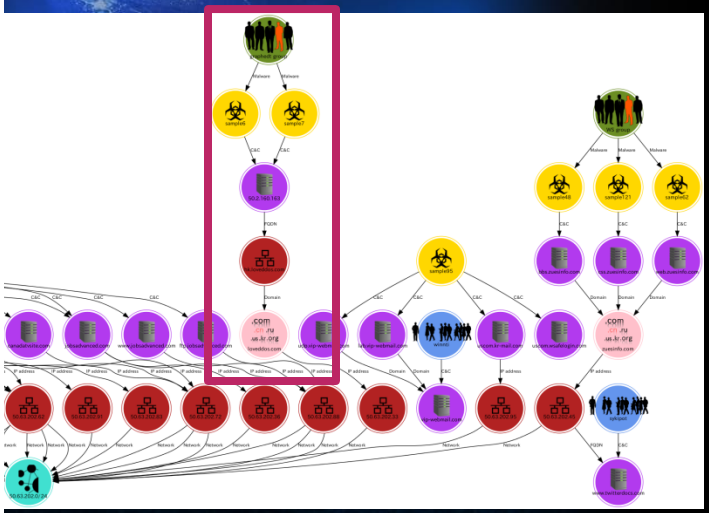
APT1


APT1
C&C
C&C
C&C
C&C
C&C
C&C
C&C
C&C
aolon1ine.com
ftp.applesoftupdate.com
www.aolon1ine.com
www.applesoftupdate.com
avvmail.com
IP address
IP address
IP address
IP address
IP address
IP address
IP address
50.63.202.84
50.63.202.76
50.63.202.80
50.63.202.63
50.63.202.69

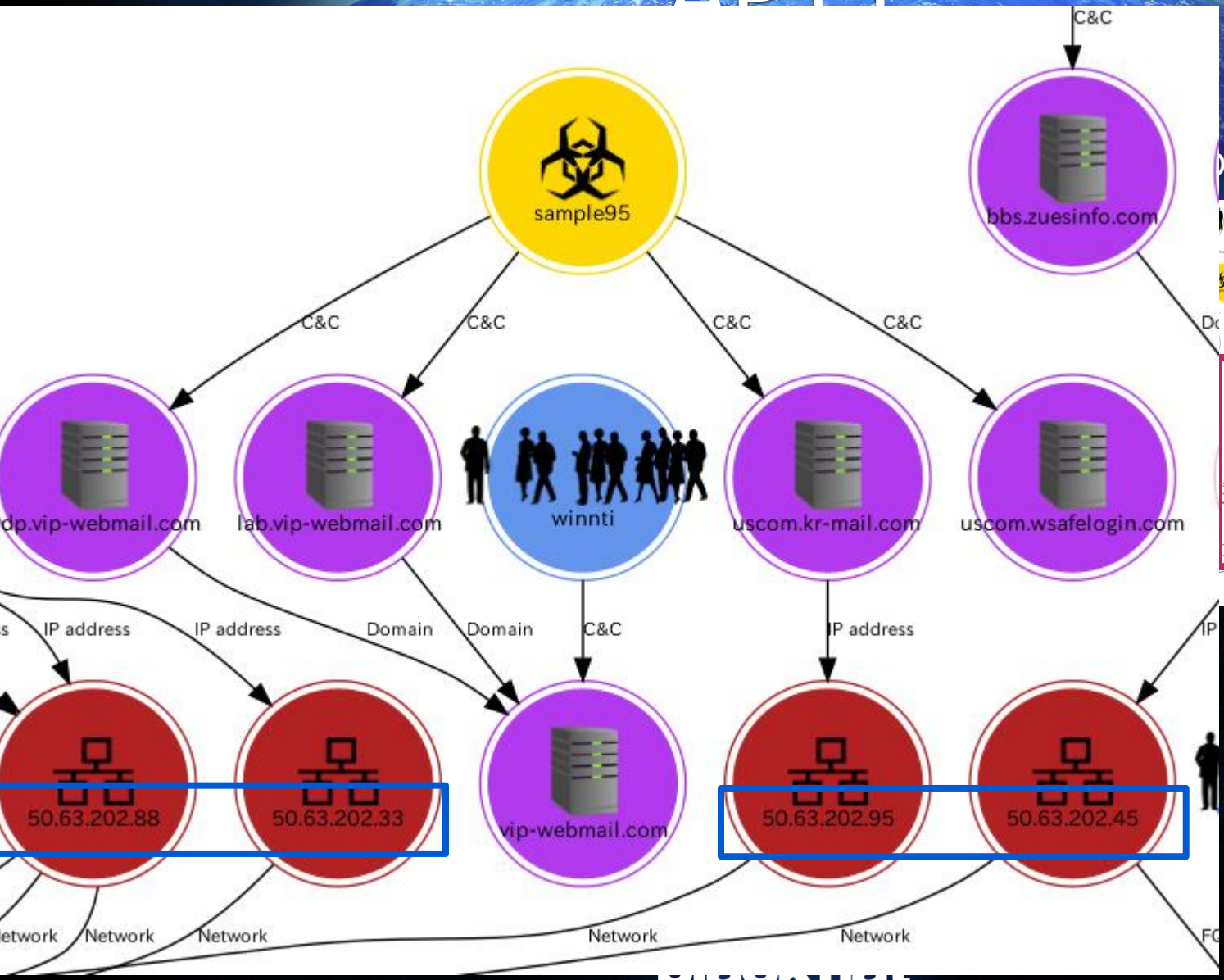
- Many C2 of APT1 samp “WS” group belong to the same network range.



