Introducing CS2BR

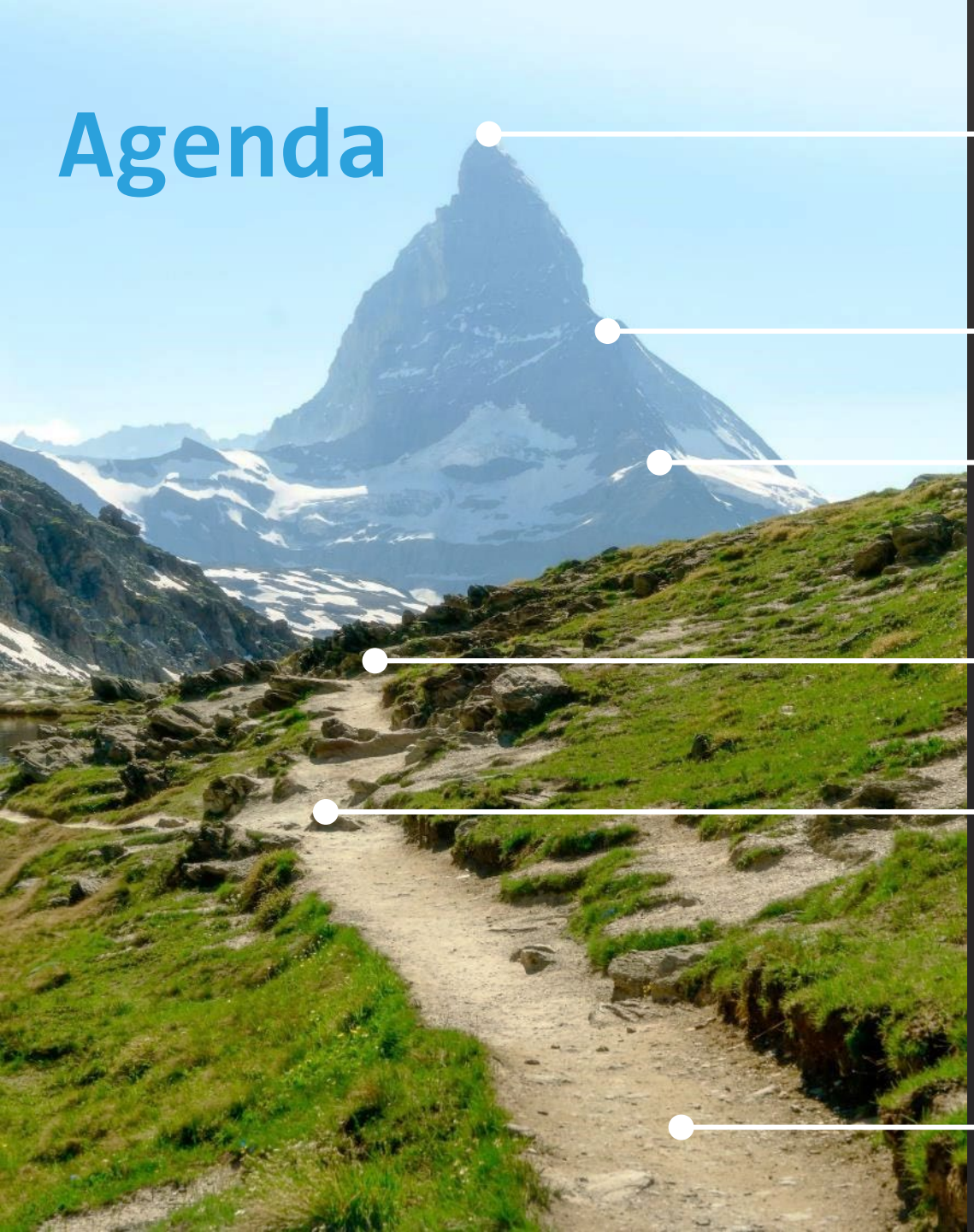
Teaching Badgers new Tricks

Moritz Thomas // Patrick Eisenschmidt

CobaltStrike 2 BruteRatel

bof.c

bof.c



Agenda

- CS2BR binpatch
- Introducing CS2BR + Demo
- BOFs & BOF APIs
- Brief RT History Lesson
- Why?
- About Us

