ITOCHU Cyber & Intelligence Inc.

SOPHOS

HITCON '25

The Evolving Bloodline: Revealing the Hidden Operations of BloodAlchemy

Suguru Ishimaru, ITOCHU Cyber & Intelligence Inc.

You Nakatsuru, Sophos

Kiyotaka Tamada, Sophos

About Us

ITOCHU Cyber & Intelligence Inc.

Suguru Ishimaru

Executive Cybersecurity Researcher

ITOCHU Cyber & Intelligence Inc.

×

SOPHOS

You Nakatsuru

Principal Threat Researcher

Kiyotaka Tamada

Principal Threat Researcher

Counter Threat Unit, Sophos
(Formerly Secureworks)

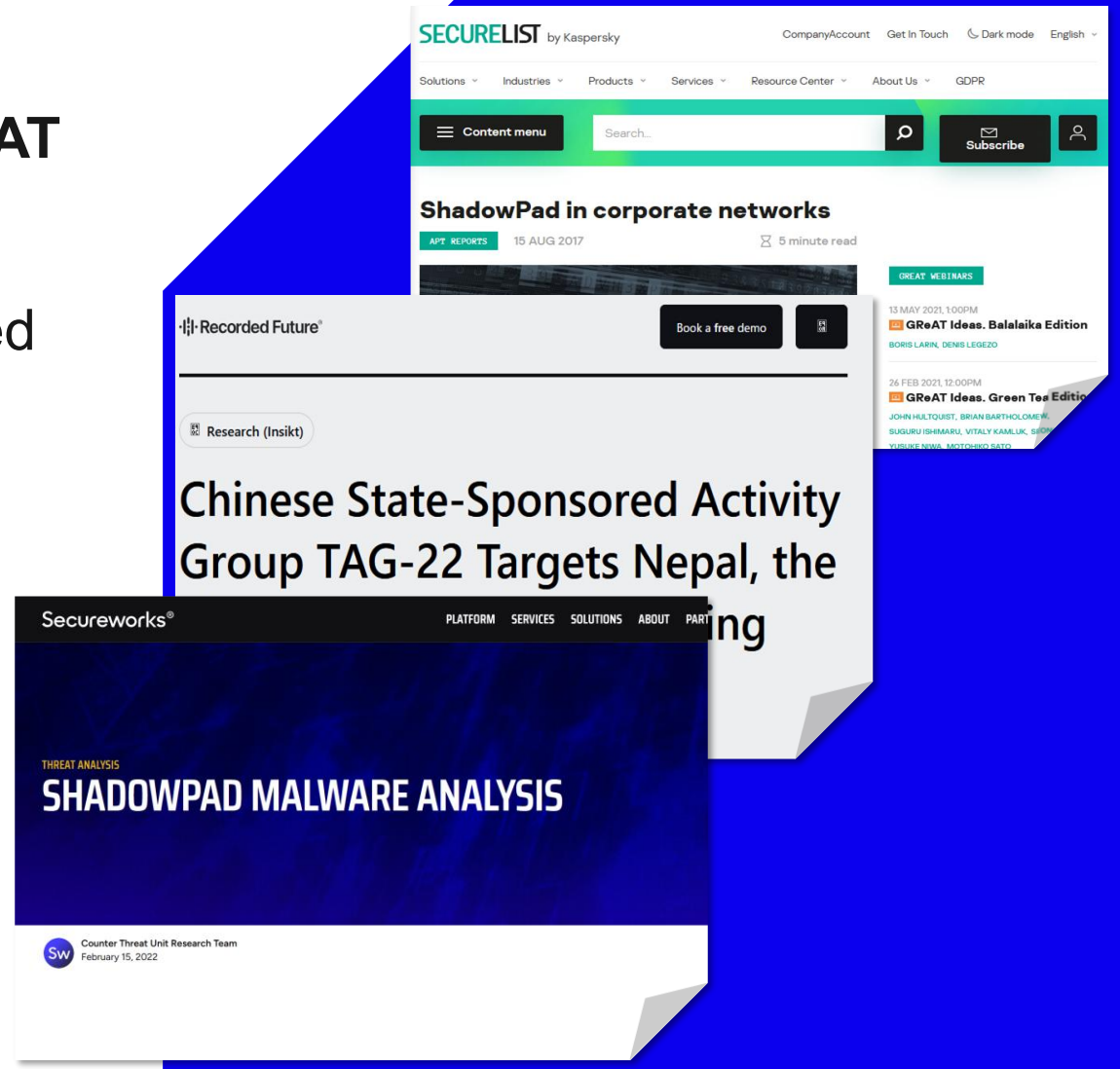
Agenda

- The Malicious Bloodline
- Diving into DeedRAT and BloodAlchemy
- Unveiling Additional Modules
- Attack Campaigns Involving DeedRAT/BloodAlchemy
- Conclusions

The Malicious Bloodline

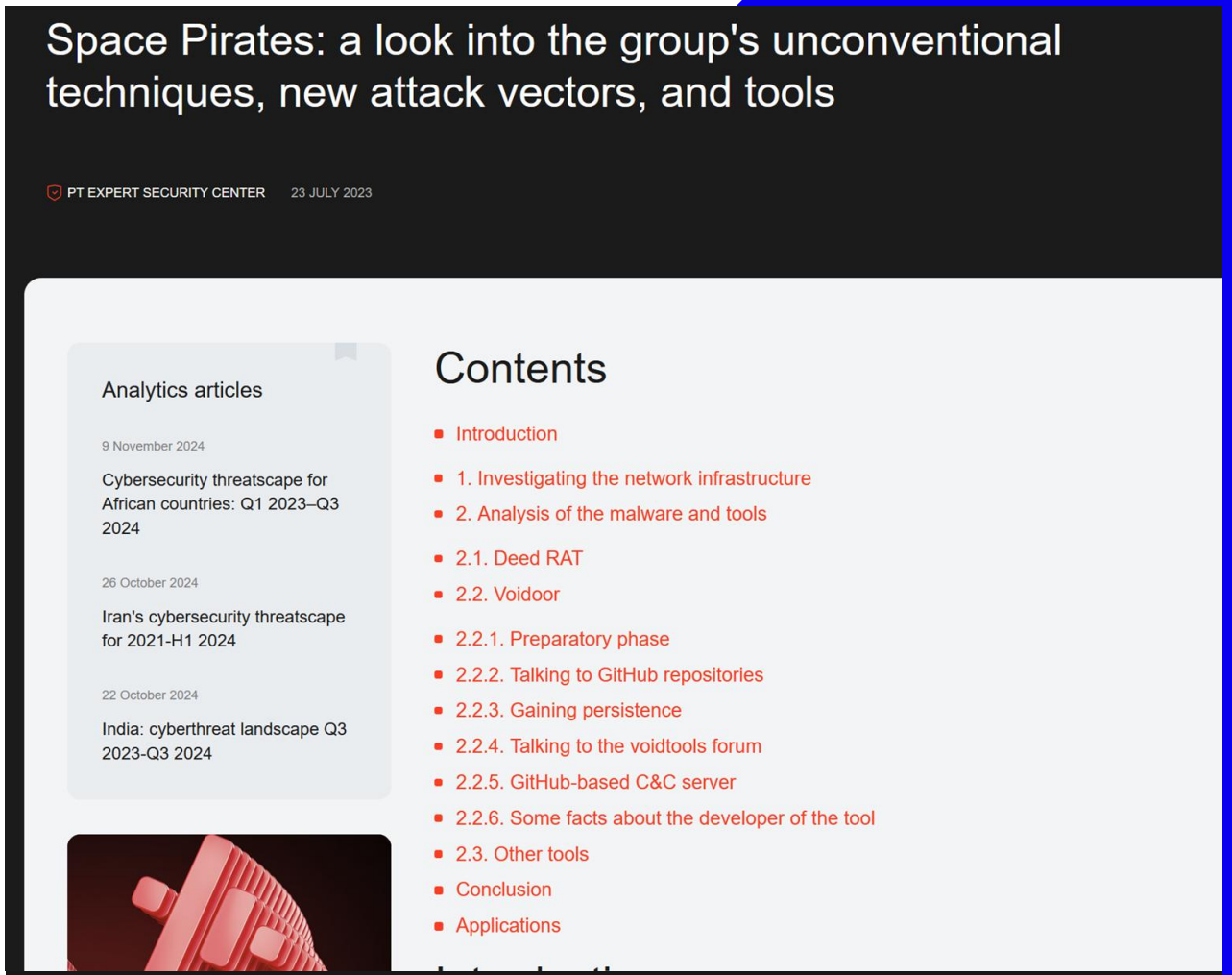
Notorious Malware ShadowPad

- ShadowPad is **a sophisticated modular RAT**
- Discovered in a supply chain attack in 2017
- Since 2020, It has been reported as a shared malware by **multiple Chinese-speaking actors**:
  - BRONZE ATLAS (APT41)
  - BRONZE BUTLER (Tick)
  - Tonto
  - TropicTrooper
  - Etc...



DeedRAT Overview

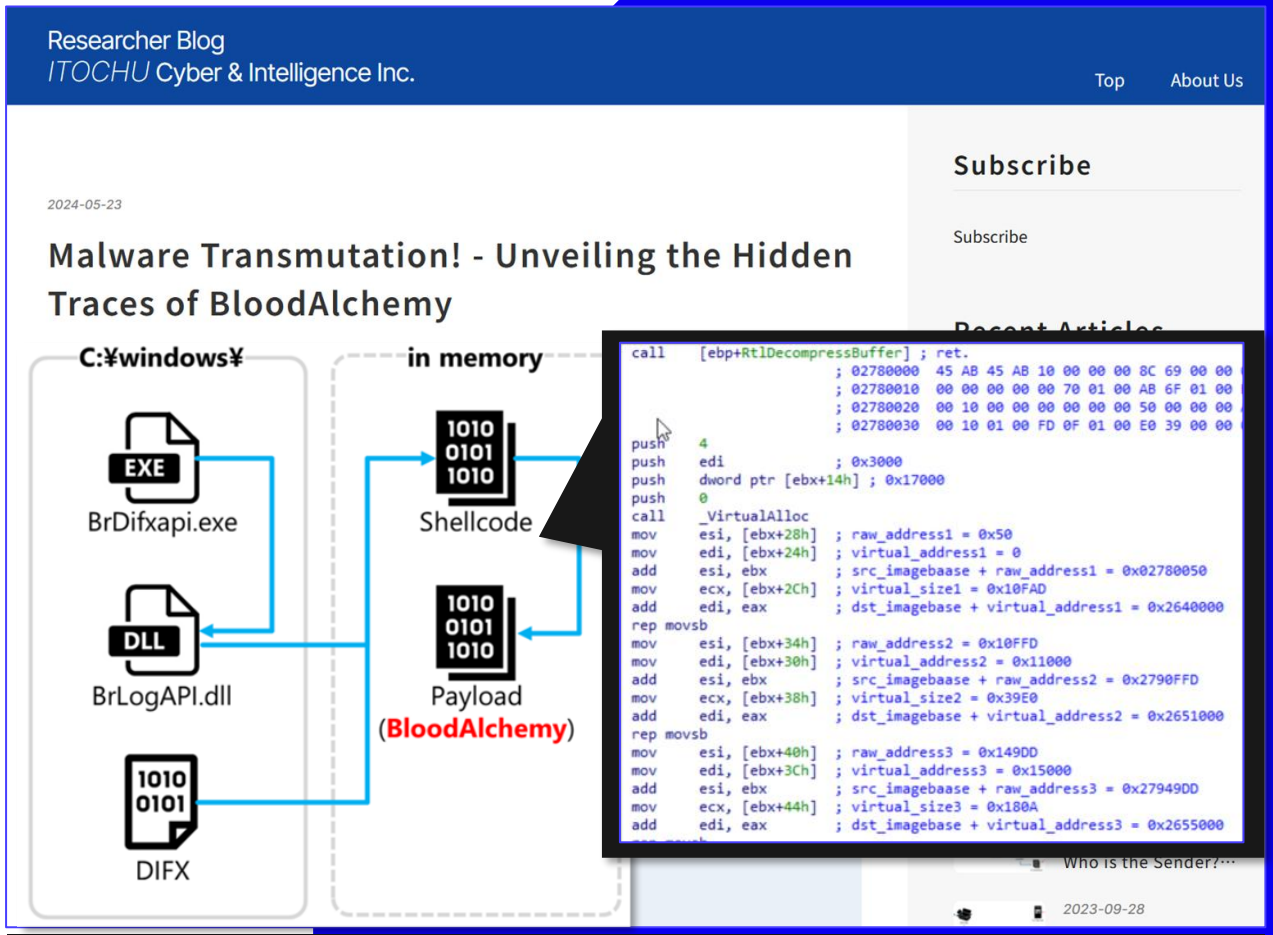
- DeedRAT (a.k.a SNAPPYBEE) is **a variant of ShadowPad**
- Published by Positive Technologies in 2022 and has been observed since 2019
- The name DeedRAT originates from the magic bytes **0xDEED**
- Used in several campaigns linked to Chinese-speaking actors



https://global.ptsecurity.com/analytics/pt-esc-threat-intelligence/space-pirates-a-look-into-the-group-s-unconventional-techniques-new-attack-vectors-and-tools

BloodAlchemy Overview

- BloodAlchemy is **another variant** of DeedRAT/ShadowPad
- First reported by Elactic Security Labs in 2023
- It is **exceptionally rare to find**
- The name BloodAlchemy seems to originate from the magic number 0xAB too
- A C2 overlaps the IoCs of Earth Estries

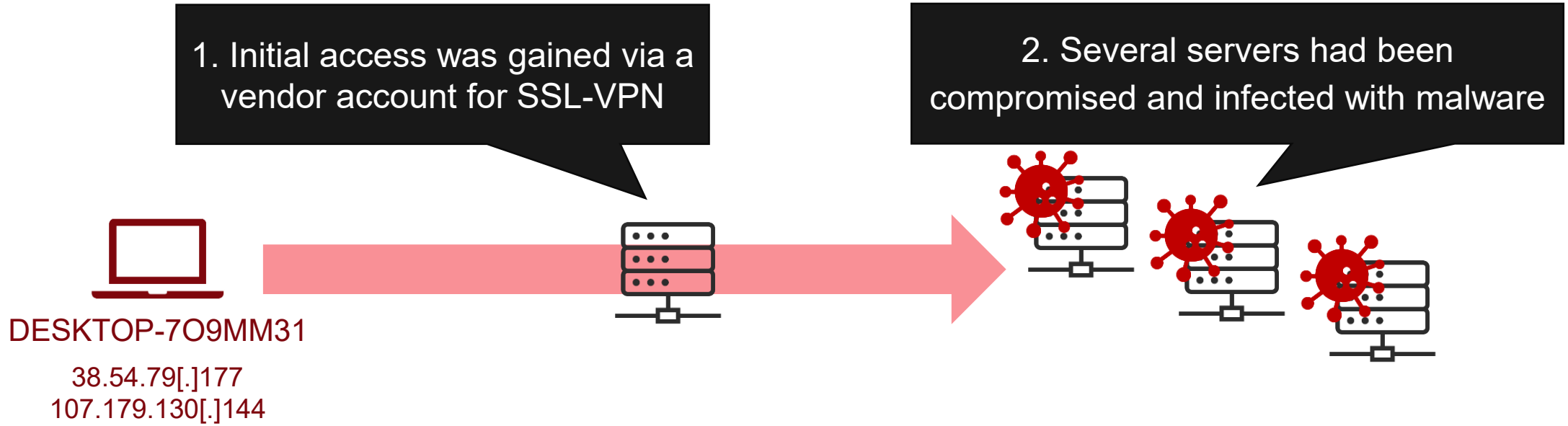


https://blog-en.itochuci.co.jp/entry/2024/05/23/090000

The Beginning of Our Investigation

Discovering the unknown entity called BloodAlchemy

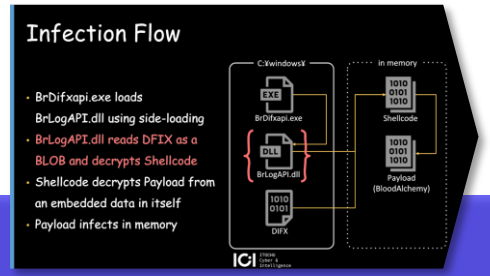
- In May 2023, we observed an APT activity and investigated it deeply
- **BloodAlchemy** was found from the one of the compromised servers