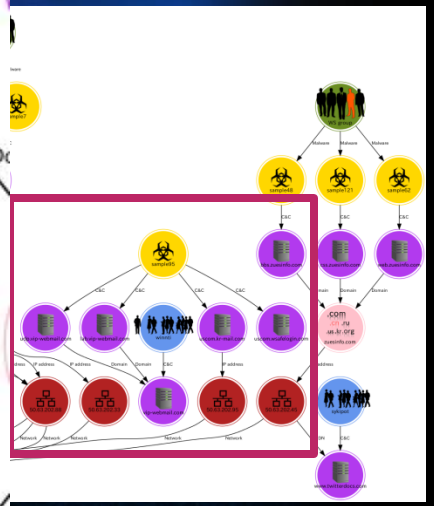




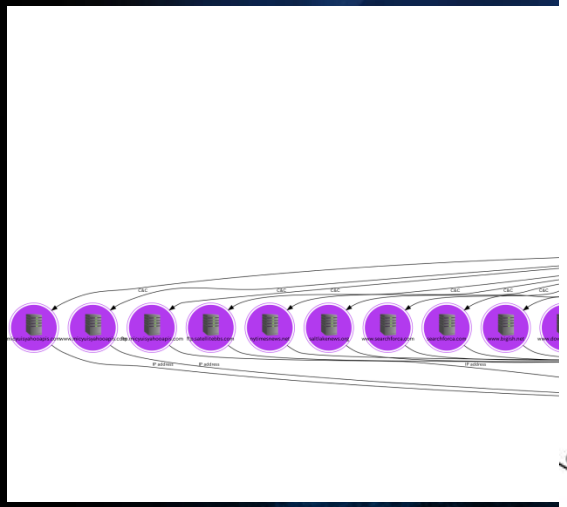
APT1

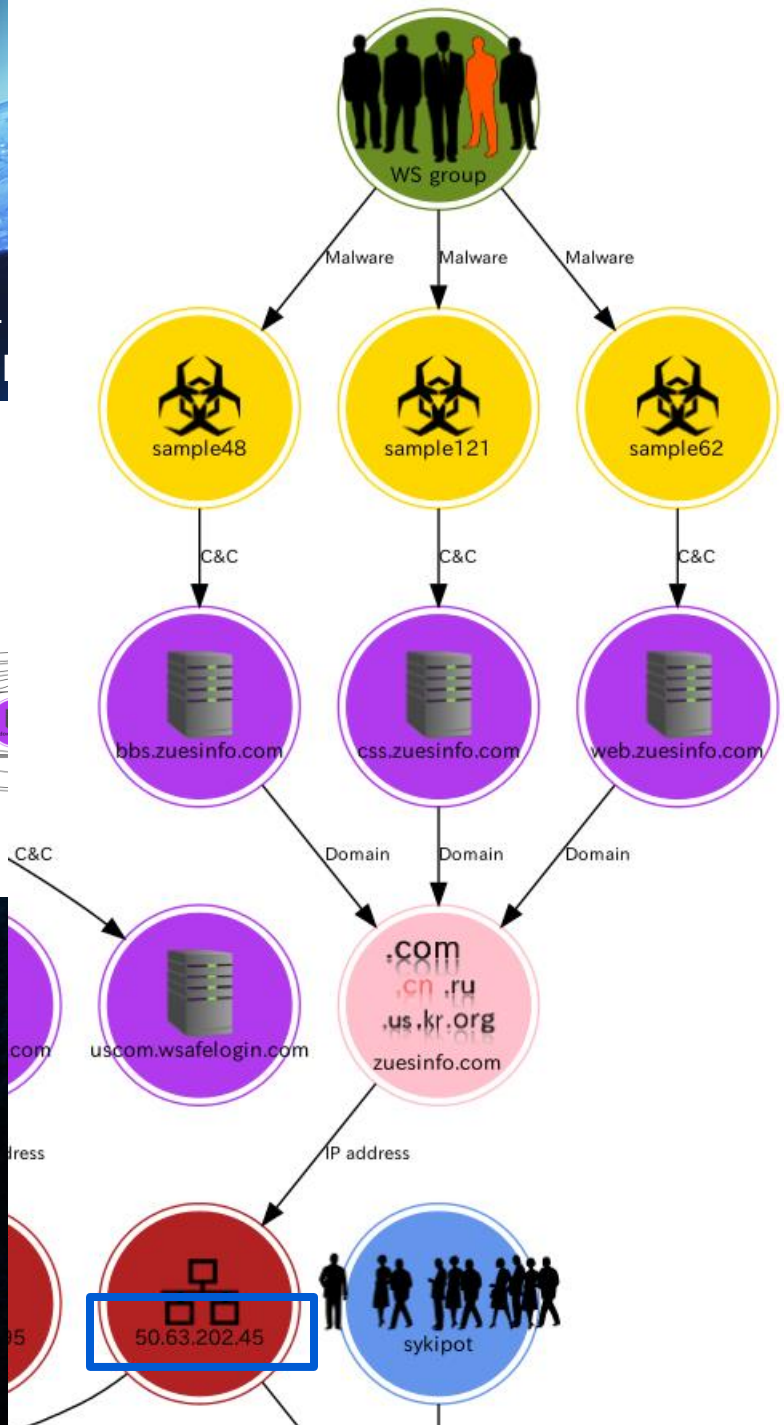


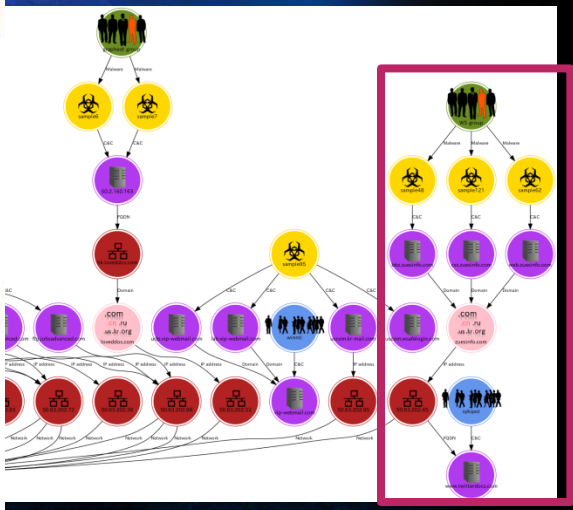
belong to the



- Many C2 of APT1 group belong to the same network ra



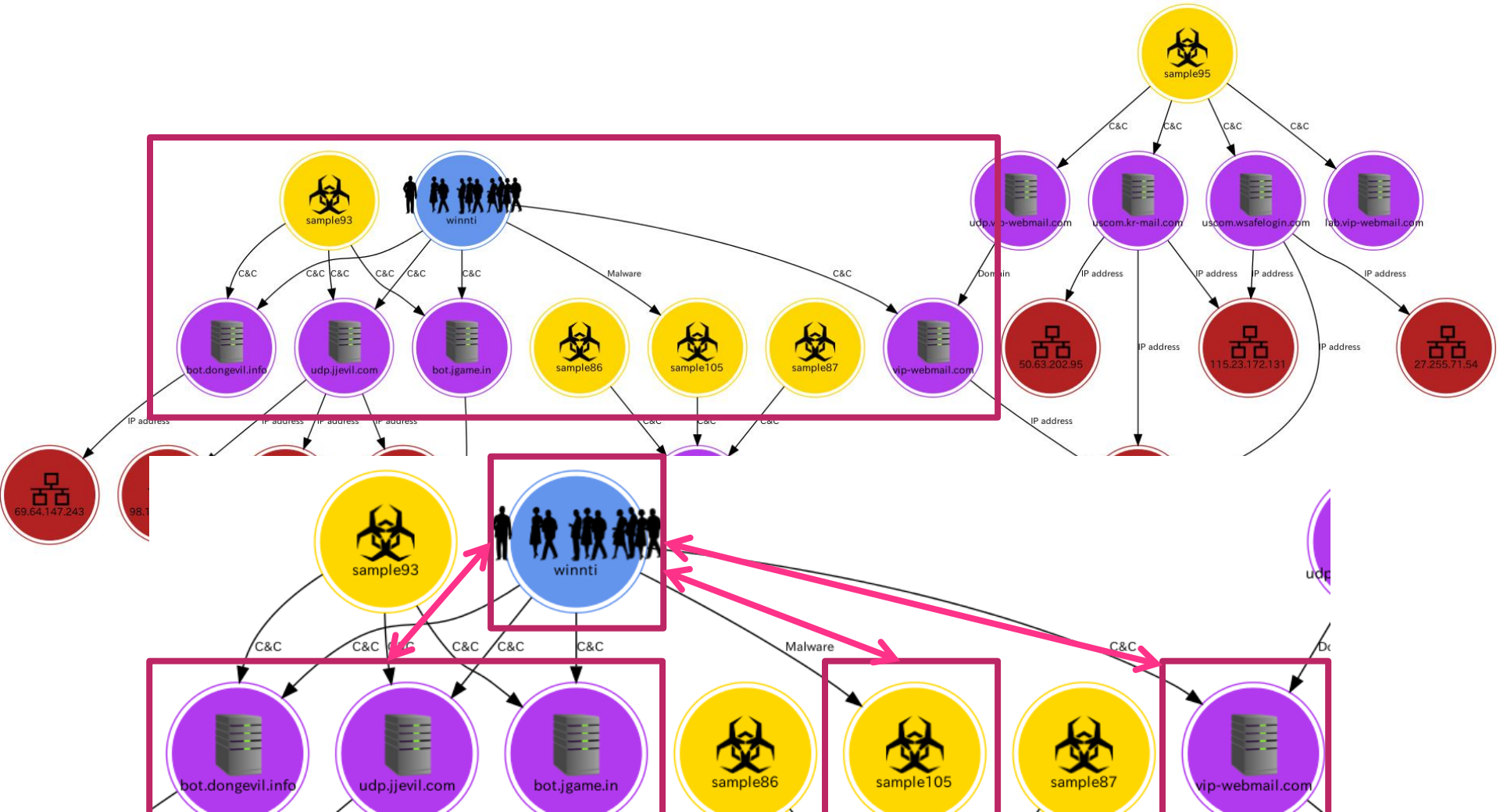




WINNTI