$ whoami

Moritz Thomas

@moritzlthomas

NVISO ARES Red Team

NVISO ARES R&D

Patrick Eisenschmidt

@secdu_de

NVISO ARES Red Team Lead

NVISO ARES Exposure Lead

ARES

ADVERSARIAL
RISK EMULATION
& SIMULATION

nviso

Projects

Pentesting

Red Teaming

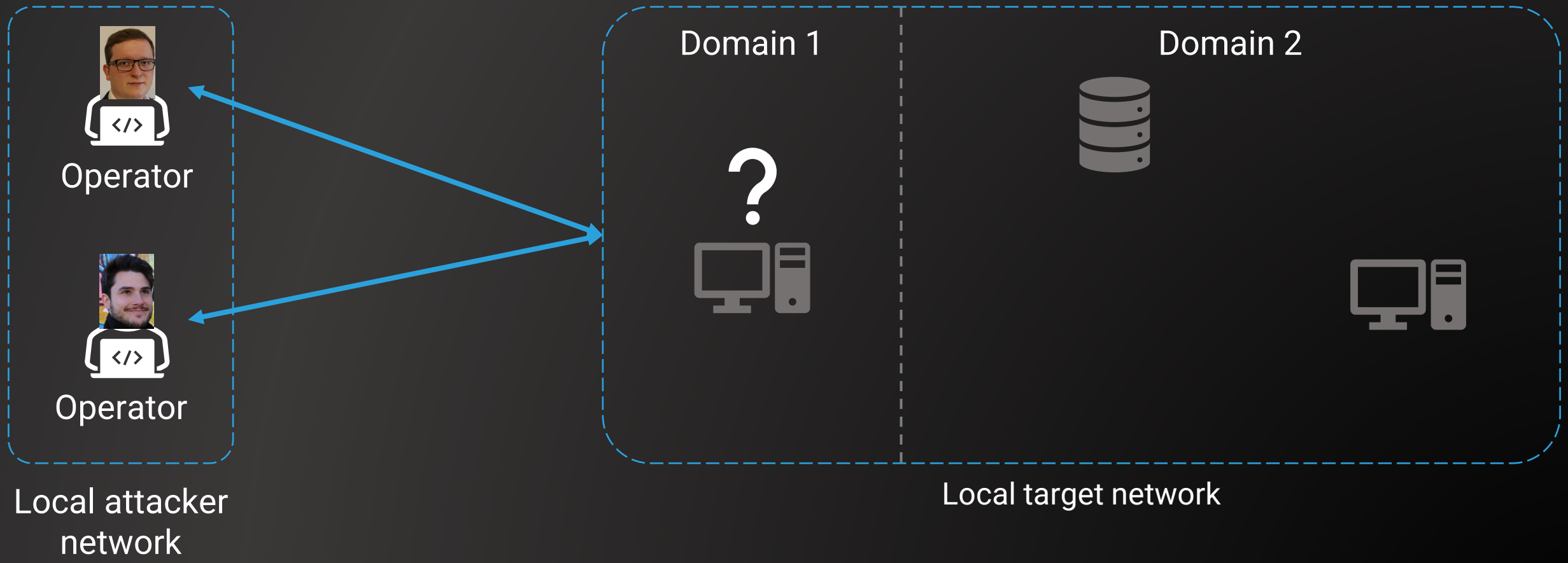
TIBER-EU

Challenges

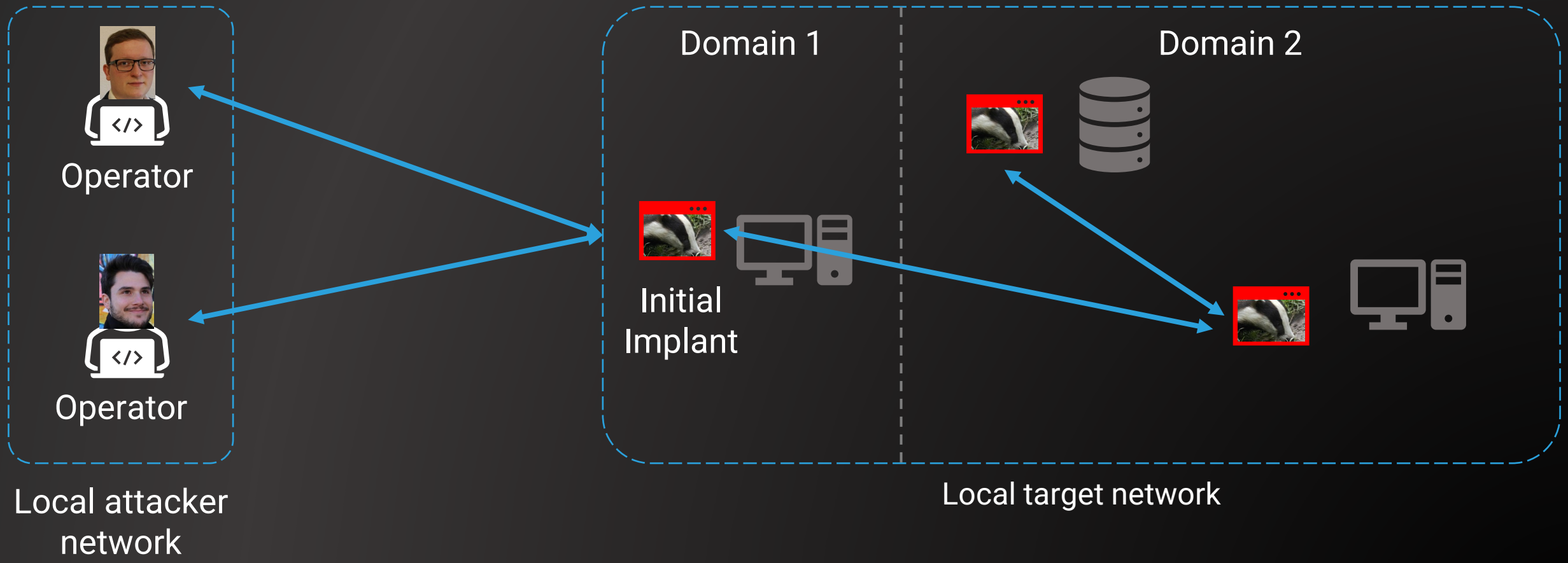
- Lots of assets
- Highly sensitive data/infra
- High maturity
  - Training (people)
  - EDRs/AVs (detection)
  - Infrastructure (machines)
  - Policies and Processes