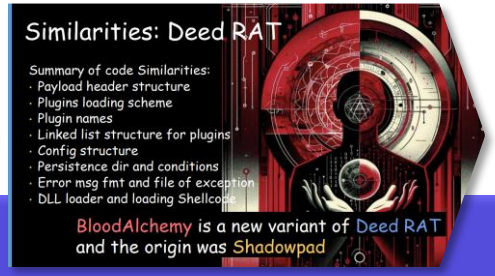


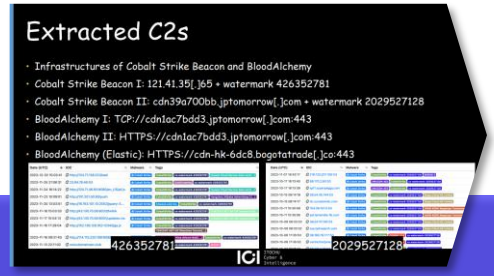
Previous Research on BloodAlchemy

https://www.botconf.eu/wp-content/uploads/formidable/2/BOTCONF2024-NiwaIshimaruSato-IntotheVapor_ICI.pdf

- The details of the initial infection vector
- Analyzed the features and characteristics of the BloodAlchemy
- Revealed it as **a variant of DeedRAT, originally derived from ShadowPad**
- Tracked the actor's infrastructures to identify potential C2 servers



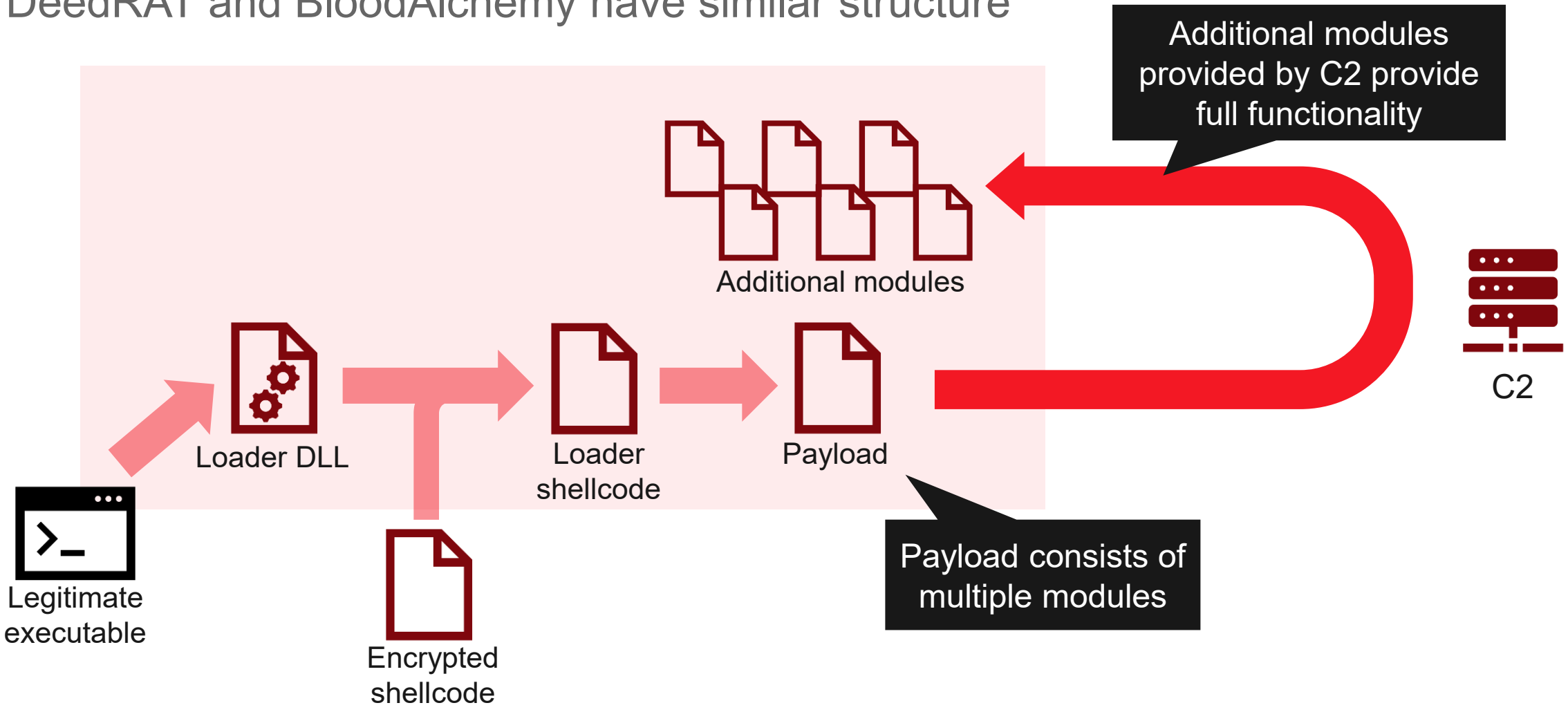




Diving into DeedRAT and BloodAlchemy

ShadowPad Modular Malware

DeedRAT and BloodAlchemy have similar structure



Known Similarities – Module Header, Code, etc.

See: https://blog-en.itochuci.co.jp/entry/2024/05/23/090000

DeedRAT

offset	descriptions	data
0x00	magic number	54 45 ED DE
0x04	plugin id	0x20
0x08	entry point	0x1740
0x0c	original base	0x400000
0x10		
0x14		
0x18		
0x1c		

```
// %ALLUSERSPROFILE%\error.log
// fmt:
// %4.4d-%2.2d-%2.2d %2.2d:%2.2d:%2.2d Exception Address: 0x%p, Code: 0x%8.8x
int __stdcall c_write_error_log(_DWORD **a1)
{
  void (*v1)(void); // eax
  char v3[1024]; // [esp+0h] [ebp-418h] BYREF
  __int16 v4[8]; // [esp+400h] [ebp-18h] BYREF
  int v5[2]; // [esp+410h] [ebp-8h] BYREF

  v5[0] = 0;
  v5[1] = 0;
  if ( getprocaddrbyhash(&p_GetLocalTime, &e_GetLocalTime) )
    p_GetLocalTime(v4);
  dec(&fomat__4_4d__2_2d__);
  v1 = p_user32_wsprintfA(v3, v5[0], v4[0], v4[1], v4[3], v4[4], v4[5], v4[6],
  v1();
  if ( getprocaddrbyhash(&kernel32_OutputDebugStringA, &e_outputdebugstringA)
    kernel32_OutputDebugStringA(v3);
  write_error_log(v3);                          // %ALLUSERSPROFILE%\error.log
  c_c_localfree(v5);
  return 0;
```

BloodAlchemy

offset	descriptions	data
0x00	magic number	45 AB 45 AB
0x04	plugin id?	0x10
0x08	entry point	0x698c
0x0c	original base?	0x400000
0x10		
0x14		
0x18		
0x1c		

```
// %ALLUSERSPROFILE%\error.log
// fmt:
// %4.4d-%2.2d-%2.2d %2.2d:%2.2d:%2.2d Exception Address: 0x%p, Code: 0x%8.8x
int __stdcall c_write_error_log(int **arg0)
{
  int ModuleHandleA; // eax
  int v3; // [esp-4h] [ebp-424h]
  char v4[1024]; // [esp+4h] [ebp-41Ch] BYREF
  int a1[3]; // [esp+404h] [ebp-1Ch] BYREF
  __int16 v6[8]; // [esp+410h] [ebp-10h] BYREF

  memset(a1, 0, sizeof(a1));
  dec__0(a1, &fomat__4_4d__2_2d__, 0x62u);
  p_kernel32_GetLocalTime(v6);
  j_memset(v4, 0, sizeof(v4));
  v3 = **arg0;
  ModuleHandleA = kernel32_GetModuleHandleA(0, (*arg0)[3]);
  p_user32_wsprintfA(v4, a1[0], v6[0], v6[1], v6[3], v6[4], v6[5], v6[6], Module
  kernel32_OutputDebugStringA(v4, v3);
  write_error_log(v4);                          // %ALLUSERSPROFILE%\error.log
  j_memclear_localfree(a1);
```

Another Examples of Code Similarity

Initial code of C2 main routine

Module entry

DeedRAT

```
lea     ecx, [ebp+recv_data]
call    aa_zeromemory_0
and     [ebp+netsocket_class], 0
call    aa_SetErrorMode_SEM_FAILCRITIC
call    aa_SetErrorMode_SEM_FAILCRITIC
call    aa_SetErrorMode_SEM_FAILCRITIC
call    aa_check_working_hours
lea     ecx, [edi+aa_network.c2_url] ;
mov     edx, ebx        ; lpString2
call    aa_lstrcpyA_1
cmp     [ebp+proxy], 0
jz      short loc_20101E
mov     edx, [ebp+proxy]
lea     ecx, [edi+aa_network.proxy]
```

```
loc_1903F4:                             ; CODE XREF: aa_init_module
                push    offset aConfig  ; "Config"
                lea     ecx, [ebp+lpString2]
                mov     [ebp+lpString2], esi
                mov     [ebp+var_4], esi
                call    aa_decode_string_0
                mov     edx, offset aLstrcpyw_0 ; "lstrcpyW"
                mov     ecx, offset lstrcpyW
                call    aa_get_proc_address_kernel32_0
                test    eax, eax
                jz      short loc_190426
                push    [ebp+lpString2] ; lpString2
                push    [ebp+lpString1] ; lpString1
                call    ds:lstrcpyW

loc_190426:                             ; CODE XREF: aa_init_module
                lea     ecx, [ebp+lpString2]
                call    aa_free_8
                jmp     short loc_190461
; ---------------------------------------------------------------------------

case_2:                                 ; CODE XREF: aa_init_module
                mov     eax, [ebp+lpString1]
                mov     dword ptr [eax], 40h
                jmp     short loc_190461
; ---------------------------------------------------------------------------

case_1:                                 ; CODE XREF: aa_init_module
                mov     eax, [ebp+lpString1]
                mov     ds:offset_aa_func_0, eax
                mov     ds:module_config_func_table.get_config, off
                mov     ds:module_config_func_table.get_reg_data, o
                mov     ds:module_config_func_table.exec_c2_command
```

BloodAlchemy

```
mov     esi, ds:SetErrorMode
push    SEM_FAILCRITICALERRORS ; uMode
call    esi ; SetErrorMode
push    SEM_FAILCRITICALERRORS ; uMode
call    esi ; SetErrorMode
push    SEM_FAILCRITICALERRORS ; uMode
call    esi ; SetErrorMode
call    aa_check_working_hours
push    328h            ; uBytes
push    40h ; '@'       ; uFlags
call    ds:LocalAlloc
mov     esi, eax
test    esi, esi
jz      short loc_1F2388
call    aa_init_module_list
mov     [esi+aa_network.field_0], eax
test    eax, eax
```

```
                call    ds:lstrcpyW
                lea     ecx, [ebp+lpString2]
                call    aa_free
                jmp     short loc_1F1EA1
; ---------------------------------------------------------------------------

loc_1F1E83:                             ; CODE XREF: aa_init_
                push    8
                mov     edx, offset aBaseapi ; "baseapi"
                jmp     short loc_1F1E5C
; ---------------------------------------------------------------------------

loc_1F1E8C:                             ; CODE XREF: aa_init_
                mov     eax, [ebp+lpString1]
                mov     dword ptr [eax], 10h
                jmp     short loc_1F1EA1
; ---------------------------------------------------------------------------

loc_1F1E97:                             ; CODE XREF: aa_init_
                mov     ds:baseapi_func_table, offset aa_exec
```

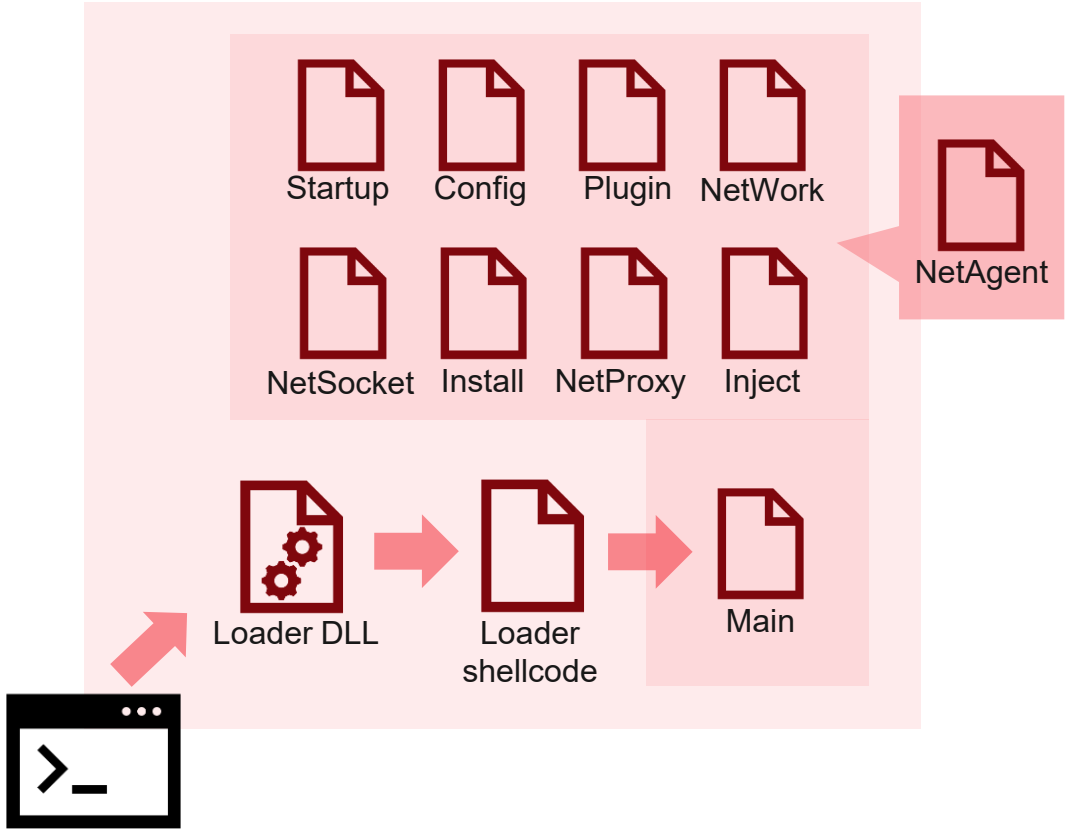
ShadowPad

```
mov     esi, ds:SetErrorMode
push    edi
xor     ebx, ebx
push    SEM_FAILCRITICALERRORS ; uMod
mov     [esp+1Ch+var_4], ebx
call    esi ; SetErrorMode_0
push    SEM_FAILCRITICALERRORS ; uMod
call    esi ; SetErrorMode_0
push    SEM_FAILCRITICALERRORS ; uMod
call    esi ; SetErrorMode_0
mov     eax, ds:aa_func_2
movzx   esi, ds:word_297042
push    SHADOWPAD_MODULE_ONLINE
mov     [esp+1Ch+var_C], ebx
mov     [esp+1Ch+var_8], ebx
call    [eax+aa_func.get_module_data]
push    ebx
```