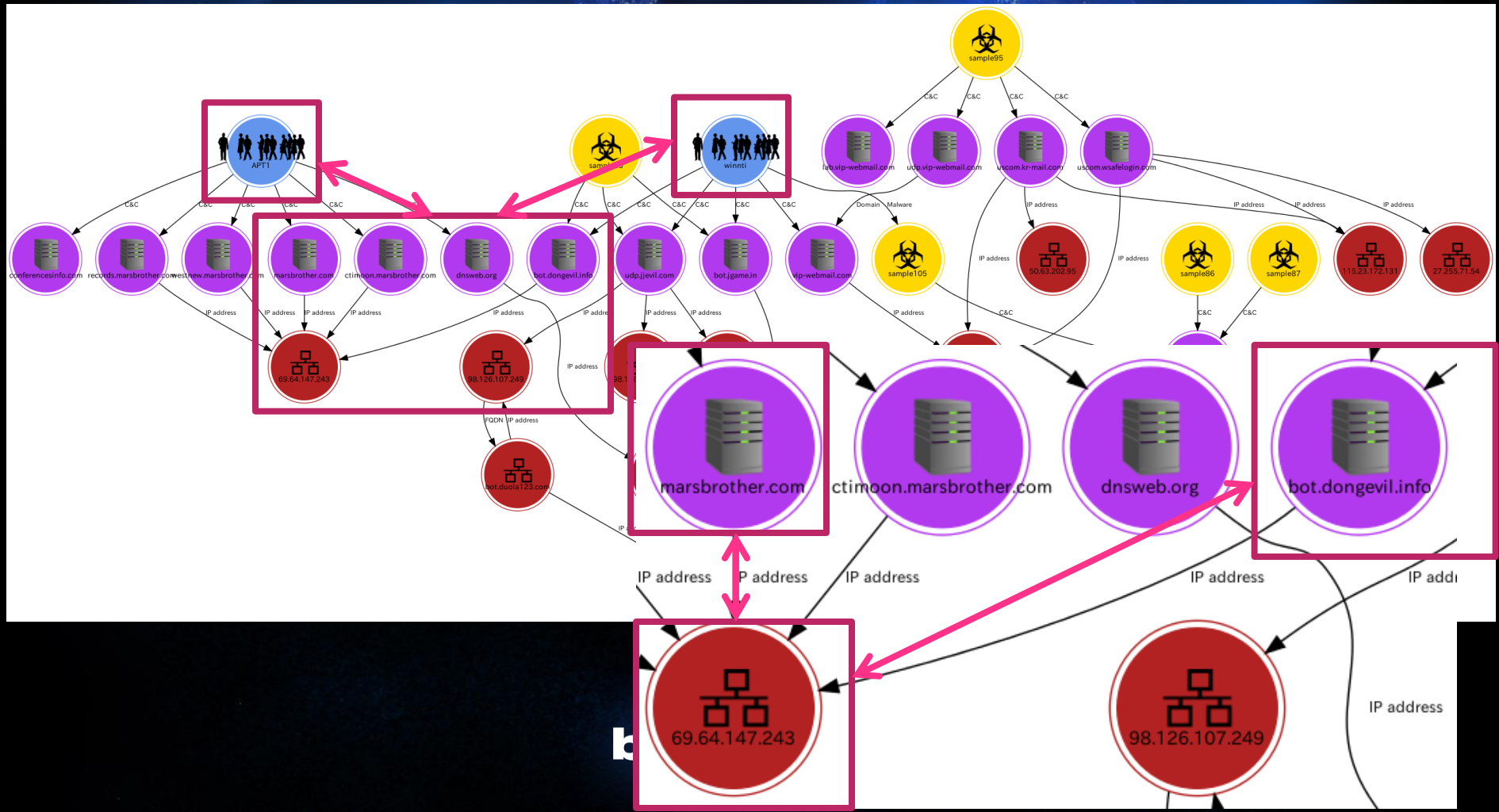
Winnti

- The C2s of some samples are the same as the C2s used by Winnti [15] [16].



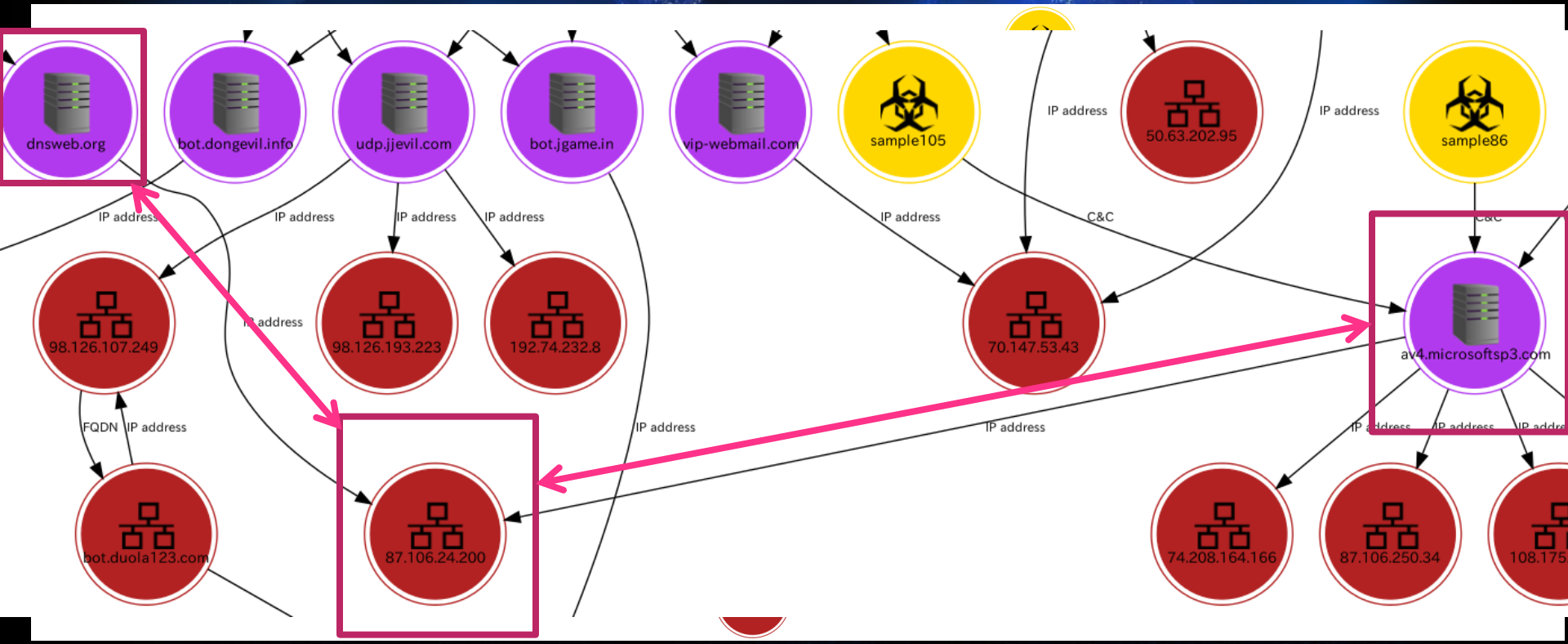
Winnti

- Those Winnti samples are also related to APT1.



Winnti

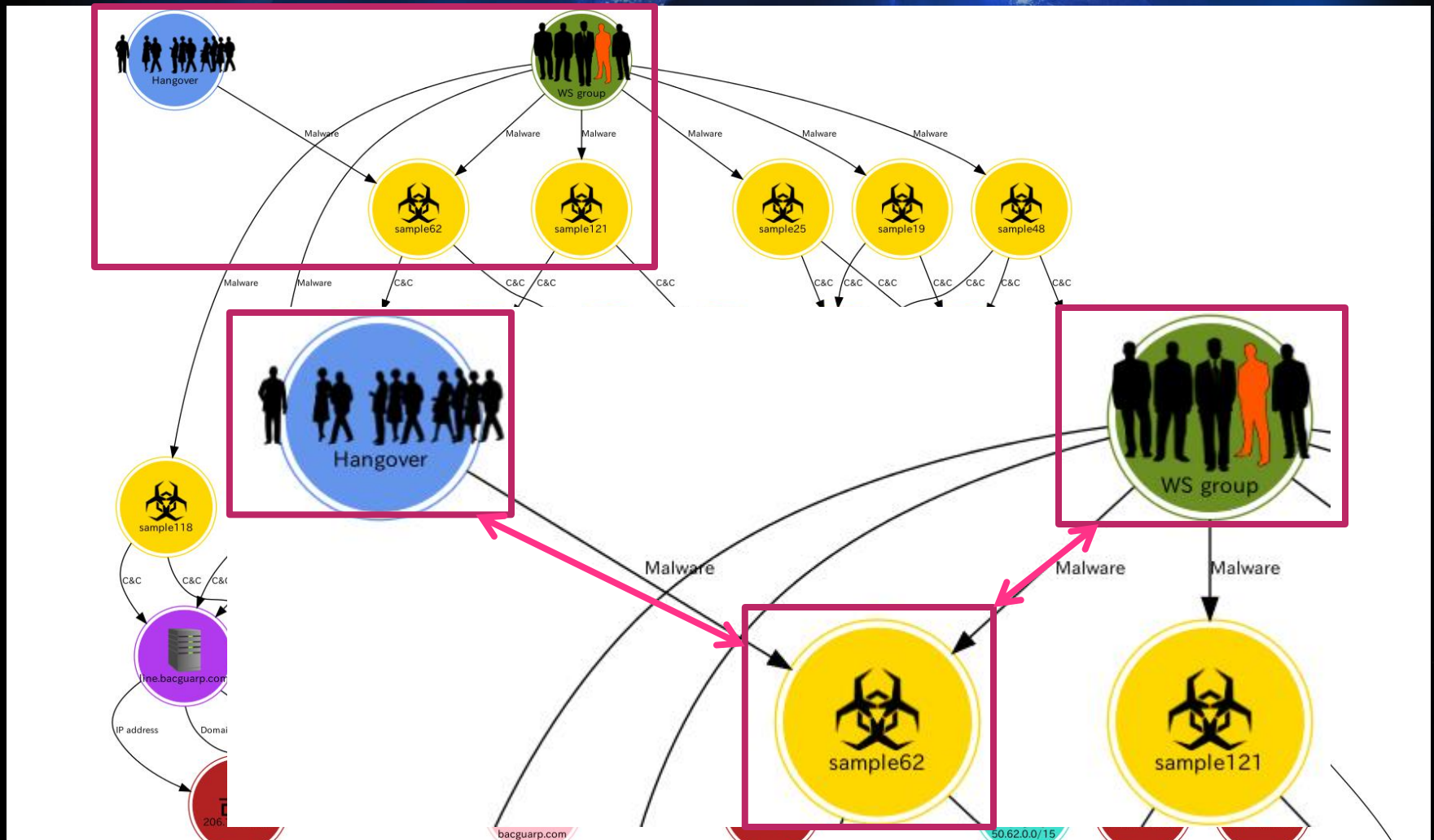
- Those Winnti samples are also related to APT1.




black hat
ASIA 2014
HANGOVER

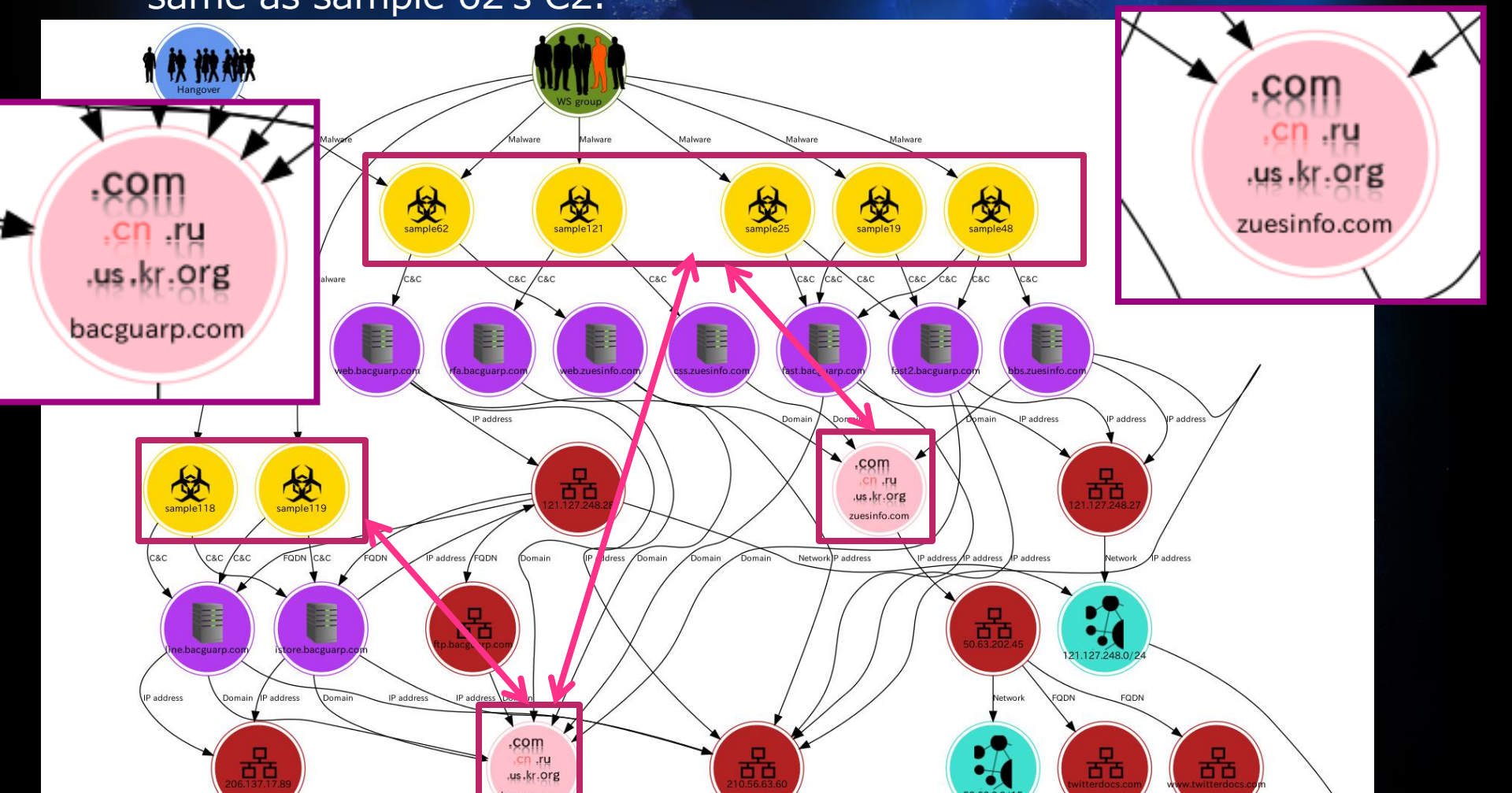
Hangover

- "WS" group is the same as Hangover group [17].



Hangover

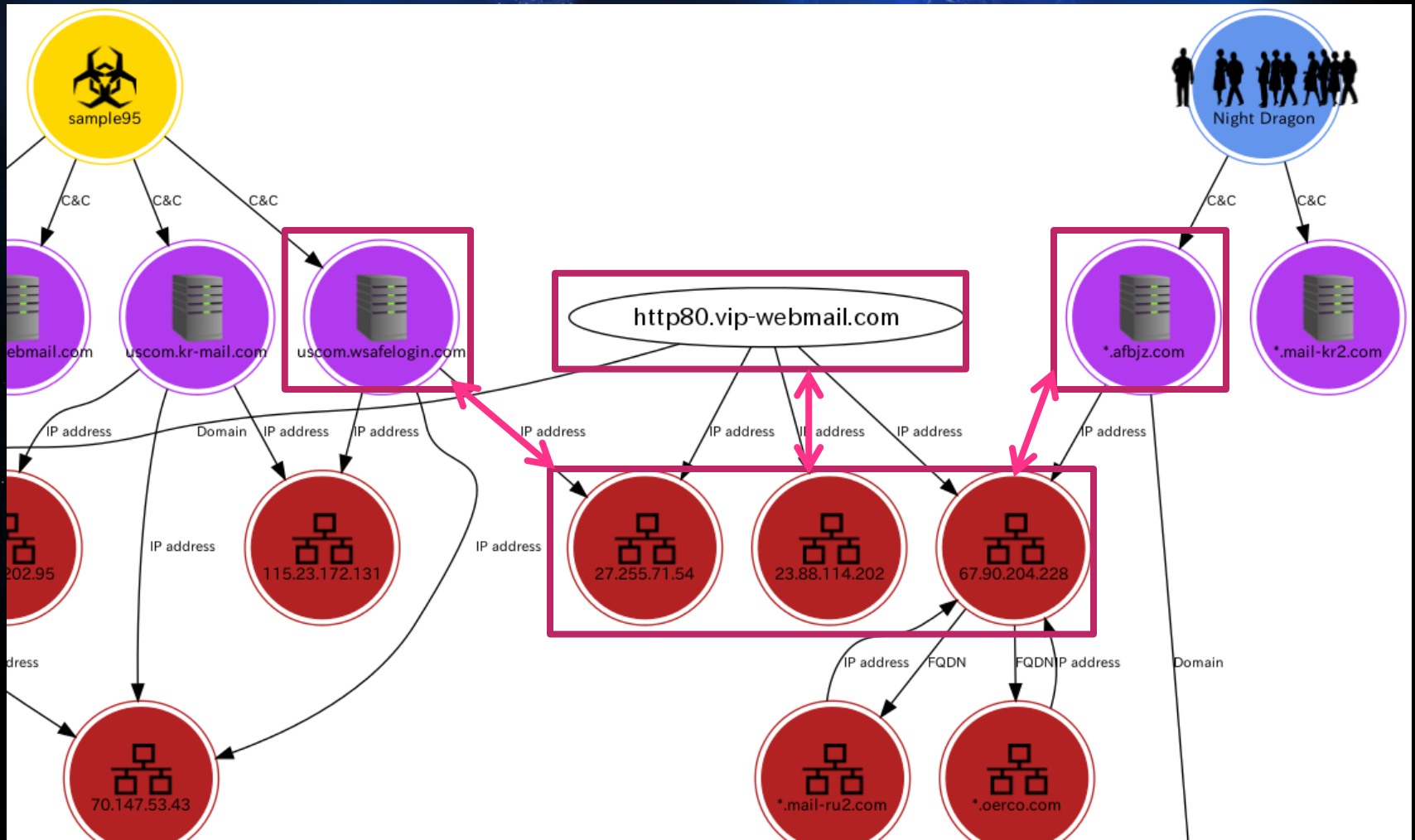
- All samples of “WS” group connect to two domains. Those are the same as sample 62’s C2.