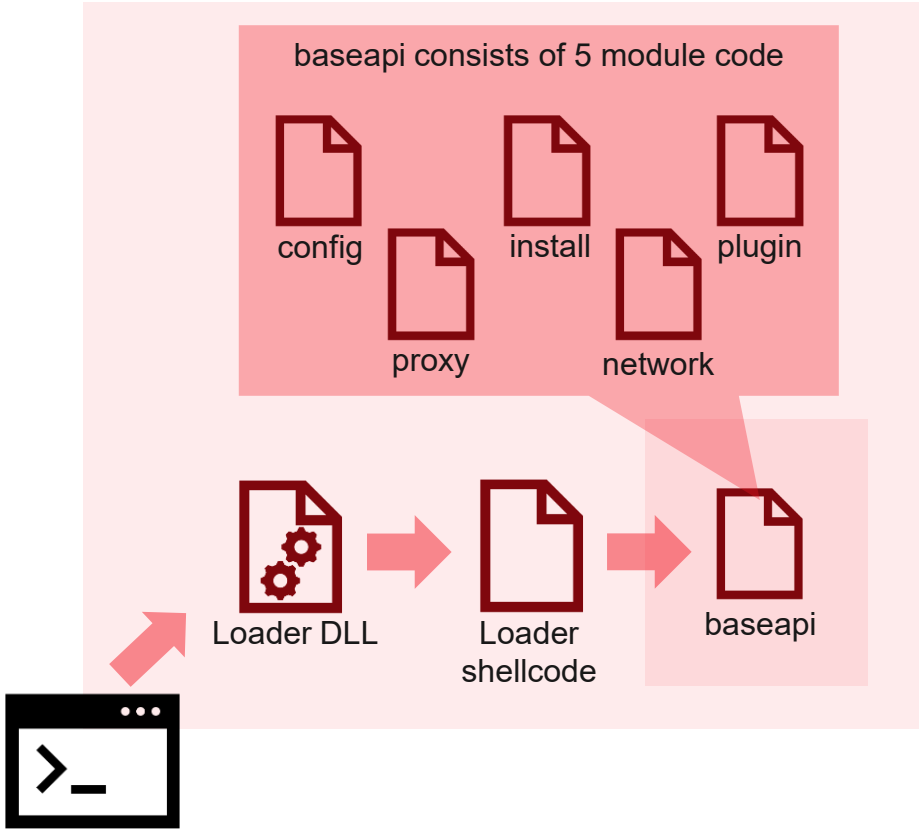
```
loc_2A110E:                             ; CODE XREF: aa_init_modul
                push    esi
                push    offset aConfig  ; "Config"
                lea     eax, [ebp+var_10]
                call    aa_decode_string_1
                push    dword ptr [eax+8] ; lpString2
                push    [ebp+lpString1] ; lpString1
                call    aa_lstrcpyW_0
                lea     esi, [ebp+var_10]
                call    aa_free_2
                pop     esi
                jmp     short loc_2A1165
; ---------------------------------------------------------------------------

loc_2A1132:                             ; CODE XREF: aa_init_modul
                mov     eax, [ebp+lpString1]
                mov     dword ptr [eax], 66h
                jmp     short loc_2A1165
; ---------------------------------------------------------------------------

loc_2A113D:                             ; CODE XREF: aa_init_modul
                mov     eax, [ebp+lpString1]
                mov     ds:aa_func_0, eax
                jmp     short loc_2A1165
; ---------------------------------------------------------------------------

loc_2A1147:                             ; CODE XREF: aa_init_modul
                mov     ds:module_config_func_table.exec_command,
                mov     ds:module_config_func_table.get_current_co
```

Multi-Module Payload v.s. Unified Payload

DeedRAT



BloodAlchemy



Different Encoding for Module (Payload and Plugins)

Both can be easily decoded, and might be samples that use another methods

DeedRAT

```
def decode_payload(ea):
    size = struct.unpack('>L', ida_bytes.get_bytes(ea, 4))[0]
    if size == 0:
        return None

    enc = ida_bytes.get_bytes(ea+4, size)
    key_num = struct.unpack('>L', enc[:4])[0]
    out = b''
    for c in enc:
        key_bytes = struct.pack('<L', key_num)
        k = 0
        for j in key_num:
            tmp = (0x401 * (k + j)) & 0xFFFFFFFF
            k = ((tmp >> 6) ^ tmp) & 0xFFFFFFFF

        key_num ^= (((((k * 9) & 0xFFFFFFFF) ^ ((((k * 9) &
0xFFFFFFFF) >> 11) & 0xFFFFFFFF)) * 0x8001) & 0xFFFFFFFF)
        out += (c ^ (key_num & 0xFF)).to_bytes()
    return out
```

BloodAlchemy

```
import lznt1

def custom_fnv1a_32_2(value):
    values = struct.pack('<H', value)
    hval = 0x811c9dc5
    for c in values:
        hval = ((hval ^ c) * 0x01000193) % 0x100000000
    tmp = (0x2001 * hval) & 0xFFFFFFFF
    tmp = (9 * (tmp ^ tmp >> 7)) & 0xFFFFFFFF
    return (33 * (tmp ^ ((tmp >> 17) & 0xFFFFFFFF))) & 0xFFFFFFFF

def decode_payload(ea, size):
    key = struct.unpack('<H', ida_bytes.get_bytes(ea, 2))[0]
    enc = ida_bytes.get_bytes(ea+2, size-2)
    out = b''
    for c in enc:
        key = (key ^ custom_fnv1a_32_2(key & 0xFFFF)) & 0xFFFFFFFF
        out += (c ^ (key & 0xFF)).to_bytes()
    out = lznt1.decompress(out)
    return out
```

Different String Encoding

Most of hard-coded strings are encoded

DeedRAT

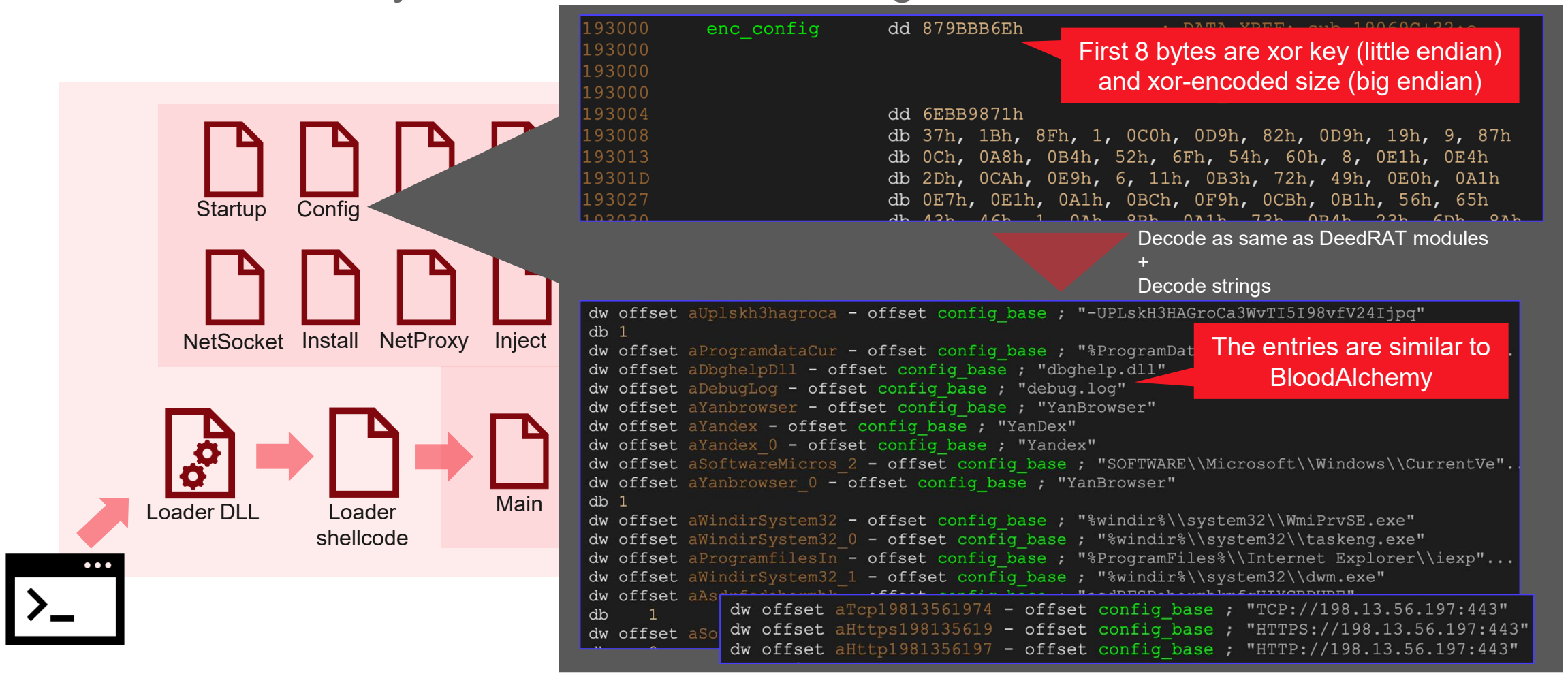
```
def ror(c, n):
    return 0xff & ((c >> n) | (c << (8-n)))

def decode_string(enc_addr):
    out = ""
    c = ida_bytes.get_byte(enc_addr)
    for i in range(4096):
        c_1 = ida_bytes.get_byte(enc_addr+i+1)
        out += chr((c ^ c_1) & 0xFF)
        if c == c_1:
            break
        rol3 = rol(c, 3)
        c = ((((c * c + rol3 * rol3) & 0xFF) ^
            ror((c * rol3) & 0xFF, 3)) + c) & 0xFF
    return out
```

BloodAlchemy

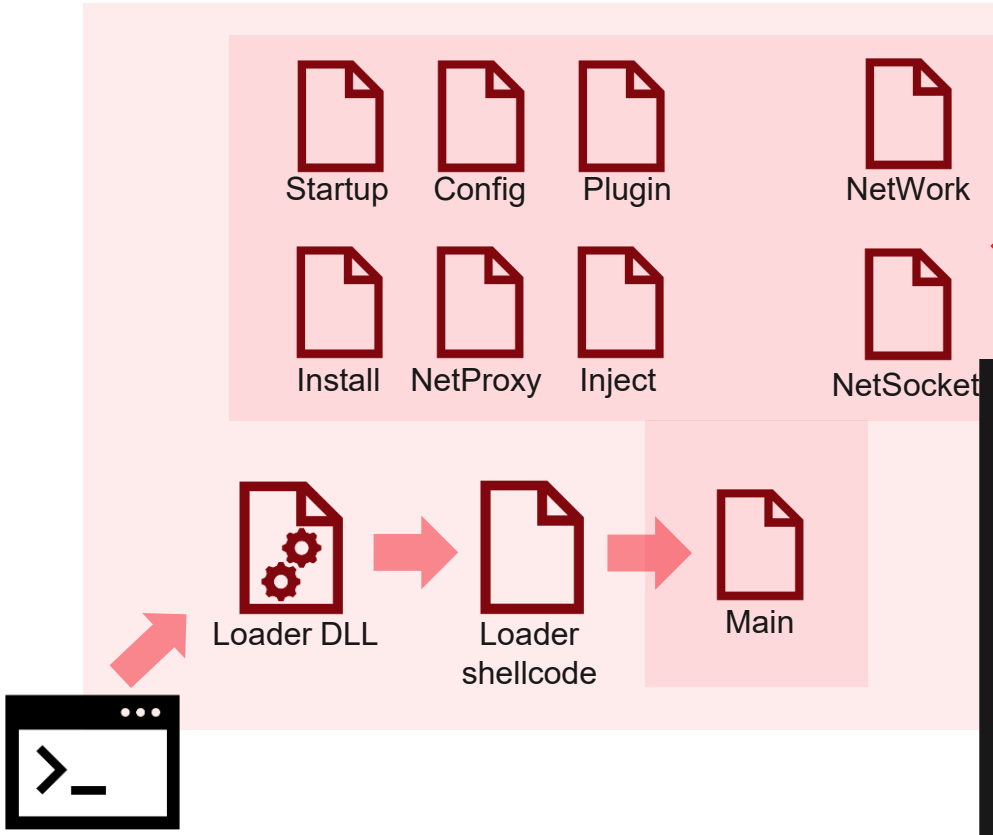
```
def rol(c, n):
    return 0xff & ((c << n) | (c >> (8-n)))

def decode_string(enc_addr):
   out = ""
   c = ida_bytes.get_byte(enc_addr)
   for i in range(4096):
       c_1 = ida_bytes.get_byte(enc_addr+i+1)
       if c_1 == 0:
           j = 1
           tmp_ea = enc_addr+i+1+j
           while ida_bytes.get_byte(enc_addr+i+1+j) == 0:
               j += 1
           if ida_xref.get_first_dref_to(enc_addr+i+1+j) !=
0xffffffffffffffff or j >= 4:
               break
       elif ida_xref.get_first_dref_to(enc_addr+i+1) !=
0xffffffffffffffff:
           break
       out += chr(c ^ c_1 & 0xFF)
       c = (c + rol(c, i%5+1))  & 0xFF
   return out
```