black hat
ASIA 2014
NIGHTDRAGON

Night Dragon

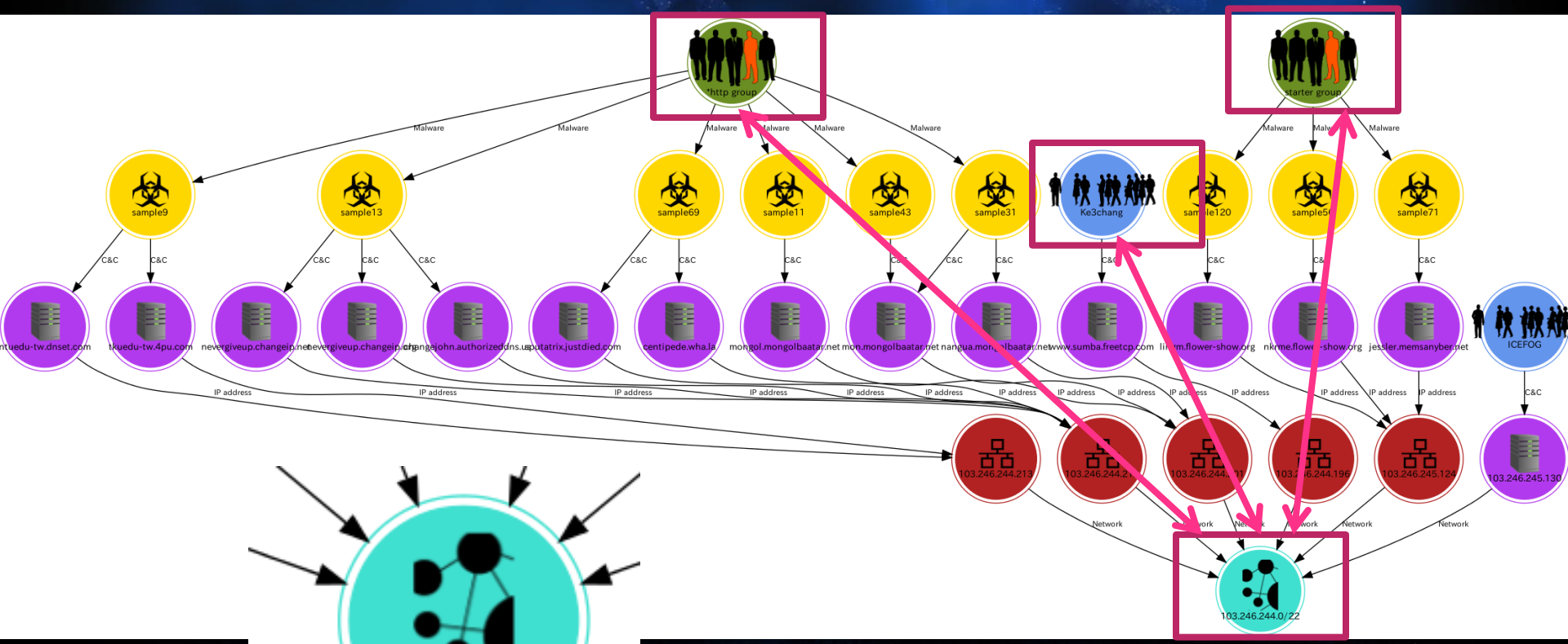
- Sample 95 related to Winnti is also related to Night Dragon [18].



KE3CHANG

Ke3Chang

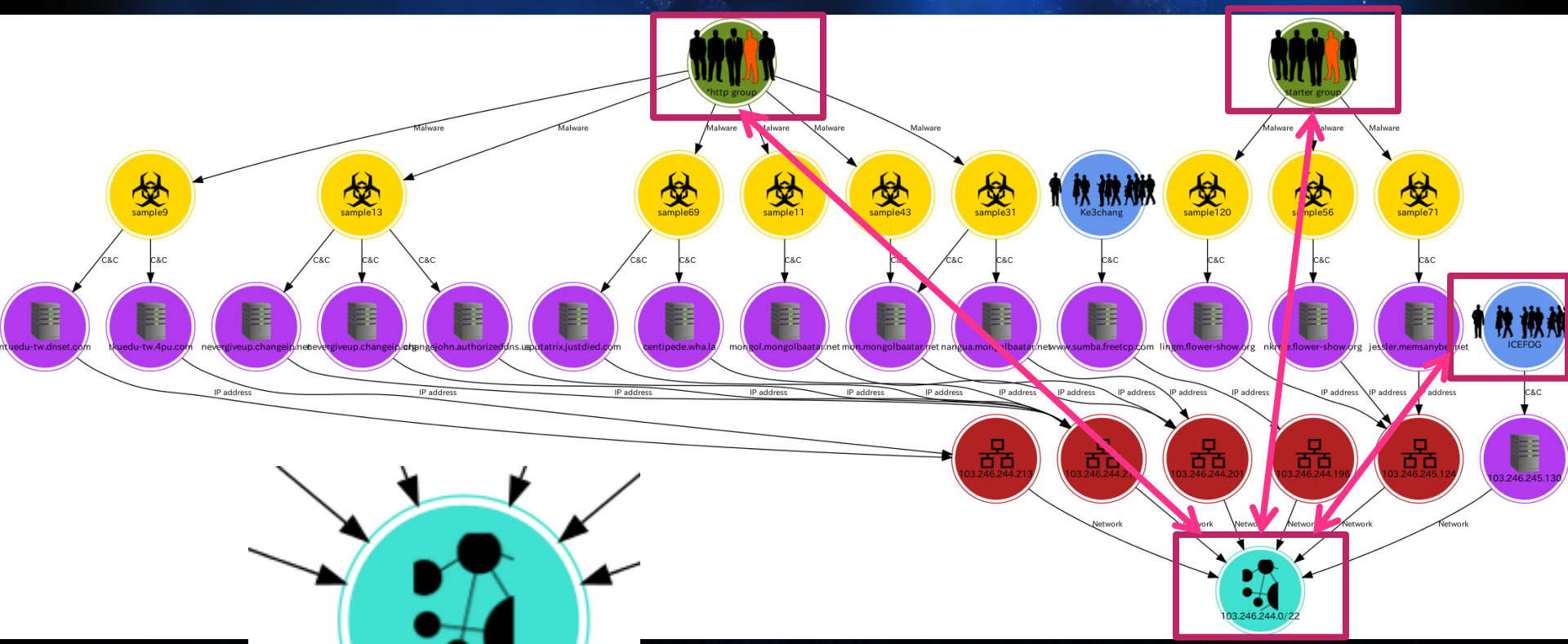
- Many C2s of "*Http" / "Starter" group and Ke3Chang [19] samples belong to the same network range.



ICEFOG

ICEFOG

- Many C2s of “*Http” / “Starter” group and ICEFOG [20] samples belong to the same network range.



103.246.244.0/22

KHAAN QUEST

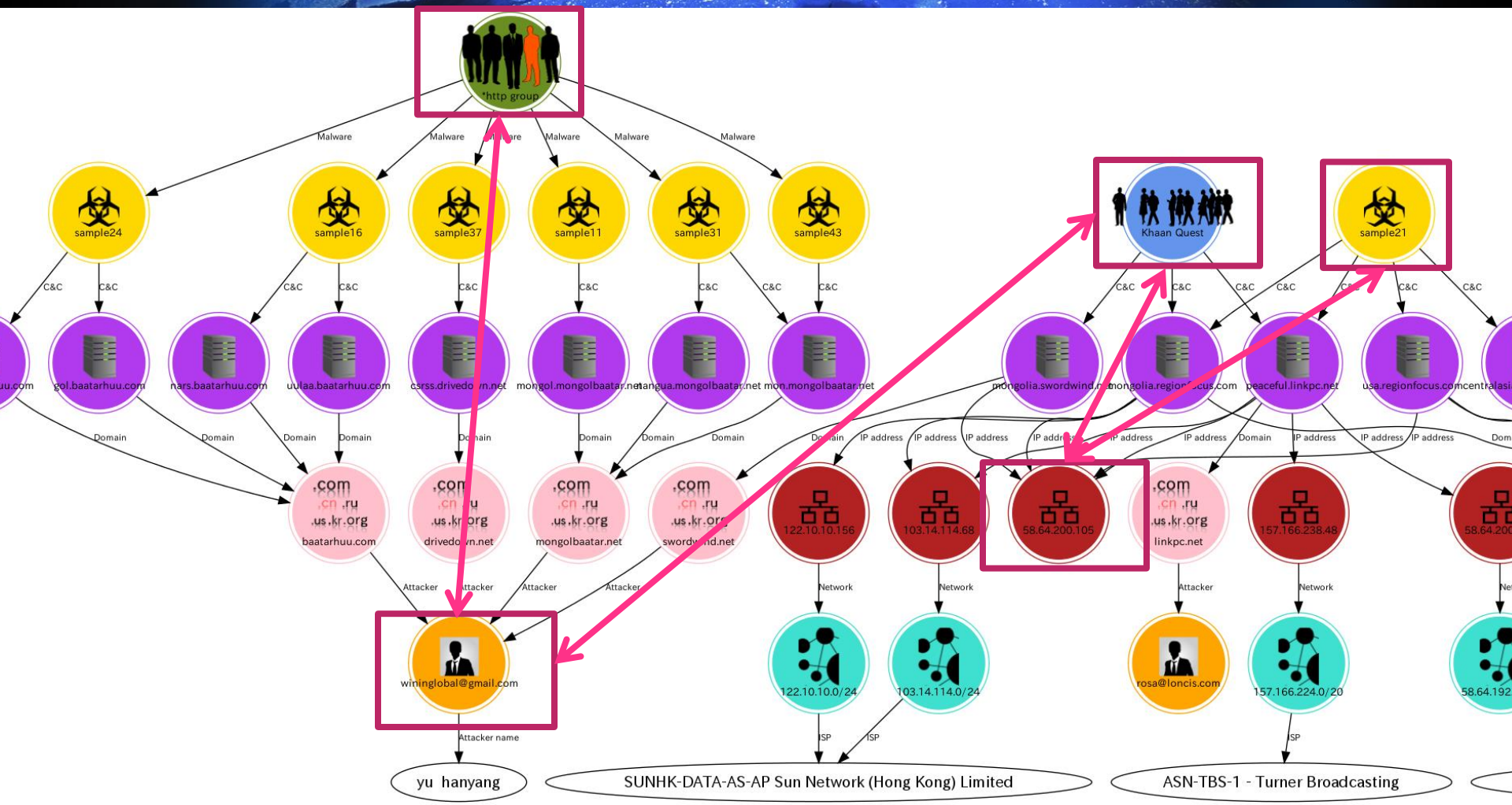
Khaan Quest

Domain	IP Resolution
bss.publicvm.com	58.64.200.105
central.swordwind.net	58.64.200.105
lwl.publicvm.com	58.64.200.105
mongolia.regionfocus.com	58.64.200.105
mongolia.swordwind.net	58.64.200.105
peaceful.linkpc.net	58.64.200.105
peaceful.publicvm.com	58.64.200.105
peaceful.swordwind.net	58.64.200.105
usa.regionfocus.com	58.64.200.105
blog.cainformations.com	58.64.200.106
bluesnow.alternate009.com	58.64.200.106
loggol.cainformations.com	58.64.200.106
module.cainformations.com	58.64.200.106
mongol1.mine.nu	58.64.200.106
mongolia.regionfocus.com	58.64.200.106
newsgdeep.alternate009.com	58.64.200.106
nuyoahz.alternate009.com	58.64.200.106
oayoahzfks.alternate009.com	58.64.200.106
peaceful.linkpc.net	58.64.200.106
peaceful.publicvm.com	58.64.200.106
transfer.cainformations.com	58.64.200.106
usa.regionfocus.com	58.64.200.106
wing.alternate009.com	58.64.200.106

Domain	Registrant Email Address
mongol1.mine.nu	hostmaster@dyndns.com
bss.publicvm.com	jchen@dnsexit.com
lwl.publicvm.com	jchen@dnsexit.com
peaceful.publicvm.com	jchen@dnsexit.com
peaceful003.publicvm.com	jchen@dnsexit.com
blog.cainformations.com	lekomc@gmail.com
bluesnow.alternate009.com	lekomc@gmail.com
loggol.cainformations.com	lekomc@gmail.com
module.cainformations.com	lekomc@gmail.com
newsgdeep.alternate009.com	lekomc@gmail.com
nuyoahz.alternate009.com	lekomc@gmail.com
oayoahzfks.alternate009.com	lekomc@gmail.com
transfer.cainformations.com	lekomc@gmail.com
wing.alternate009.com	lekomc@gmail.com
peaceful.linkpc.net	linkpc.net@private.netdom.com
central.swordwind.net	wininglobal@gmail.com
mongolia.swordwind.net	wininglobal@gmail.com
peaceful.swordwind.net	wininglobal@gmail.com
centralasia.regionfocus.com	yyan_79@hotmail.com
mongolia.regionfocus.com	yyan_79@hotmail.com
regionfocus.com	yyan_79@hotmail.com
ssupdate.regionfocus.com	yyan_79@hotmail.com
usa.regionfocus.com	yyan_79@hotmail.com

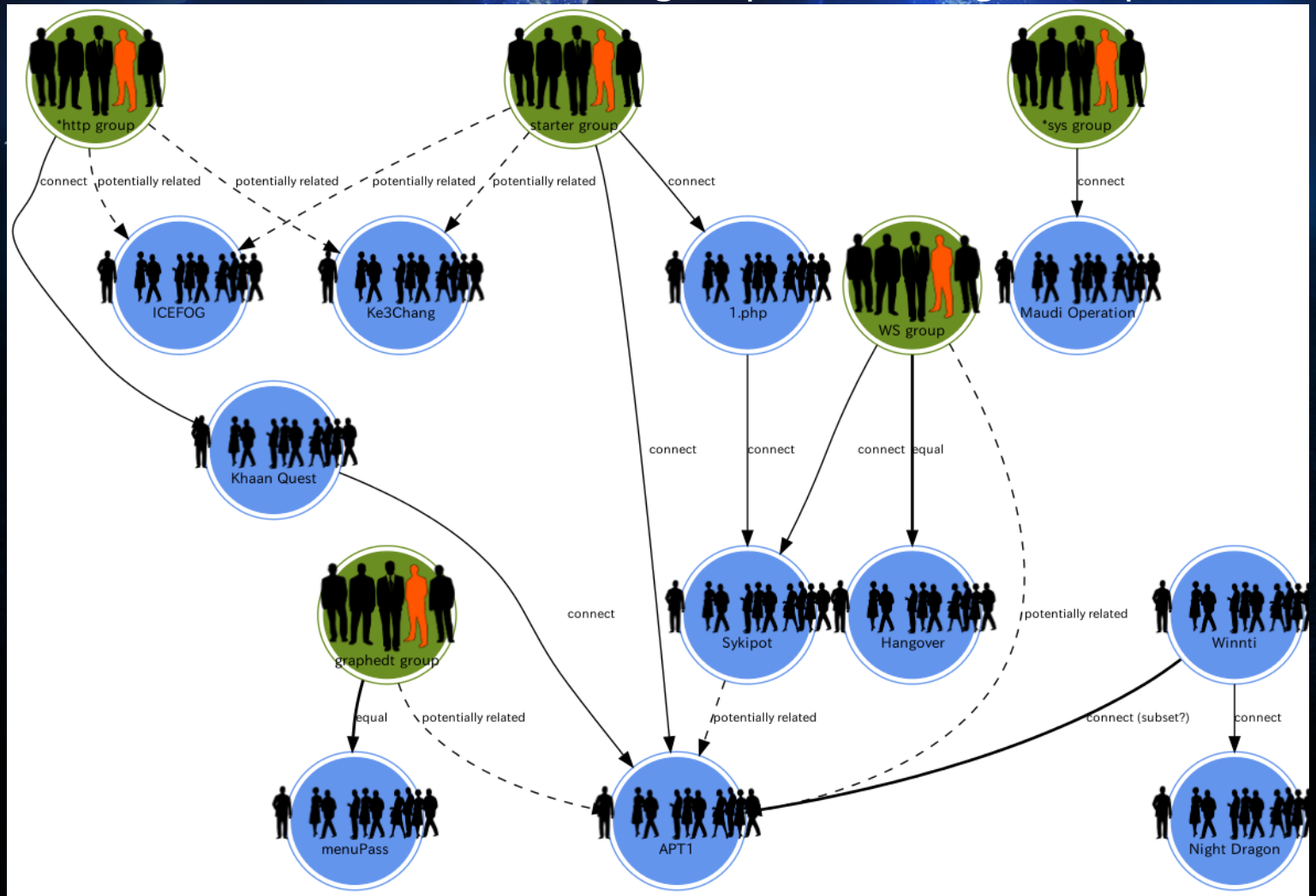
[21] http://www.threatconnect.com/news/khaan-quest-chinese-cyber-espionage-targeting-mongolia/

Khaan Quest


*http group
Malware
sample24
sample16
sample37
sample11
sample31
sample43
C&C
gol.baatarhuu.com
nars.baatarhuu.com
uulaa.baatarhuu.com
mongol.mongolbaatar.net
mon.mongolbaatar.net
Domain
.com
.cn .ru
.us .kr .org
baatarhuu.com
mongolbaatar.net
Attacker
wininglobal@gmail.com
Attacker name
yu hanyang
Khaan Quest
sample21
peaceful.linkpc.net
usa.regionfocus.com
IP address
122.10.10.156
103.14.114.68
58.64.200.105
157.166.238.48
Network
122.10.10.0/24
103.14.114.0/24
linkpc.net
rosa@loncis.com
157.166.224.0/20
ISP
SUNHK-DATA-AS-AP Sun Network (Hong Kong) Limited
ASN-TBS-1 - Turner Broadcasting

Summary for relation

- We found some known attacker groups from PlugX samples.