DeedRAT Configuration

Unlike BloodAlchemy, it exists within the Config module, not the Loader



DeedRAT Command & Control



1. Command request to the C2 (Supported protocols are TCP, HTTP, HTTPS, PIPE)

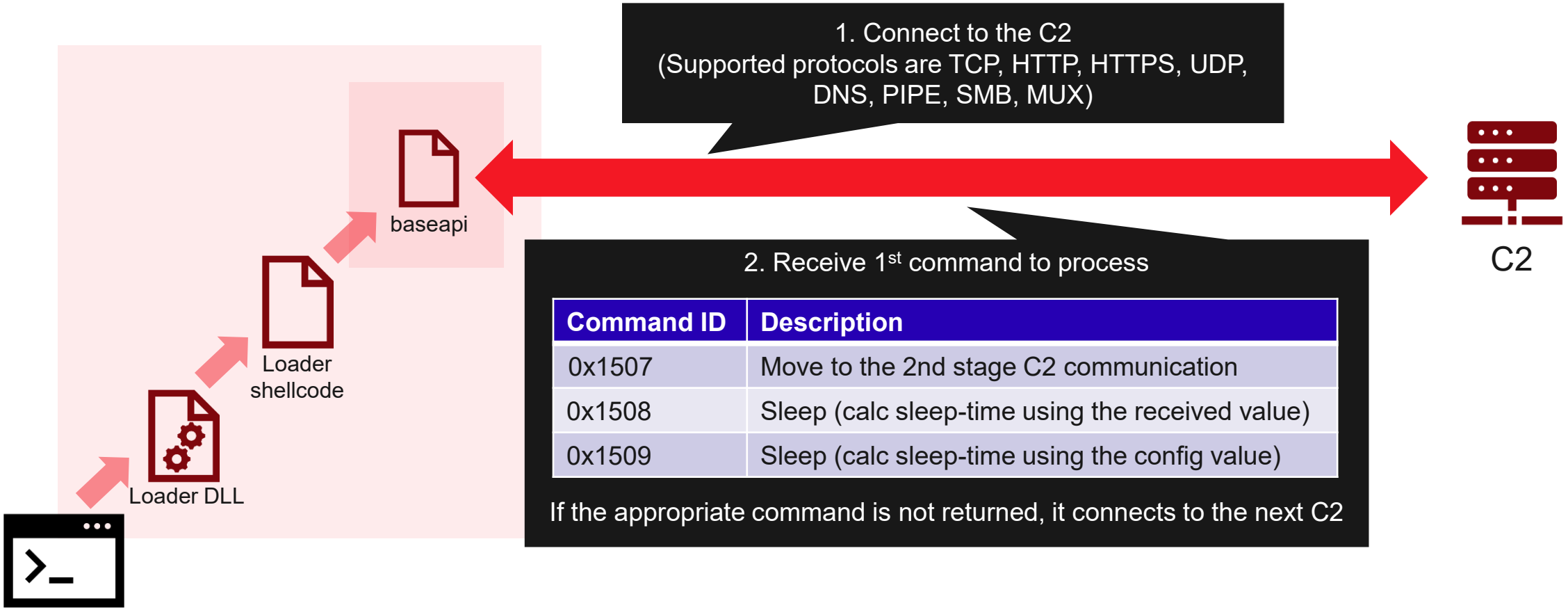
C2

2. Receive command, exec the command on each payload modules

Command ID	Module	Description
0x61	NetWork	Get systeminfo
0x62	NetWork	Start another c2 thread
0x64	NetWork	Deactive
0x40	Config	Check current config
0x41	Config	Update config
0x50	Plugin	Check current plugins
0x51	Plugin	Add plugin module
0x52	Plugin	Remove plugin module
0x53	Plugin	Search plugin module
0xA0	Install	Reinstall
0xA1	Install	Uninstall

BloodAlchemy Command & Control

1st stage – preprocessing?

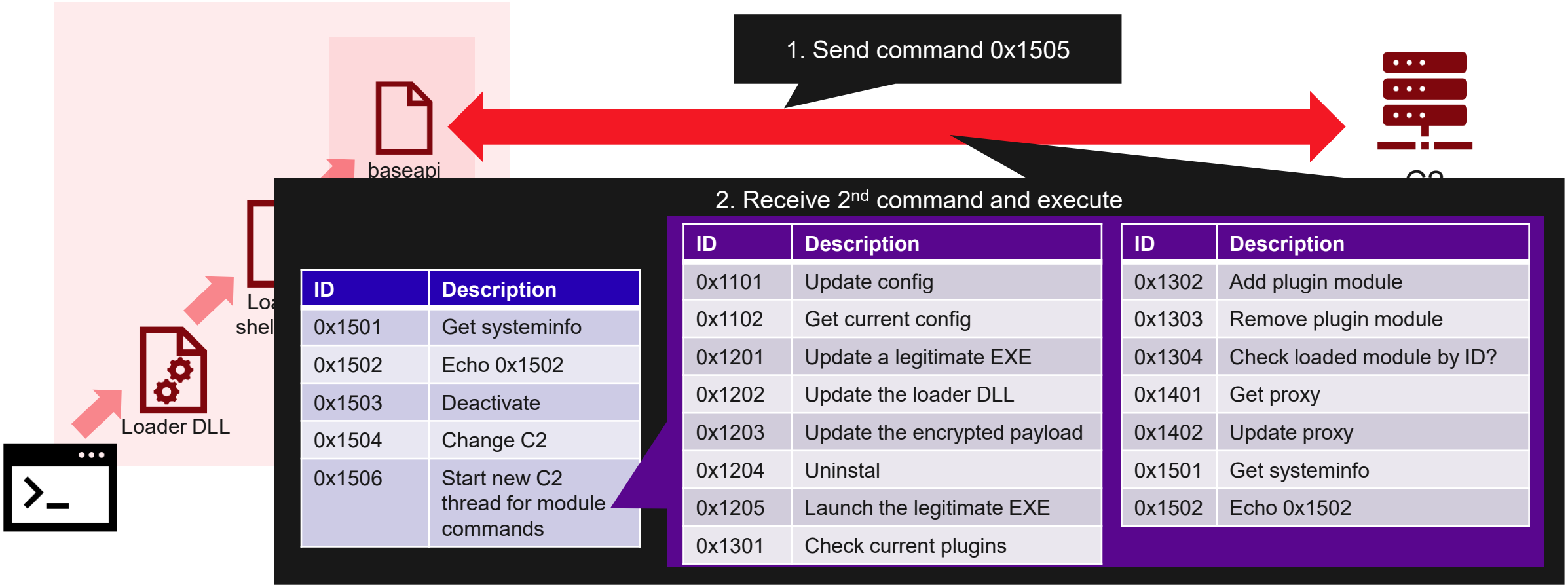


Command ID	Description
0x1507	Move to the 2nd stage C2 communication
0x1508	Sleep (calc sleep-time using the received value)
0x1509	Sleep (calc sleep-time using the config value)

If the appropriate command is not returned, it connects to the next C2

BloodAlchemy Command & Control

2nd stage – main command and control



ID	Description
0x1501	Get systeminfo
0x1502	Echo 0x1502
0x1503	Deactivate
0x1504	Change C2
0x1506	Start new C2 thread for module commands

ID	Description
0x1101	Update config
0x1102	Get current config
0x1201	Update a legitimate EXE
0x1202	Update the loader DLL
0x1203	Update the encrypted payload
0x1204	Uninstal
0x1205	Launch the legitimate EXE
0x1301	Check current plugins

ID	Description
0x1302	Add plugin module
0x1303	Remove plugin module
0x1304	Check loaded module by ID?
0x1401	Get proxy
0x1402	Update proxy
0x1501	Get systeminfo
0x1502	Echo 0x1502

In-depth Comparison Summary

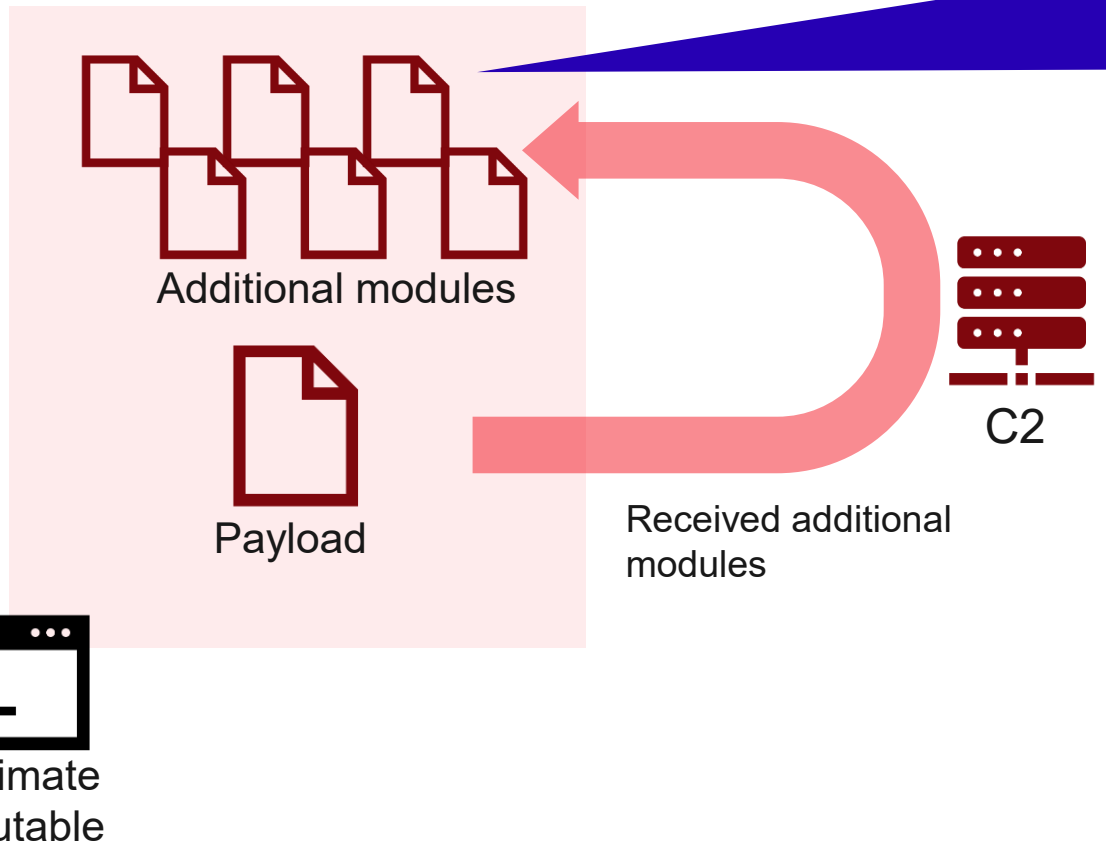
This suggests that there is a relationship between all the shadows

	ShadowPad (Casper sample)	DeedRAT	BloodAlchemy
Payload style	Multi-module payload	Multi-module payload	Unified payload
Payload modules	Root (0x64), Plugins (0x65), Online (0x68), Config (0x66), Install (0x67), HTTP (0xC9)	Main (0x20), Startup (0x30), Config (0x40), Install (0xA0), Inject (0xB0), NetWork (0x60), NetSocket (0x70), Plugin (0x50), NetAgent (0x80), NetProxy (0x90)	baseapi(0x10), config (0x11), install (0x12), plugin (0x13), proxy (0x14), network (0x15)
Payload/plugin module header	• Original module header • Also supports PE	descriptions / data magic number: 54 45 ED DE plugin id: 0x20 entry point: 0x1740 original base: 0x400000	descriptions / data magic number: 45 AB 45 AB plugin id?: 0x10 entry point: 0x698c original base?: 0x400000
Anti-analysis	None	None	Sandbox detection Debug detection
Supported C2 protocol	TCP, HTTP, UDP, DNS, HTTPS, SSL	TCP, HTTP, HTTPS, PIPE	TCP, HTTP, HTTPS, UDP, DNS, PIPE, SMB, MUX
C2 flow	Single stage communication	Single stage communication	Multi stages communication
Known algorithms used	QuickLZ	RSA, AES, LZNT1, ChaCha20	RSA, AES, LZNT1, FNV-1A

Unveiling Additional Modules

Additional Modules of ShadowPad Families

Received additional modules for enhanced



https://vms.drweb.com/virus/?i=21995049

The algorithm for loading additional modules is also similar to **BackDoor.ShadowPad.1**; however, there are new modules in this has 16 modules in total. A list of their names with codes and timestamps is provided in the following table:

Module name	Code	Timestamp
Config	0x66	2019-05-06 08:33:07
Disk	0x12C	2019-05-06 08:29:55
ImpUser	0x6A	2019-05-06 08:33:18
Install	0x67	2019-05-06 08:33:34
KeyLogger	0x132	2019-05-06 08:30:26
Online	0x68	2019-05-06 08:33:13
PIPE	0xCF	2019-05-06 08:29:11
Plugins	0x65	2019-05-06 08:33:02
Process	0x12D	2019-05-06 08:30:00
RecentFiles	0x13D	2019-05-06 08:31:23
Register	0x12F	2019-05-06 08:30:10
Screen	0x133	2019-05-06 08:30:31
Servcie (the original spelling)	0x12E	2019-05-06 08:30:05
Shell	0x130	2019-05-06 08:30:15
TCP	0xC8	2019-05-06 08:28:45
UDP	0xCA	2019-05-06 08:28:56

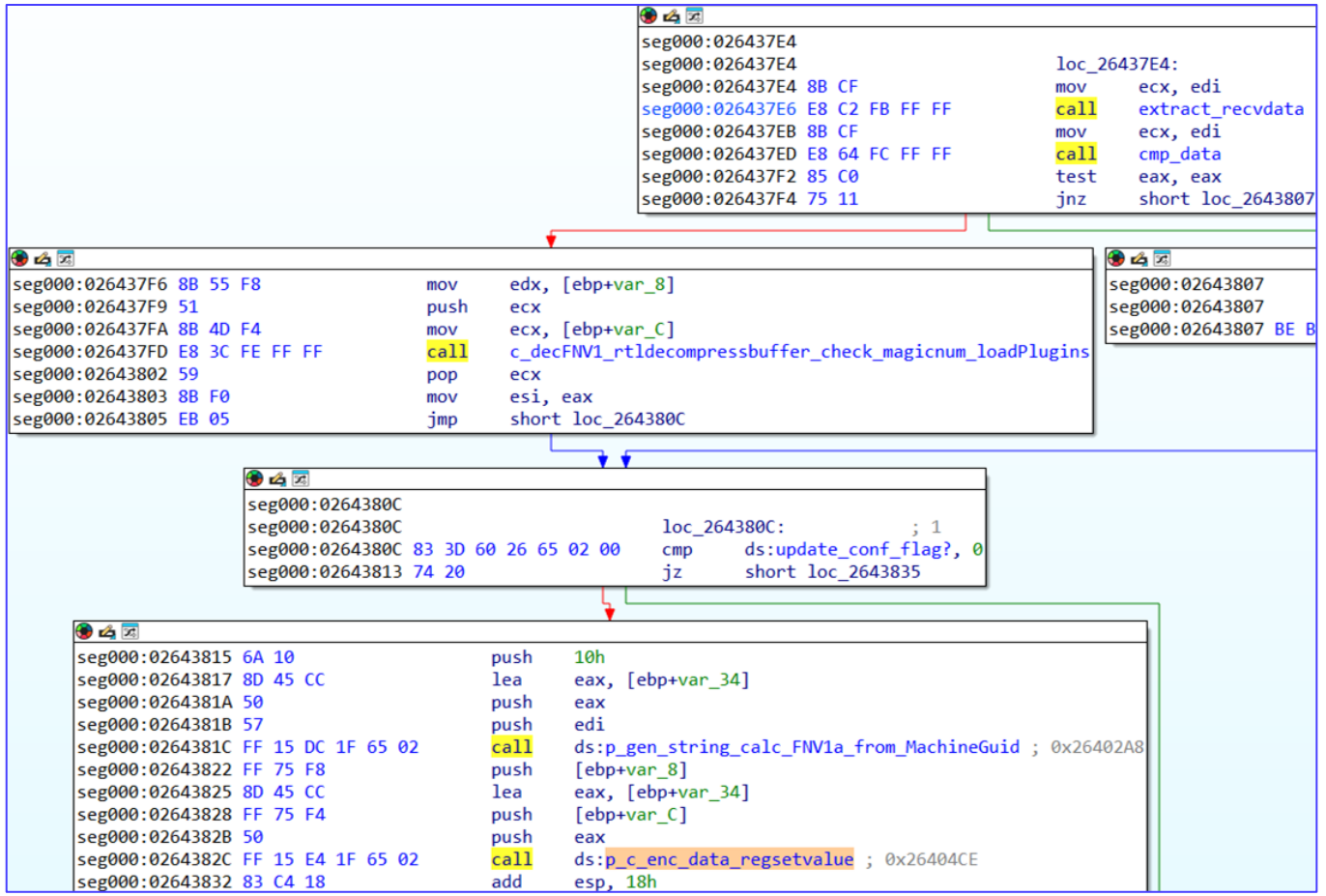
For each loadable module a structure is formed that is added to the list that modules can use to call each other's functions. To w other auxiliary tasks, the `Root` module exports the function table.