WRAP-UP

Wrap-up

- Analyzed 3 types of PlugX samples
  - Implemented PlugX config parsers for all types
- categorized samples into groups based on PlugX config and code
  - Some groups are connected to known targeted attack groups
- The tools and results are available on BlackHat archives ☺


Questions?
(@cci_forensics/@herosi_t)
Please scan your badges for evaluation surveys!
black hat
ASIA 2014

References

[1] PlugX: New Tool For a Not So New Campaign <http://blog.trendmicro.com/trendlabs-security-intelligence/plugx-new-tool-for-a-not-so-new-campaign/>

[2] Tracking down the author of the PlugX RAT <http://www.alienvault.com/open-threat-exchange/blog/tracking-down-the-author-of-the-plugx-rat>

[3] ETSO APT Attacks Analysis <http://image.ahnlab.com/global/upload/download/documents/1401223631603288.pdf>

[4] Poison Ivy: Assessing Damage and Extracting Intelligence <http://www.fireeye.com/blog/technical/targeted-attack/2013/08/pivy-assessing-damage-and-extracting-intel.html>

[5] PlugX "v2": meet "SController" <http://blog.cassidiancybersecurity.com/post/2014/01/PlugX-v2%3A-meet-SController>

[6] PLUGX - PAYLOAD EXTRACTION <http://www.contextis.com/research/white-papers/plugx-payload-extraction/>

[7] plugxdecoder <https://github.com/kcreyts/plugxdecoder>

References (Cont.)

[8] The Chinese Malware Complexes : The Maudi Surveillance Operation

<http://normanshark.com/wp-content/uploads/2013/06/NormanShark-MaudiOperation.pdf>

[9] Alleged APT Intrusion Set: “1.php” Group

<http://www.zscaler.com/pdf/technicalbriefs/tb_advanced_persistent_threats.pdf>

[10] Jul 5 CVE-2010-2883 PDF invitation.pdf with Poison Ivy from 112.121.171.94 | pu.flower-show.org

<http://contagiodump.blogspot.jp/2011/07/message-targeting-experts-on-japan.html>

[11] The Sykipot Attacks

<http://www.symantec.com/connect/blogs/sykipot-attacks>

[12] Poison Ivy: Assessing Damage and Extracting Intelligence

<http://www.fireeye.com/resources/pdfs/fireeye-poison-ivy-report.pdf>

[13] Internet Infrastructure Review （IIR） Vol.21

<http://www.iij.ad.jp/en/company/development/iir/pdf/iir_vol21_EN.pdf>

References (Cont.)

[14] APT1: Exposing One of China's Cyber Espionage Units
<http://intelreport.mandiant.com/Mandiant_APT1_Report.pdf>

[15] “Winnti” | More than just a game - Securelist
<https://www.securelist.com/en/downloads/vlpdfs/winnti-more-than-just-a-game-130410.pdf>

[16] The rush for CVE-2013-3906 - a hot commodity
<http://www.securelist.com/en/blog/208214158/The_rush_for_CVE_2013_3906_a_hot_commodity>

[17] Exploit Proliferation: Additional Threat Groups Acquire CVE-2013-3906
<http://www.fireeye.com/blog/technical/cyber-exploits/2013/11/exploit-proliferation-additional-threat-groups-acquire-cve-2013-3906.html>

[18] Command and Control in the Fifth Domain
<http://www.commandfive.com/papers/C5_APT_C2InTheFifthDomain.pdf>

[19] Operation “Ke3chang” - Targeted Attacks Against Ministries of Foreign Affairs
<http://www.fireeye.com/resources/pdfs/fireeye-operation-ke3chang.pdf>

References (Cont.)

[20] THE 'ICEFOG' APT: A TALE OF CLOAK AND THREE DAGGERS <http://www.securelist.com/en/downloads/vlpdfs/icefog.pdf>

[21] Khaan Quest: Chinese Cyber Espionage Targeting Mongolia <http://www.threatconnect.com/news/khaan-quest-chinese-cyber-espionage-targeting-mongolia/>

Appendix – Sample Hashes (Only Samples with Config)

sample1	1d942ec52f42651139df13e837854c2cbd76880d12b513ad3d1550b4b6da8488
sample2	1f19b9fcbc3b6d09f48a4764eeb7b9e52016ff038bc293ab90f8b34603333928
sample3	b05ddea622e6a00c43ee9b20958cba2dc22f89b8a5ebece7f28443332c77196f
sample4	1acd0212ee970abfea717c2eac9a73613787f86dab42298f1d74a06f79554f0f
sample5	f75ddb8104bd84b15c1bc9fae54d6a0da809ad001fc9e5c76ab2e733ccb684d0
sample6	4aefdb49c2722358e0479fc91e6e1a45863cd560653f404b57f3176f3048c9d0
sample7	cc1ea63125f4564523af3803db42c981ce306fac5f1a187ffd3a86dbc1cca06f
sample8	bf630442adaced2410a6f1962d76f3f8948a5a810d92001bb11f09aa1d01f9ac
sample9	ab1e5e0c91d950cd07d71c17af746697b04dd1906b6f9137445a28670d5bbb76
sample10	22f9f267b6f5d608892a1bf8a95f8d5adf8172b31f1434b601406fce038211ba
sample11	29305867194O186b60ba1eebaeeb9ff2fdca824c93fee7a936f90698a993865d
sample12	a5db8938ef9627ced151ba32dda949cb7d08d38297e2595073b62db97e9f6b03
sample13	70b3508af1602207bf6ed1f212b9758c422fa72dc25b2f000b1780b5f940044d
sample14	2aa4aee13f057e9f13be2aa25f6df662b447fdf10230ee013f7e734b9b8fbfbc
sample15	c5d23f066a4f51149211caecae50c6f602a520557bdb441e787bc4506217bb32
sample16	71323668c2993892445e286e028db05ca3cdc582140c28d5d044aa7eccd9ac29
sample17	71f7a9da99b5e3c9520bc2cc73e520598d469be6539b3c243fb435fe02e44338
sample18	8f638b1dee03064ce71eee889ce9adf55e9ce9c16f0694929d1d108fa0d30c11
sample19	90a5c1c5dc2278063478fbc8f2ac072ccf0489d7b3f81a6ed35b7d712b4b7b84
sample20	e6940c142f3bed04eb532e78516da195b35f9fdd77b465a979b96a74c738da0b

Appendix – Sample Hashes (Only Samples with Config) (Cont.)

sample21 2a8d18a59cd648637deb830079b460008d81411681f0eb41dc327c3f447326f6
sample22 10e20e496b4ea73c3ac656304d6add6c234a045640eb0a54b88aeb68c6c08751
sample23 67c29f047285a0401afa8cd1167344031b375fa0b11a79ce746c0ca7fa1b8196
sample24 61468e3775e2171c306d339dfd7a117405c13b4462399e8e286c3e86b8e1d3d9
sample25 66bca3f92841b7bffae4d27c3ddb5adbf8084ad40ee0edda1edc1d25f5e1b967
sample26 5dccc1a35857cb3ea2107fae171afd228c0c25da426c160c09b98c6686666966
sample27 b05c3de04eb93a6e175222117eb63e6e5894517914a7b055dcb2416cc7f6c5a8
sample28 0576bc326b3ece524867e2c9b3feff1d0a4f02a36e375a3e7b0e11b2fafc5629
sample29 f63808ddf0e0ee31c124f509a102a4624fdb919678e9186de67841a70fd5e509
sample30 e1b91899c1233a933caaf62f17338677eda79522daba41856d1e23f2d0bd625d
sample31 55db2171cdd76aec00af4f020cd099cd2dbf7afe738e75c80c94e26e9e774631
sample32 555952aa5bcca4fa5ad5a7269fece99b1a04816d104ecd8aefabaa1435f65fa5
sample33 257023f2b2082f016265070291e1eb926f3107907e1528455cbc53c573d9cbb3
sample34 c6dffd8891d2f5fb843133e5716f330774b8ca48b4954dbff57e0d2a89406a59
sample35 cb9f92eab414dd2915be0f4122387dd3270c49d84e44a0de29d22e8aa176e046
sample36 2b484ef33d09b5b0fd69dca9ce1621ff70be87dcd144b3c85749daff232b51e3
sample37 07d288df69d49f48d1ac38c29a45f1f229817c3889f352e766c66eea72d1942e
sample38 803fe2bce423bf3100ed03c53cf71fbacba05d096db0dbcedce13f80bd58e6fc
sample39 4956e7f345d1e235c7088c950fb5daf7936204ee443e74871e45360291b3343a
sample40 8865387f29f8603ef4e74d2ab25afb74f2725266b03991d2cc0e7d9316a460de

Appendix – Sample Hashes (Only Samples with Config) (Cont.)

sample41 b6999f03c228e11601d09055fccbd8513c0fb596ab24c10b0c7c33f7c60b7665
sample42 9bfcaeb45cf9f1d661871013d4c387462186fff3447a63b3e653c0df70855e96
sample43 50d630d2f194c804bfd33130b895a4bba8241c1bbe9e6b4d8ec574c38923e724
sample44 616ab9849512d35a9612e3c7f976a0517eaf18301650d977d4ea8a0c1d7b9ae9
sample45 48742c093f566bea237ffe7e66f2dcafc7c27950e5e5e8a34ed2308e6e9213a5
sample46 a003c5353c9abb01ca9a243bca467b4e9f6067c88471924d5f800fc10b3dc59b
sample47 4f7590d4268af785ccd289d634ff6074815e0835c9e6e45756d7b9f3c526b159
sample48 8dd55b5878b65953cb1e0cd523f1aa3a826fbca16ff070a5459d0af563acbee5
sample49 08665a8838adf7e04c1154daf6ca4b9aaf7ad6f179288415c707b7566cc11ca6
sample50 074960dcfe2307ffc245e756d461c84053ca4d5d4071623d39289bef5402587b
sample51 93447e5f1b1a58b191b9c5043e72222772c1992f7786191e71bfe4d60d6d6543
sample52 c8a26b8c3367a6d08a88239ea01df9565f4b9bf6acc50eab3c806f5423be924e
sample53 c729089055c1b2e0e13627878e9aaa319198c9f70fb247fc7d52e4071b4cd067
sample54 c6b211edee54299aaaf234c9679932bd3b4836e010a0fbf46293fa7e2faf766d
sample55 a3c4cb110064086fd7491d9cf5ffd7552384916c92effca20c8b16dfc625f37b
sample56 32d234535e3a950eb89d83a4132f24f37490354dd4008f317df541febfdb516b
sample57 c11ac6ecb48b37ff0afb2f1fdadd4e3905c2cfa98766084c470b74b33da51550
sample58 f88660de3151900ccf25bc11fc24de17e0d40128a1aff73c561c9cb4a559cac6
sample59 166b489bea58e877a8f97eac6c9ef6d5db08dd5ff87392657f8400a608dda6b1
sample60 d5852ea9026098196ae02c33280192e51a6c0d284ba07fb1578ad3e089b8ddab

Appendix – Sample Hashes (Only Samples with Config) (Cont.)

sample61 df1f547cdc627d1651bcf52baa74f30455f94a2ae1d76e900eb3c8b84bb99383
sample62 dfd13c1d72399bae6ce3910e269e45f9bc717de605e64dff326c86172c4d5a98
sample63 64e257b7a6b22487a955180e0be313c61c385f54f4753fc81758c80b50c52bb0
sample64 96c577a5e2d09d297e8527b3bf2a0d8fafdc922884a7924d7339663b92af37cf
sample65 06bf32e95d25508f7bc8217e4d0de5057dc1109d8cf5b94a3cb60e2fbce58774
sample66 55b352ad38782f834170c5d33dba0342b27f1623985b987a345ac216e43fd2aa
sample67 4397db55354059398731 75e583c767b209fe9dc365931b60800f3ece279c46b2
sample68 a99b37c27b74eebdbf90992303eb573d6b406eca0840f2e8690d9f5e91b502a0
sample69 2cae4c0cd1e7d7a87d6a10735970dafc50344df3042957c00bbc0cce1d02a602
sample70 b5b625c18f9cdc53ea080a1726cf7222f6f8578849481f5a44bcfae4283af283
sample71 de09b7ae5e1899b04bddb8476ccd8f0633ed244f0b9875b14881b93d820a6396
sample72 4e96d174595200929d5e4c6ba64b5cc13a03c7c606534ce32fe007abbb0d9355
sample73 98dfa9713aa0030b1fcebd34c5a1dc8e76635694548bb683b3d8043c123f0b11
sample74 0ccdc6dacb13a92f95b9824315158715ba269afdf1778ba20447aab3716e1728
sample75 9282e85300e1dacd9c388760459612bb2a6e516cad0e1836ba4916e97a271507
sample76 6199613fd468919293a33ccc9894febf82b6c71df062b960ca82b0c68b160d2e
sample77 a6d3f5f6d3ae1c307184d92b313c5cd321da5fdcc7771cc532e53b9950daa060
sample78 e1cdada171022961c25a58783403e7dac07968514dace472bd4e82fe60d6fa35
sample79 2bc5ce39dd9afe2157448d3f6d8cb9c549ed39543d159616e38480b9e6c11c49
sample80 d85c72ad2016a5e1a858b5a1b67316873927e13f9ff52b8456852c5758736f26

Appendix – Sample Hashes (Only Samples with Config) (Cont.)

sample81 7f305dcd024ed7f5a6f7359d0baa35f06d90ae4cb8c35ffed65ed5c651fb0577
sample82 cb93e86cbb7b048ef2f3607bc97225a03f0143424247913ad903dedadcaebc57
sample83 4234f27a48a440d46def787f4fbae259da9e53d5dc976e2d1eaf6f4c5fc1623c
sample84 413bad05ff92bb9895cae3db0e9cbacb4a9f735970044b4b6c56cfb737f154ba
sample85 c6b74ff3409dd91f62677e4da52f77a2810c539a161efb768e86ad31f9500e10
sample86 881359a2e2d0294f639caf19336d16c6c6447cbfb778b669ccdf3ffba61a1d04
sample87 3356b9dea77d333af9b7c4ac72a3433a9ee4b74f34e84ccdb26c56a53609fd85
sample88 70dc54ea5a4f2c2815fb1b60c0ee70d316798a6a4adbc2476a4bdbdccb07ec76
sample89 a832e6cd838389901e4e229ea4c662be81114fc60506d537d66760ed46cbdb80
sample90 d29caef55fa0eae09838176dc29bd4313e04113061d194ebb5d8b3b23b04ca45
sample91 6658aefa05c72fb361cd8710bf442ef7f41a1d6cf6315ca68438fcbd36df3080
sample92 6895c0875c627401d00857d426ad9f222075d5ff5f0a217e9de91424a777acdc
sample93 445288ad6d9b7b2c11bd572bc87d248146274502a0557cabf63a89d58869f581
sample94 60fbea9897389c9dc1bb19fae2de4fe485797dd7958e09ac211137c7ae87a68b
sample95 73525df5a5b7e12a49d045f92cbb09f87d679dbe3156b9bc5871346e1ddc2d2a
sample96 349e19c3a2e3f4071da8b32a052c93b56aa5f4307ee5c07f722bdcc243d02bd2
sample97 9cb2b1a45e2e427e73b6decc9cbe76790877b052d6053ef90c1dd42d0aab7abc
sample98 30a0ac4f02caf9acc6b29a1e9419352a56be9d84d1b534389cf3391b19868347
sample99 9399d2cae2218d94dde1ed314a9c3b7f5a12e7e94774198a2338ff368f8a6aa8
sample100
647f5f3cc8b07f3f87e4c8ae6f63bd843f8c97d86917f4501dd3e0fcf1649c95

Appendix – Sample Hashes (Only Samples with Config) (Cont.)

sample101 ab8654480201f0f74126fe6373f7b1605f0142fe53b0aabadaa1e0cef8da6c45
sample102 65c710188b3b19309d75848212e921c57b0a7784ca5d6affa08acfeb7c7150dd
sample103 2afa3a7f029f857ed0cda12e84ed4b51f41fba3e320f2d5f4d51c88685c5de11
sample104 39503394318533b7631cea7b1f19a86f8e4362a672e5f0df322d3ad761350136
sample105 4d464f9def2276dac15d19ccf049b7c68642290bc0e345e06d4b6e9103fde9e6
sample106 65bbf0bd8c6e1ccdb60cf646d7084e1452cb111d97d21d6e8117b1944f3dc71e
sample107 8446c945371f94eefa5b56ae17c4bf7c9c13dd37a38e158658fb8ef89e11b5a2
sample108 fe4985b13b2270c0e71a2c0755a22c17bba968ac66b94899fe6dccc22aacbd54
sample109 a78dbafaca4813307529cafbed554b53a622a639941f2e66520bbb92769ee960
sample110 a08caf2d12e1405e615b9dfffca71fa8444347e4a09f43d981c387f14ef18d82
sample111 abb466652a24d2525d0a348bca5e000a83272ed4a862cc545075801fd5ff5ad5
sample112 abce952d75a4ffa93a42dcce883b48ee85cd574793feac0b1442d5b990ea822d
sample113 861b0fc01cf05c8556fb2037b6e1f59b6ca9f4f69a93b0aa8c477c53aa13d1b2
sample114 5889d81082283c087d53d45d9092e552c74a9b446b78579989c1cb4d5e574362
sample115 82c49e96a8f625059084a7519be94fb67c7581a8ea5f39c08c77943abb719fc2
sample116 991a2a75e18a2421cf9f887a4fe83d132de6ab73c15b97bd030b8d6037591b1a
sample117 95852da6976c0b3f46eac1988490edd3a0b3e9165c17e3a6e934fd4f899fa204
sample118 20ba00b8df292fdb29beb692d4c8558fffb3f01e9b19d1037a28b3675e09d5e7
sample119 882d79db4eeddd403605739ce14f0283752762847fc0535249140ea90a3e96e9
sample120 542b8483008b476dcb86874c8dd906a72ac12883a5cde2b395f6ec19cb7f02b8
sample121 5146a8483b7a947a193dedd8a840635d2bdd42d384835d8270de7cd38c82d25f
sample122 8a1a8009b19dd08677209ef54e7ec71e2a41ae383ebd686e95e43f091a8d0a39
sample123 98446b2e9f1b9559cf475dd175555a3557dce973d653be87262b8b8ff38e7710