Additional modules of Shadowpad have already been reported as above, but additional modules for DeedRAT and BloodAlchemy have not been reported yet

Storage Location of Additional Modules

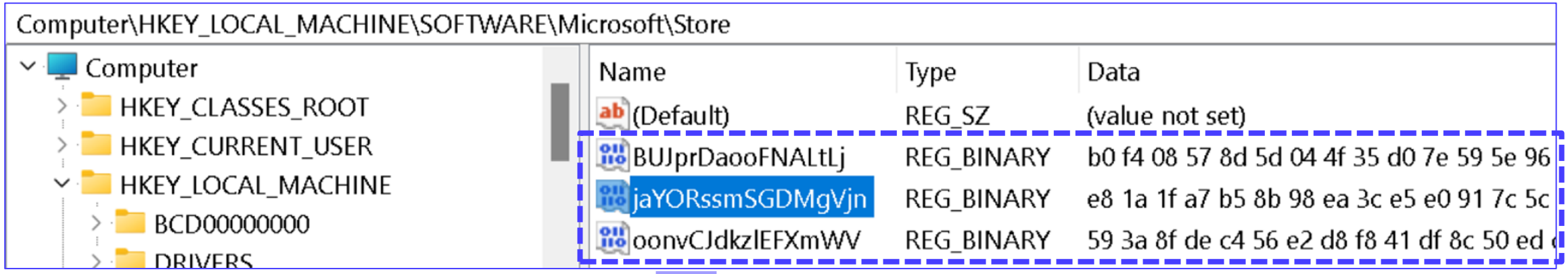
Additional modules from C2 were encrypted and stored into the registry

- BloodAlchemy's baseapi command 0x1302 loads an additional module received from the C2
- Encrypts and compresses the data from C2, then stores the encrypted data into **the registry**

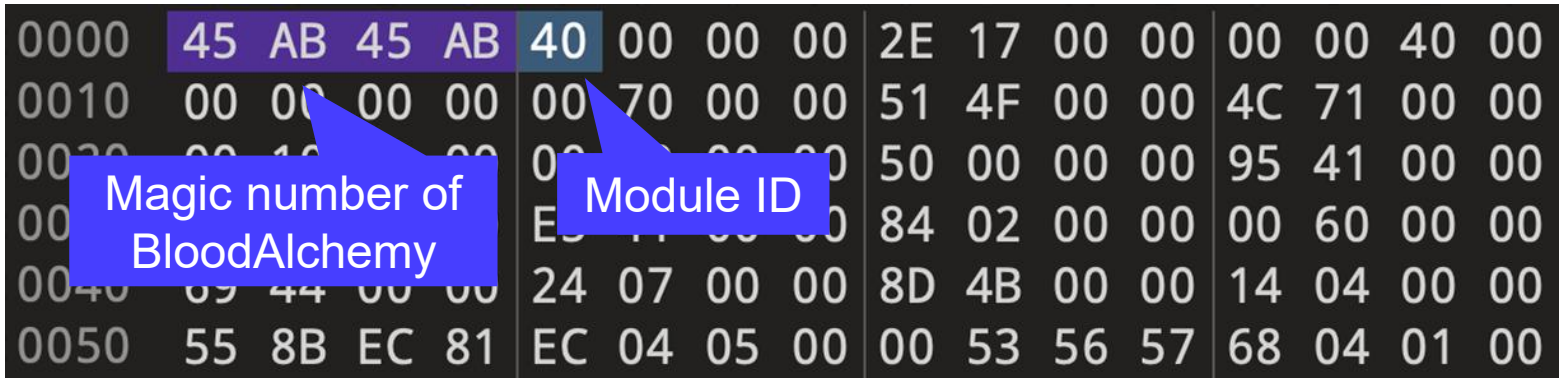


Hunting Additional Modules of BloodAlchemy

Three encrypted modules were discovered from the registry



Decompress & decrypt

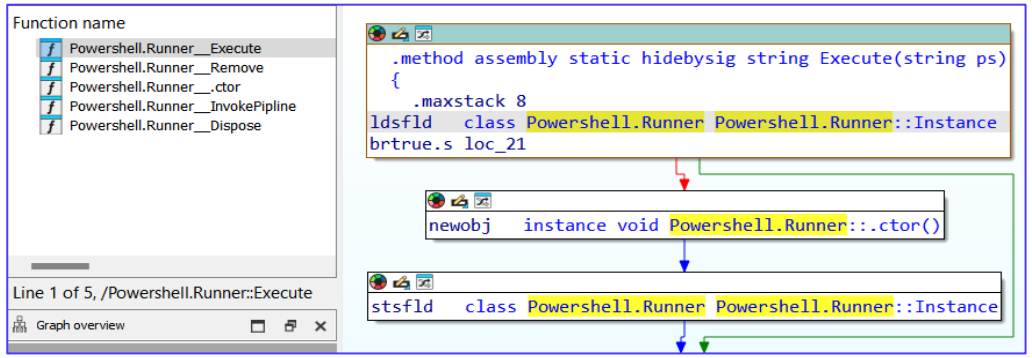


Module ID	Name
0x20	stdapi
0x40	netapi
0x50	impapi

BloodAlchemy's Additonal Module: stdapi (0x20)

stdapi enable the RAT capabilities

- File operations
  - List, move, copy, and execute files/directories
- Shell execution
  - Remote command-line access via anonymous pipes and WMI
- Payload execution
  - Execute received or embedded .NET PEs (e.g., a tiny PowerShell runner)



Command ID	Description
0x2101	Send drive information
0x2102	Send files/directories list
0x2103	Send recursive file/directory list
0x2104	Move or copy files/directories
0x2105	Execute a file
0x2106	Read data from specified offset of a file
0x2108	Receive data from C2 and save it as a file
0x210A	Create directories
0x210B	Resolve environment path
0x210C	Change file creation time
0x210D	Change file access time
0x210E	Change file write time
0x210F	Send list of network resources (e.g. shared folders)
0x2110	Connect to network resource (e.g. shared folders) with auth
0x2201	Start interactive shell via anonymous pipe
0x2301	Start interactive shell via console (WMI)
0x2401	Execute received .NET PE from C2
0x2402	Execute embedded .NET PE (a tiny Powershell runner)

BloodAlchemy's Additonal Module: netapi (0x40)

This module provides network reconnaissance features

- Network scanner
  - Use scanners to map the local network and identify open ports
- Manipulate traffic
  - Act as a bridge/proxy to forward network traffic
- Capture data
  - Sniff network packets and send the captured data to the C2 server

Command ID	Description
0x4101	Start TCP communication to an URL received from C2
0x4103	Start UDP bridge
0x4201	Start TCP communication to IP and port received from C2
0x4202	Connect named pipe and communication
0x4203	Start UDP communication to IP and port received from C2
0x4204	Start MUX communication to IP and port received from C2
0x4401	Send list of adapter ipv4 and netmasks to C2
0x4402	Start ARP sweeping
0x4404	Start TCP port scanner
0x4501	Get IP address which resolved by localhost
0x4502	Start for network capture and save them into a file
0x4503	Stop network capture and delete the file
0x4504	Send network captured data
0x4601	Send list of port transfer session to C2
0x4602	Manage udp/tcp port transfer session
0x4603	Close session

BloodAlchemy's Additonal Module: impapi (0x50)

impapi specializes in the impersonation of user session

- Impersonate user sessions
  - Execute with another user's privileges via CreateProcessAsUserW
- Manage sessions
  - Control processes across all user sessions, both active and disconnected
- Establish persistence
  - Ensure the malware runs for all users, regardless of their session state

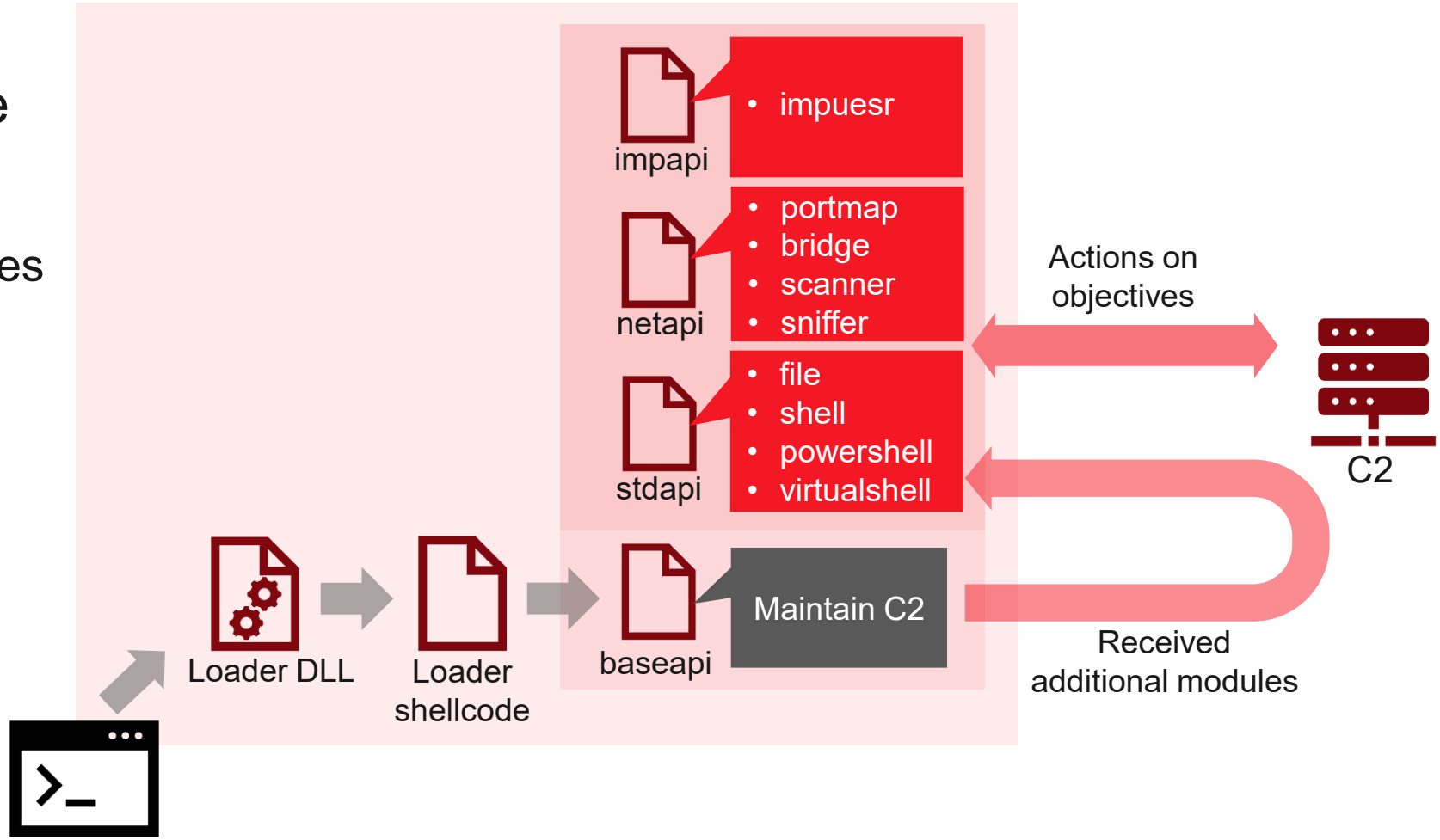
Command ID	Description
0x5101	Send impersonated list of user session information to C2

```
13  active = kernel32_WTSGetActiveConsoleSessionId();
14  if ( active != -1 )
15  {
16    if ( hProcess )
17    {
18      if ( active )
19      {
20        kernel32_TerminateProcess(hProcess, 0);
21        kernel32_CloseHandle(hProcess);
22        j_msvcrt_memset(&hProcess, 0, 284);
23        dword_E221A0 = -1;
24      }
25    }
26    if ( !c_CreateProcessAsUserW(v2) )
27      return 0x643;
28    hProcess = v2[0];
29    dword_E221A0 = active;
30    dword_E2219C = v2[2];
31    kernel32_CloseHandle(v2[1]);
```

BloodAlchemy's Complete Form

C2-delivered modules turn it into a full-featured RAT

- basepi is root module
  - manage and load for each additional modules
  - Hot-swap additional modules based on the purpose and use case



Capabilities and Obstacles

Flexible functionality	Additional modules from C2 enable expanded capabilities
Analysis difficulty	The complex interdependencies between modules make analysis difficult
Evasive countermeasures	Since modules are only deployed during the post-exploitation stage, they are hard to acquire, making investigation challenging

Attack Campaigns Involving DeedRAT/BloodAlchemy

Developer Speculation

- In 2017, Microsoft complaint alleges that Barium (subgroup of BRONZE ATLAS (APT41)) deployed ShadowPad [*1]
  - At the time, ShadowPad was used only by Barium
  - They also have special version of ShadowPad
- The DOJ indictments allege that Chengdu 404 (attributed to BRONZE ATLAS) deployed ShadowPad [*2]
- Tan Dailin (Wicked Rose, member of Barium) and whg (author of PlugX) played a major role in developing important tools leading up to the ShadowPad [*3]

https://www.fbi.gov/wanted/cyber/apt-41-group

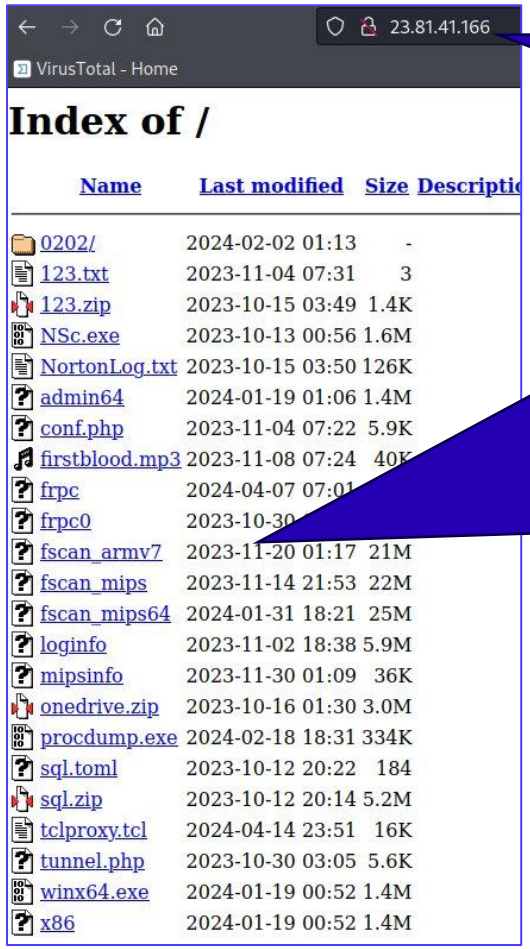
Barium or associates may have developed DeedRAT/Blood Alchemy from ShadowPad

2007 NCPH Remote Control Software → 2008 PlugX → 2015 ShadowPad → 2019 DeedRAT → 2023 BloodAlchemy

*1 https://www.noticeofpleadings.net/barium/files/COMPLAINT_AND_SUMMONS/Complaint.pdf
*2 https://www.fbi.gov/wanted/cyber/apt-41-group
*3 https://assets.sentinelone.com/c/Shadowpad?x=P42eqA

New DeedRAT Intrusion Set Exposed in Apr 2024

https://x.com/banthisguy9349/status/1788981985810972904



23.81.41[.]166
Leaseweb Japan K.K

Hosted China related malware/tools:

- 2 DeedRAT sets
- ProxyTool
  - Stowaway
  - FRP
- ScanTool
  - Fscan
- Webshell (HTTP Tunneling)
  - Neo-Regeorg

FRP Server Address Linked to ShadowPad C2

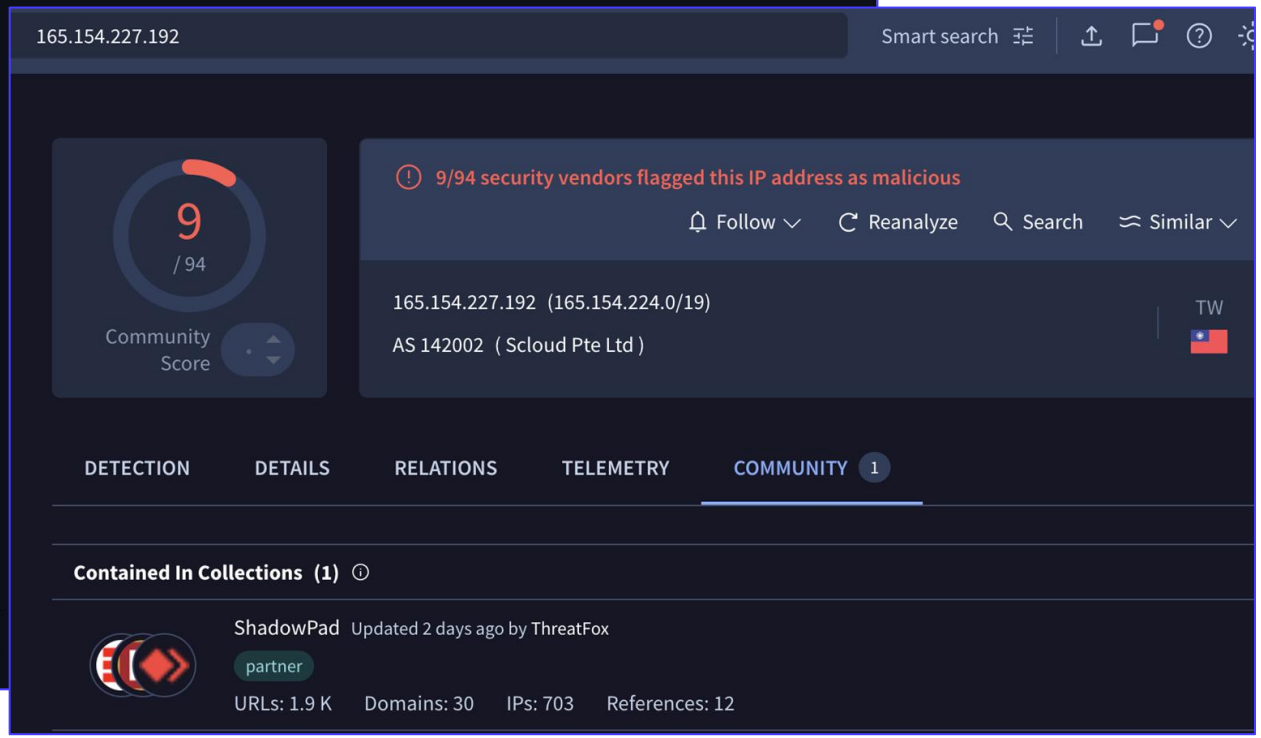
The C2 was attributed to Barium (BRONZE ATLAS, APT41) by 3rd party

```
serverAddr = "165.154.227.192"
serverPort = 7000


[[proxies]]
name = "plugin_socks5"
type = "tcp"
remotePort = 6005
[proxies.plugin]
type = "socks5"
username = "abc"
password = "abc"
```

FRP configuration file

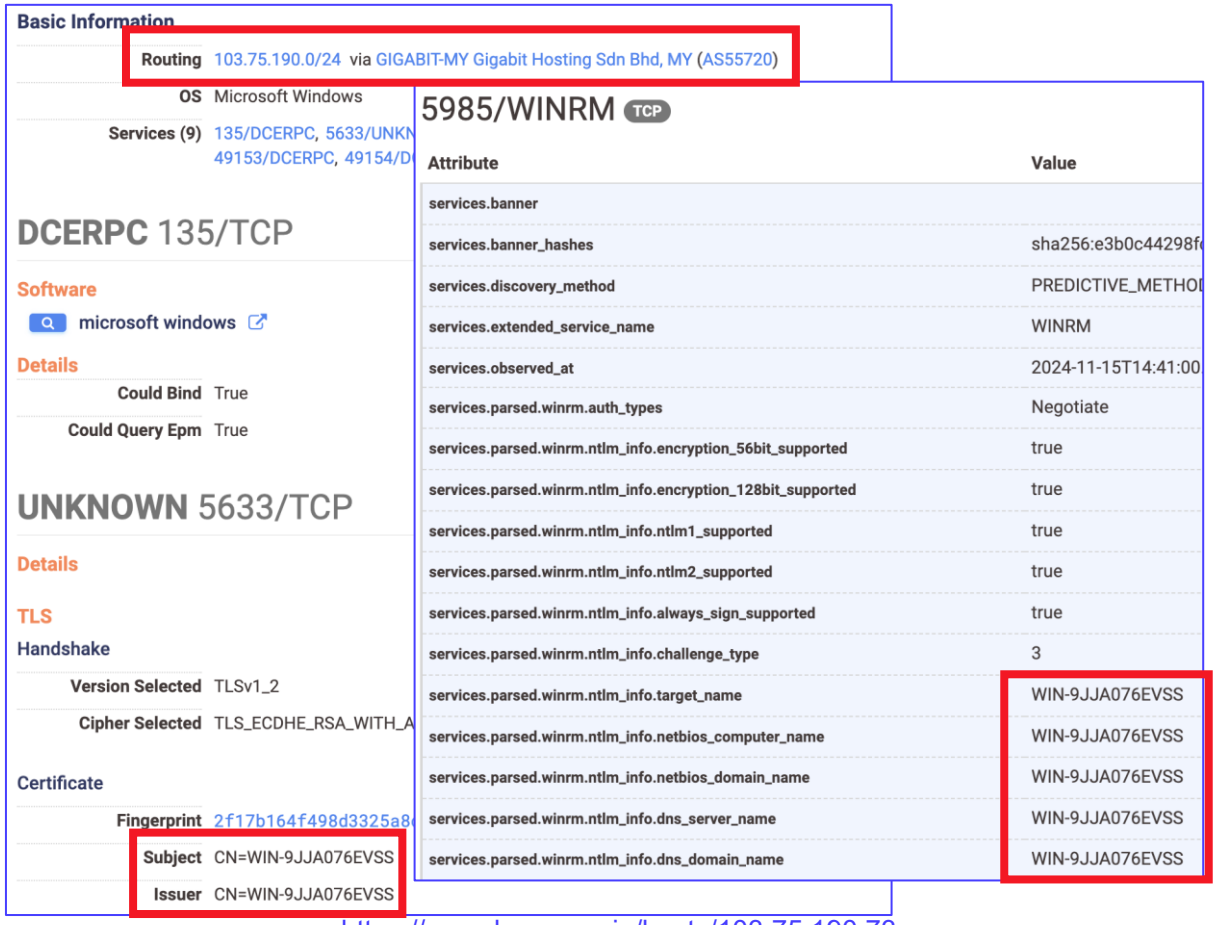
https://github.com/stamparm/maltrail/blob/master/trails/static/malware/apt_barium.txt

```
# Reference: https://threatfox.abuse.ch/browse/malware/win.shadowpad/ (# 2023-10-26)

http://103.158.190.167
http://103.255.118.149
http://103.255.118.150
http://103.51.110.5
http://104.194.129.178
http://104.233.167.99
http://118.193.56.234
http://124.126.116.7
http://139.180.193.182
http://149.202.45.103
http://149.28.157.235
http://149.88.75.49
http://156.236.114.202
http://158.247.202.188
http://158.247.203.58
http://158.247.213.14
http://165.154.227.192
http://167.179.108.149
```

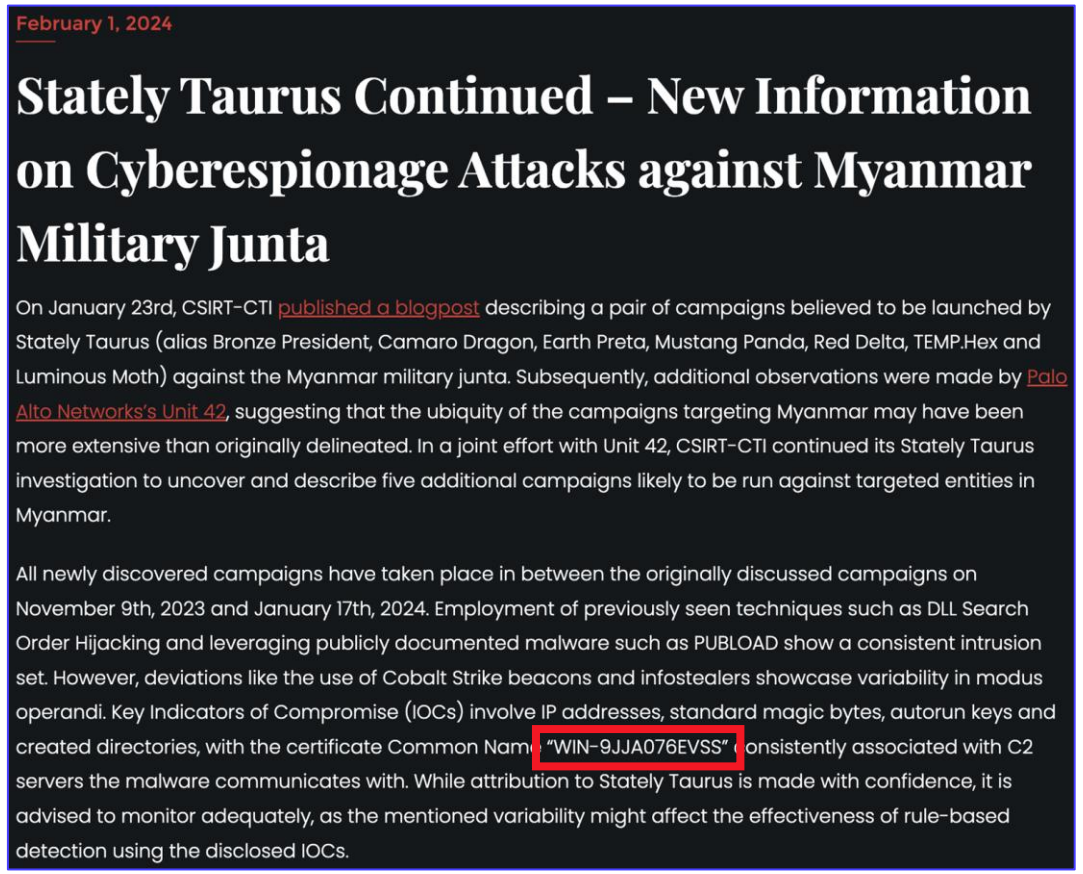


https://www.virustotal.com/gui/ip-address/165.154.227.192/community

Hostname Listed on The DeedRAT C2 Certificate Linked to BRONZE PRESIDENT (Earth Preta)

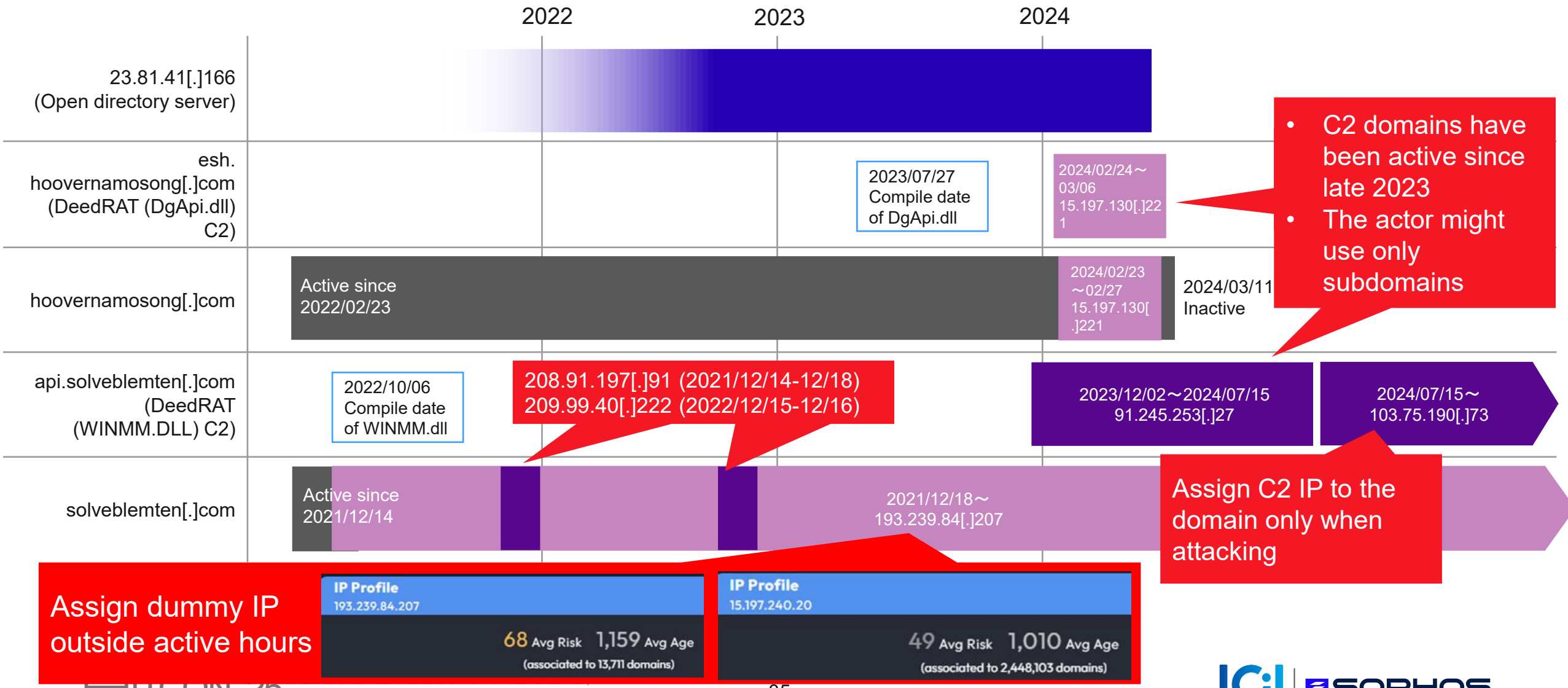


https://search.censys.io/hosts/103.75.190.73

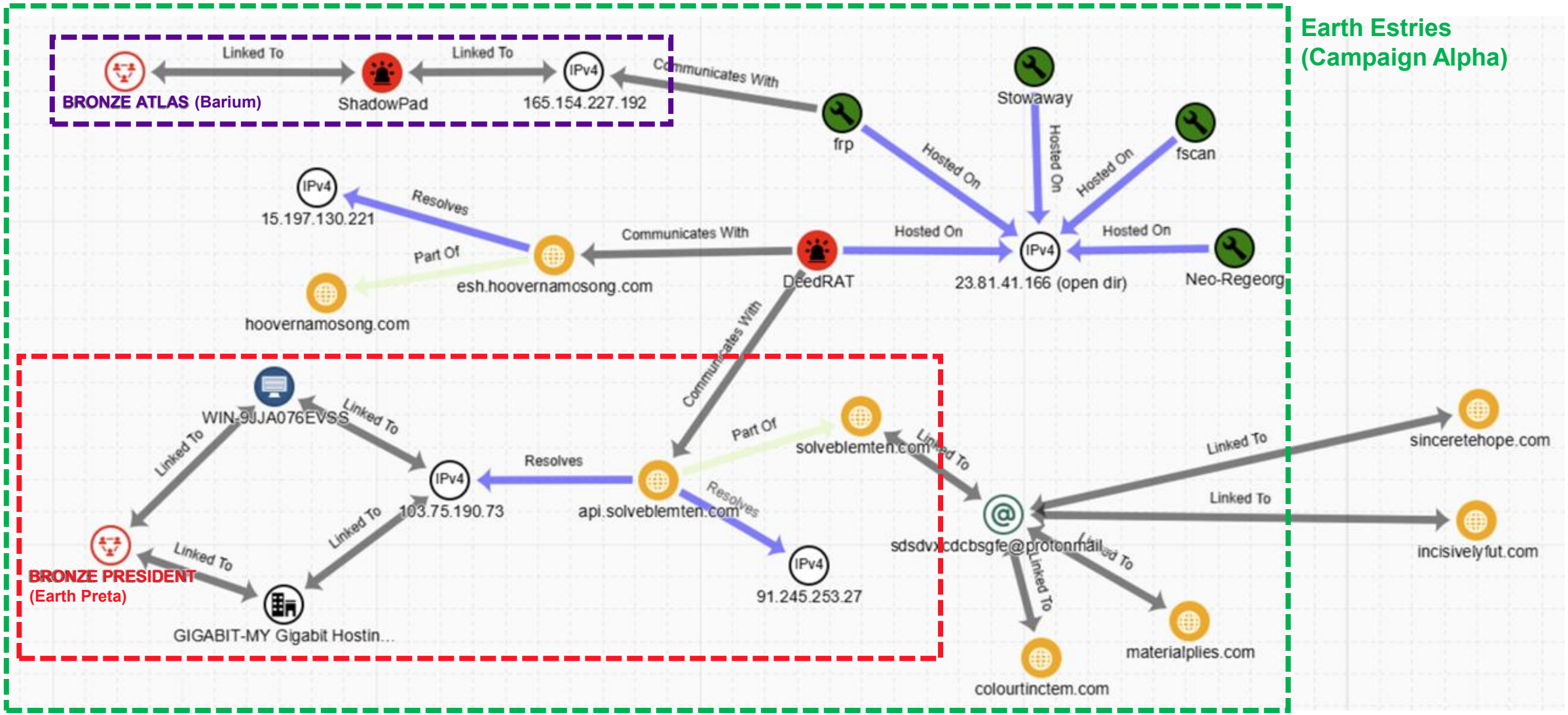


https://csirt-cti.net/2024/02/01/stately-taurus-continued-new-information-on-cyberespionage-attacks-against-myanmar-military-junta/

Timeline Analysis of Network Indicators



Summary – Linking to 3 Chinese-Speaking Groups



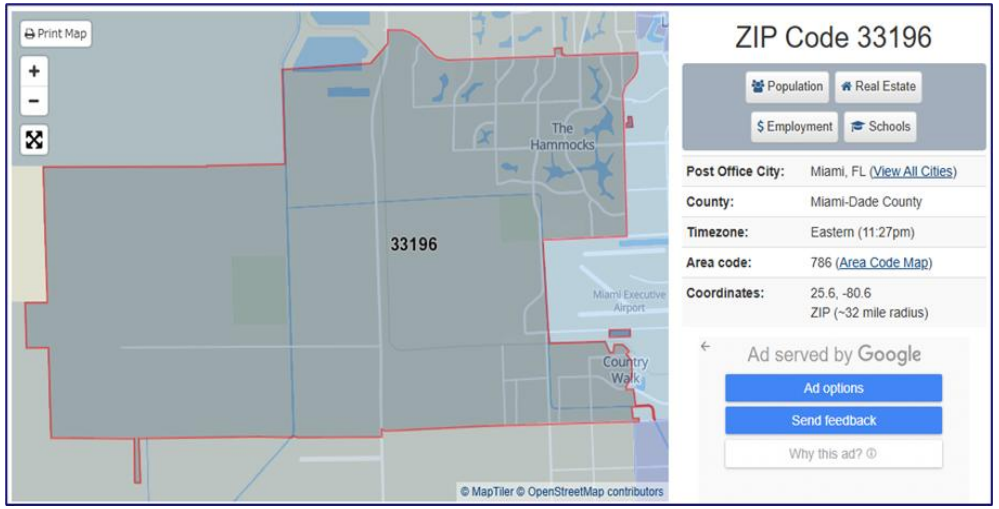
Whois Info about DeedRAT C2

Non-existent address, with mismatched phone area code

- Registrant Name: Tommie Arnold
- Registrant Organization:
- Registrant Street: 1729 Marigold Lane
- Registrant City: Miami
- Registrant State/Province: FL
- Registrant Postal Code: 33196
- Registrant Country: US
- Registrant Phone: +1.9852975116
- Registrant Phone Ext:
- Registrant Fax:
- Registrant Fax Ext:
- Registrant Email:sdsdvxcdcbsgfe@protonmail.com

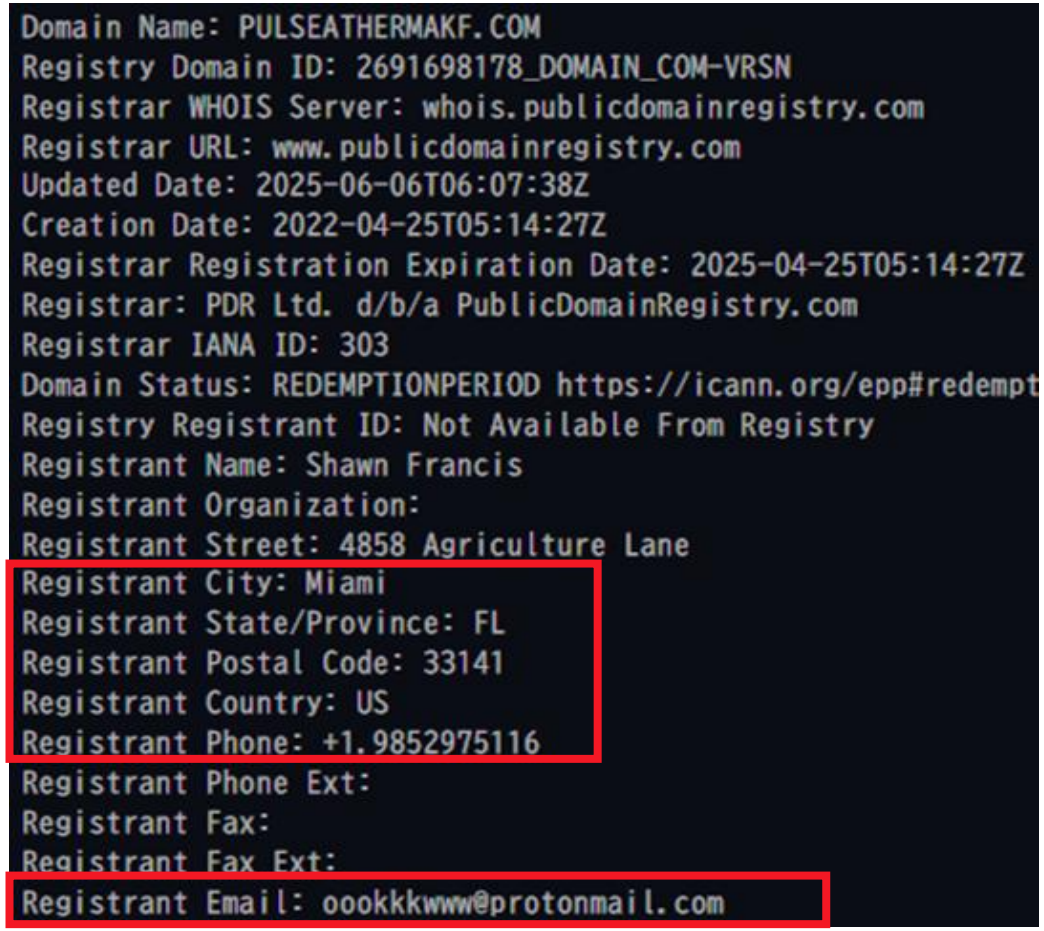
No presence in Miami

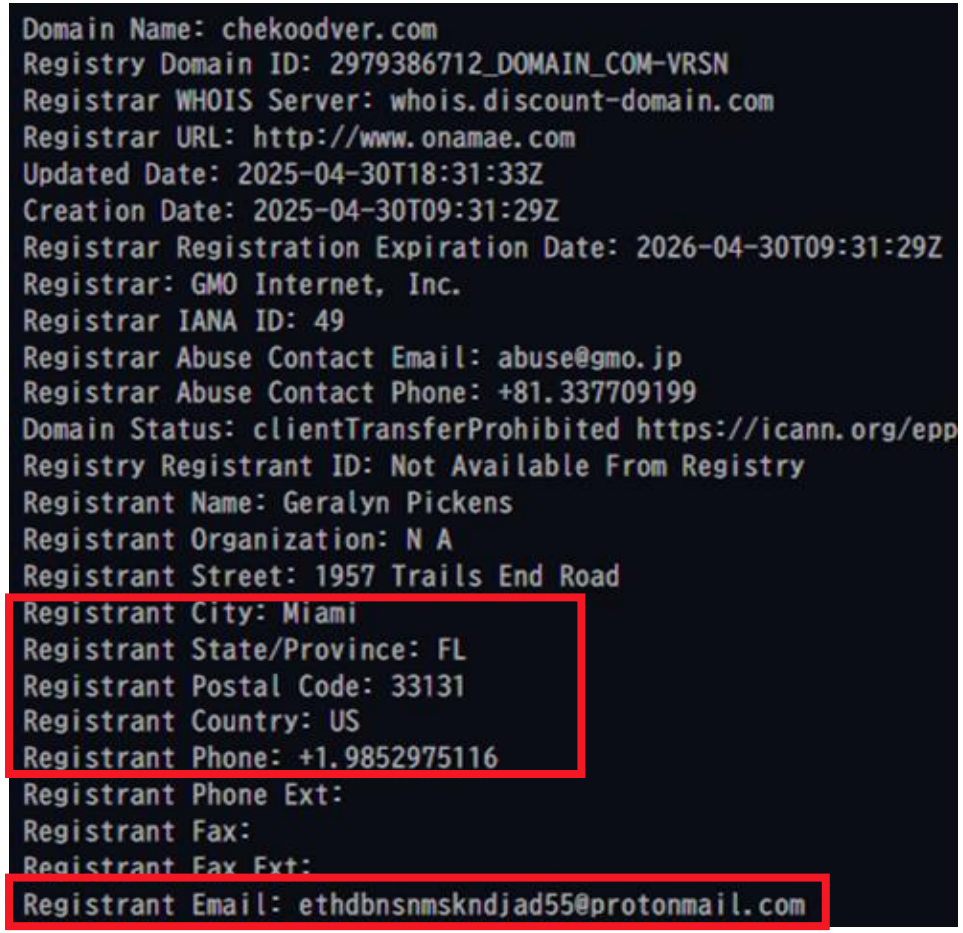
985: Louisiana area code



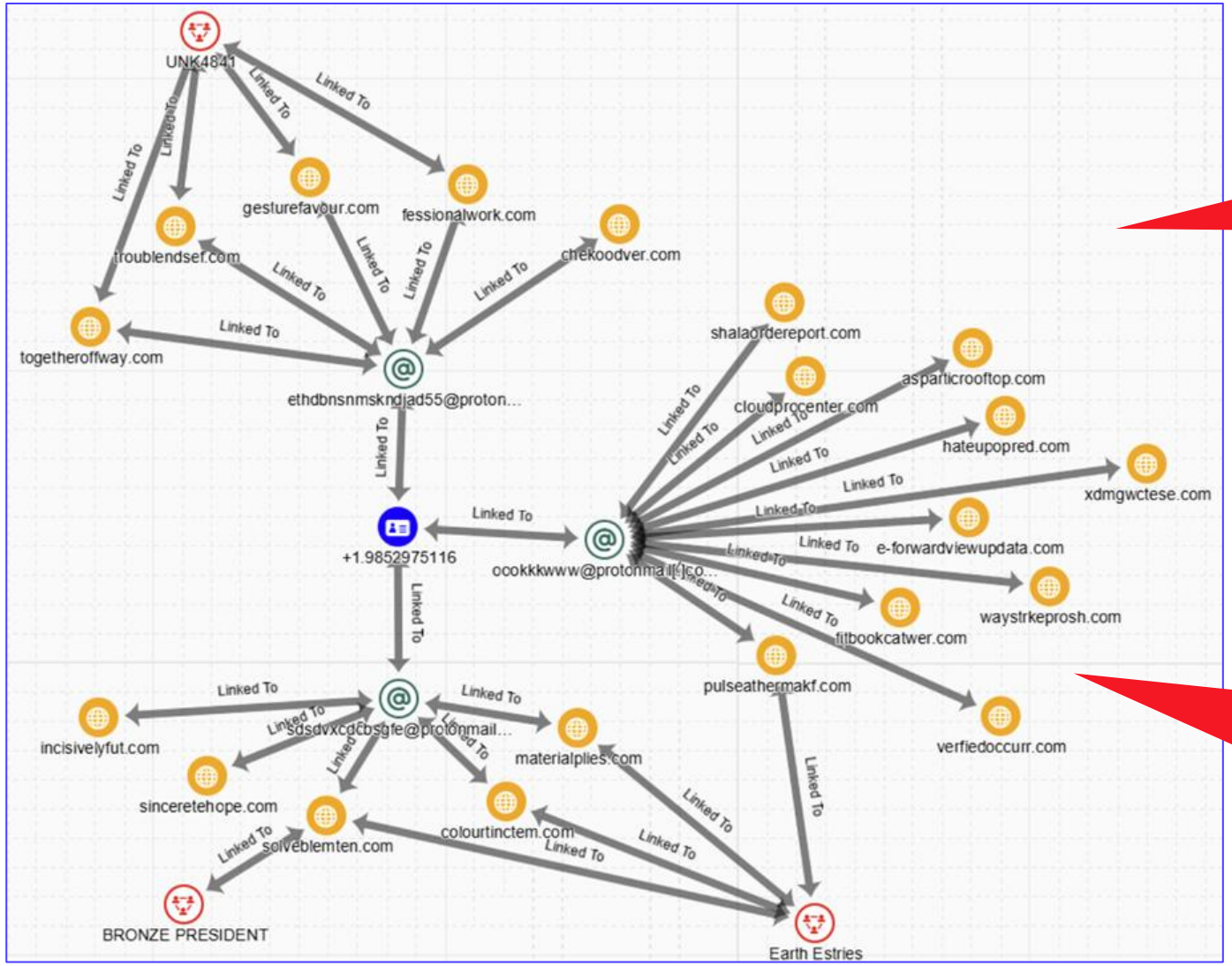
Suspicious Domains Linked to +1-985-297-5116

Identified a cluster of suspicious domains with similar Whois info





Linking Known IoCs and Domains



Several domains are linked to total three Chinese-Speaking groups

Other domains:

- May have been used in unconfirmed past attacks
- Will be used in future attacks

Domains Related sdsdvxcdcbsgfe@protonmail.com

Domain	Registrar info	DNS records	Known links
solveblemten[.]com api.solveblemten[.]com	2021/12/14 Domain Activated 2021/12/14 GMO INTERNET, INC. 2022/12/17 PDR Ltd. d/b/a PublicDomainRegistry.com	208.91.197.91 (2021/12/14-12/18) 209.99.40.222 (2022/12/15-12/16) 91.245.253.27 (2023/12/2-2024/6/29) 103.75.190.73 (2024/7/20-11/27) 193.239.84.207 and 15.197.240.20 and 35.186.223.180	BRONZE PRESIDENT Earth Estries
materialplies[.]com pop3.materialplies[.]com	2021/12/15 Domain Activated 2021/12/15 GMO INTERNET, INC. 2022/12/17 PDR Ltd. d/b/a PublicDomainRegistry.com	209.99.40.222 (2022/12/15-12/16) 208.91.112.55 (2023/2/27-2/28) 103.159.133.251 (2023/9/1-) 193.239.84.207 and 15.197.240.20	Earth Estries
colourtinctem[.]com news.colourtinctem[.]com service.colourtinctem[.]com	2021/12/14 Domain Activated 2021/12/14 GMO INTERNET, INC. 2022/12/17 PDR Ltd. d/b/a PublicDomainRegistry.com	208.91.197.91 (2021/12/14-12/18) 209.99.40.222 (2022/12/15-12/16) 103.91.64.214 (2024/8/19-11/28) 123.253.34.49 (2024/3/25-11/25) 193.239.84.207 and 15.197.240.20	Earth Estries
sinceretehope[.]com api.sinceretehope[.]com web.sinceretehope[.]com	2021/12/14 Domain Activated 2021/12/14 GMO INTERNET, INC. 2022/12/16 PDR Ltd. d/b/a PublicDomainRegistry.com	209.99.40.222 (2022/12/15-12/16) 172.107.240.4 (2024/7/16-) 202.59.10.157 (2025/3/25-) 193.239.84.207 and 15.197.240.20	No
incisivelyfut[.]com pop3.incisivelyfut[.]com	2021/12/14 Domain Activated 2021/12/14 GMO INTERNET, INC. 2022/12/17 PDR Ltd. d/b/a PublicDomainRegistry.com	208.91.197.91 (2021/12/14-12/18) 209.99.40.222 (2022/12/14-12/15) 103.75.189.170 (2024/7/6- 2024/7/7) 193.56.255.210 (2025/4/9-) 193.239.84.207 and 15.197.240.20	No

- Some domains registered a C2 IP only for a short period
- They also used subdomains

Infrastructure Overlaps Revealing Attack Links

Attacks using DeedRAT and BloodAlchemy shared the same infra with other attacks

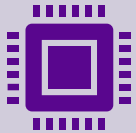
Date	Possible Actor	Malware	Infrastructures	Targets
2022/10～	UNK4841	CVE-2023-2868	+1.9852975116, Miami, FL, US	Around the world
2023/10	BRONZE ATLAS	ShadowPad	165.154.227[.]192	Unknown
2023/10～	**BRONZE ATLAS / BRONZE PRESIDENT / Earth Estries ?**	**DeedRAT**	**165.154.227[.]192** **WIN-9JJA076EVSS AS55720** +1.9852975116, Miami, FL, US	**Unknown**
2024/01	BRONZE PRESIDENT / Earth Preta	PUBLOAD	WIN-9JJA076EVSS AS55720	South Asia
2023/XX	Earth Estries	Hemigate Zingdoor	3280132818 @ qq[.]com	Asia, South Africa, Germany, US
2023/05	**Earth Estries ?**	**BloodAlchemy Cobalt Strike**	**3280132818 @ qq[.]com**	**Asia**
2023/XX	Earth Estries	Zingdoor DeedRAT Cobalt Strike	38.54.71[.]140	Asia, South Africa, Germany, US
2024/XX	Earth Estries	DEMODEX	+1.9852975116, Miami, FL, US	Southeast Asia

Attribution Summary

Multiple Chinese-speaking groups linked to DeedRAT/BloodAlchemy

At least four groups have been identified

Suggests the existence of organizations like i-Soon that share these malware



Subdomains assigned to C2 IPs only during attacks

To avoid detection and response as much as possible

OPSEC failures observed, with shared infrastructure

Hostnames, email addresses, phone numbers

Highlights the importance of sharing threat intelligence to promote public attribution

Conclusions

The Evolving Bloodline

The APT activities involving DeedRAT or BloodAlchemy

- Targeted regions include Russia and Asia, including Japan
- Detection and intelligence collection are both difficult due to limited reporting and samples

Unraveling DeedRAT and BloodAlchemy

- Both share many similarities despite differences like encoding methods
  - Likely evolution: ShadowPad → DeedRAT → BloodAlchemy
- BloodAlchemy's additional modules show its full capabilities
  - DeedRAT's additional modules are still missing

Multiple groups are sharing malware and infrastructure

- Possible entity behind preparing and distributing these resources
- Collaborative intelligence sharing is vital for uncovering these threats

//Secureworks/Confidential - Limited External Distribution

IOCs – DeedRAT / BloodAlchemy

Path	SHA256	Description
DIFX	fe29c70611cdbb86457cf06087f33c1e1ce83c448f4cbbbbe180fd61f8a545d3	Encrypted BloodAlchemy payload (cdn1ac7bdd3.jptomorrow[.]com)
N/A	25268bc07b64d0d1df441eb6f4b40dc44a6af568be0657533088d3bfd2a05455	BloodAlchemy payload
BrLogAPI.dll	e14ee3e2ce0010110c409f119d56f6151fdca64e20d902412db46406ed89009a	BloodAlchemy loader
BrDifxapi.exe	1b26db32651074ec40606d94811353b358eec4127efad8a5576e2e19134a0a6d	Legitimate EXE file
debug.log	67f7faf0161fdac7ebb619a2aa0c73a4a08def05d7752dfdd698d24410d9989e	Encrypted DeedRAT payload (198.13.56[.]197)
dbghelp.dll	b6860214fcc1ef17937e82b1333672afa5fcf1c1b394a0c7c0447357477fe7c9	DeedRAT loader
browser_diagnostics.exe	a1885a930589c20aeede98d730222bb0b0b6574234fd2072a7709e4fc90ecbaf	Legitimate EXE file
SBAMRES.DLL.CC	2d9107edad9f674f6ca1707d56619a355227a661163f18b5794326d4f81a2803	Encrypted DeedRAT payload (luckybear669.kozow[.]com)
SBAMRES.DLL	99a0b424bb3a6bbf60e972fd82c514fd971a948f9cedf3b9dc6b033117ecb106	DeedRAT loader
MicRun.exe	e356dbd3bd62c19fa3ff8943fc73a4fab01a6446f989318b7da4abf48d565af2	Legitimate EXE file

IOCs – DeedRAT Discovered in Open Directory

Path	SHA256	Description
0202¥dbindex.dat	1a38303fb392ccc5a88d236b4f97ed404a89c1617f34b96ed826e7bb7257e296	Encrypted DeedRAT payload (esh.hoovernamosong[.]com:443)
0202¥DgApi.dll	b2b617e62353a672626c13cc7ad81b27f23f91282aad7a3a0db471d84852a9ac	DeedRAT loader
0202¥imfsbDll.dll	05840de7fa648c41c60844c4e5d53dbb3bc2a5250dcb158a95b77bc0f68fa870	Legitimate DLL file of IObit malware fighter
0202¥imfsbSvc.exe	5b88f7d36fe435cd6944bda05f1758f64c7d5136a5f529a58522ac3b0dc9743a	Legitimate EXE file of IObit malware fighter
123.zip	b70ebdcfb01e5aa211c504eb15757794cc5bb17d7607512a22672c7782792c45	Include WINMM.dll
NortonLog.txt	fba149eb5ef063bc6a2b15bd67132ea798919ed36c5acda46ee9b1118b823098	Encrypted DeedRAT payload (api.solveblemten[.]com:8080)
WINMM.dll	fc3be6917fd37a083646ed4b97ebd2d45734a1e154e69c9c33ab00b0589a09e5	DeedRAT loader
NSc.exe	d74dccb0a4856e8059eafe5954f90e2ab48fe2fc5e3f82790c941dc58b8c6252	Legitimate EXE file of Symantec Norton

IOCs – Tools Discovered in Open Directory

Path	SHA256	Description
fscan_armv7	9f9675403c2be03232b1c3abe344bf0f4188454955ba89592be52ef77add4a39	fscan, https://github.com/shadow1ng/fscan
fscan_mips	aa12c40bc0ef87b1b706f1e9062d72d8c67c3b4b3347741efb38cf71817777d2	fscan, https://github.com/shadow1ng/fscan
fscan_mips64	ebaa36db295f1c3a7d59e460ce6813221d0097f3c12ce26e818d4d4ac83c0919	fscan, https://github.com/shadow1ng/fscan
loginfo	f34bd1d485de437fe18360d1e850c3fd64415e49d691e610711d8d232071a0b1	fscan, https://github.com/shadow1ng/fscan/
mipsinfo.txt	34b5a500afd6e3d383375d78643c0d19dfd8a6130ae31e2e1598531784d1fbb3	Logfile of downloading fscan_mips from the following URL on 2023-11-30 01:09:29 https://github.com/shadow1ng/fscan/releases/download/1.8.3/fscan_mips
frpc	bbadd7a8d1f24d39f4fcc65b6fd5cbe22904c049ae3a6fee8b510ce6352d59ef	FRP reverse proxy, https://github.com/fatedier/frp
frpc0	7514ae1ed274d7ea5cd6c3e76f6845f5317dc6f61fc441ac8112c8219ca26132	FRP reverse proxy, https://github.com/fatedier/frp
sql.exe	400278da13d6dee4a8d79151e643126bb9cdea61dae0ed8d45bf701db460b489	FRP reverse proxy, https://github.com/fatedier/frp
sql.toml.txt	ce638cf14d91a3ece6d5c593ecbae0c94f2669d5a8684fa9df0b506cf1bf4019	FRP configuration file
x86	45d375871a1eb311af5d7b7179ca5cc060bce353292d1912fb766a5025f3d6aa	Stowaway proxy tool, https://github.com/ph4ntonn/Stowaway/tree/master
winx64.exe	a78d737f30e03d166d4e3e3b2dca71d54f1cbf582206dfe16a1e717ce3dc0ef7	Stowaway proxy tool, https://github.com/ph4ntonn/Stowaway/tree/master
admin64	e4864472f1813776e19fd3468495cfce9745fe9c0270b85c6df4365028ce9fb6	Stowaway proxy tool, https://github.com/ph4ntonn/Stowaway/tree/master
sql.exe	400278da13d6dee4a8d79151e643126bb9cdea61dae0ed8d45bf701db460b489	FRP reverse proxy, https://github.com/fatedier/frp
conf.php.txt	f8db15375c14be16654323b422bafc992a97914740eb64e251c494447f52d075	Neo-Regeorg, https://github.com/L-codes/Neo-reGeorg
tunnel.php.txt	fc69784283251fe385411a546e9ba3ccdb771b36283ff123ff87856689c2b87c	Neo-Regeorg, https://github.com/L-codes/Neo-reGeorg
onedrived.ps1	2fd4a49338d79f4caee4a60024bcd5ecb5008f1d5219263655ef49c54d9acdec	Powershell dropper

IOCs – IP&Domain, Hostname, Email Address

IOCs	Description
38.54.79[.]177	VPN connection from TA (BloodAlchemy 2024/05)
107.179.130[.]144	VPN connection from TA (BloodAlchemy 2024/05)
121.41.35[.]65	Cobalt Strike C2 (BloodAlchemy 2024/05)
cdn39a700bb.jptomorrow[.]com	Cobalt Strike C2, Linked to Earth Estries's used domain (BloodAlchemy 2024/05)
cdn1ac7bdd3.jptomorrow[.]com	BloodAlchemy C2, Linked to Earth Estries's used domain (BloodAlchemy 2024/05)
cdn-hk-6dc8.bogotatrade[.]co	BloodAlchemy C2 (BloodAlchemy 2024/05)
198.13.56[.]197	DeedRAT C2 (debug.log)
esh.hoovernamosong[.]com	DeedRAT C2 (dbindex.dat)
15.197.130[.]221	Associated to esh.hoovernamosong[.]com
api.solveblemten[.]com	DeedRAT C2 (NortonLog.txt)
91.245.253[.]27	Associated to api.solveblemten[.]com
103.75.190[.]73	Associated to api.solveblemten[.]com
23.81.41[.]166	Opendir
165.154.227[.]192	frp server IP, Linked to Barium's ShadowPad C2
WIN-9JJA076EVSS	TLS certificate CN of 103.75.190[.]73, Linked to BRONZE PRESIDENT's used certificate CN
3280132818@qq[.]com	Registrar email of jptomorrow[.]com, Linked to Earth Estries's used domain

IOCs – IP&Domain, Hostname, Email Address

IOCs	Description
troublendsef[.]com	Domain linked to ethdbnsnmskndjad55@protonmail[.]com, +1.9852975116, Miami, FL, US
gesturefavour[.]com	
chekoodver[.]com	
fessionalwork[.]com	
togetheroffway[.]com	
pulseathermakf[.]com	Domain linked to oookkkwww@protonmail[.]com, +1.9852975116, Miami, FL, US
cloudprocenter[.]com	
fitbookcatwer[.]com	
shalaordereport[.]com	
asparticrooftop[.]com	
e-forwardviewupdata[.]com	
waystrkeprosh[.]com	
verfiedoccurr[.]com	
hateupopred[.]com	
xdmgwctese[.]com	

IOCs – IP&Domain, Hostname, Email Address

IOCs	Description
solveblemten[.]com	Domain linked to sdsdvxcdcbsgfe@protonmail[.]com, +1.9852975116, Miami, FL, US
materialplies[.]com	
colourtinctem[.]com	
sinceretehope[.]com	
incisivelyfut[.]com	