Threat Intelligence Based
Ethical Red Teaming

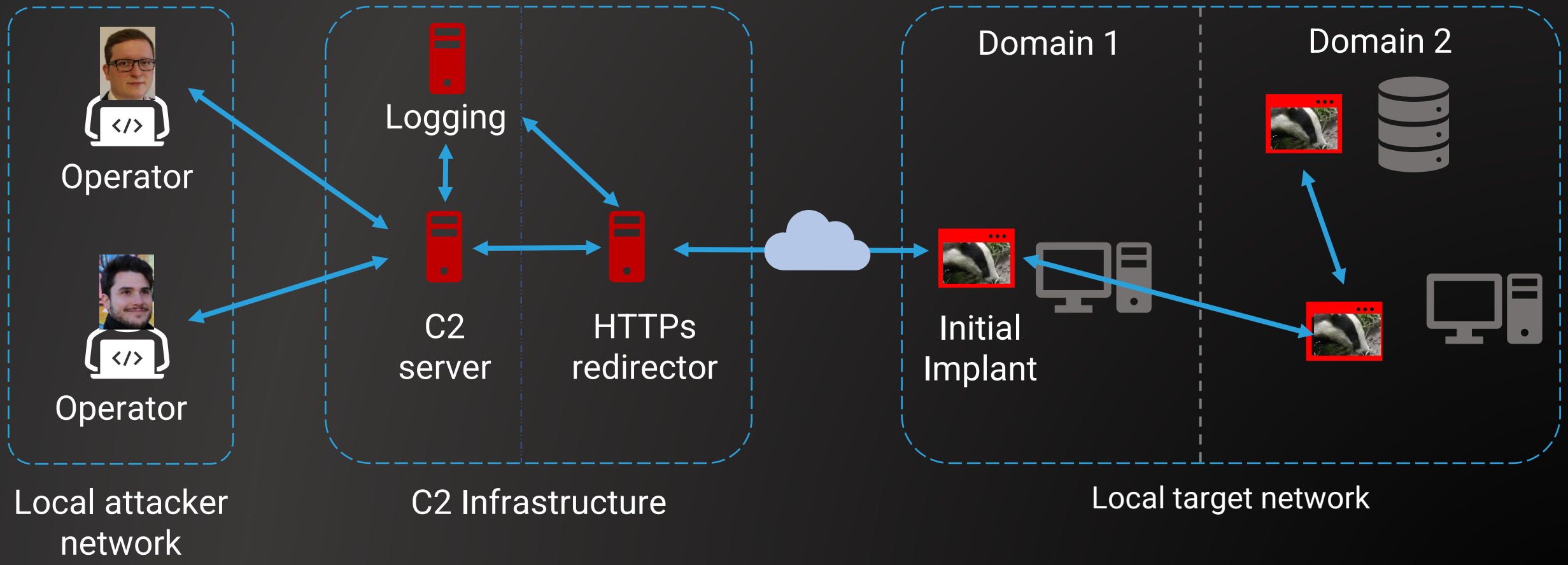
Offensive Infrastructure (C2)



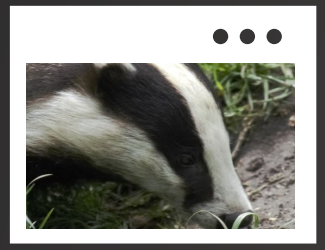
Offensive Infrastructure (C2)



Offensive Infrastructure (C2)



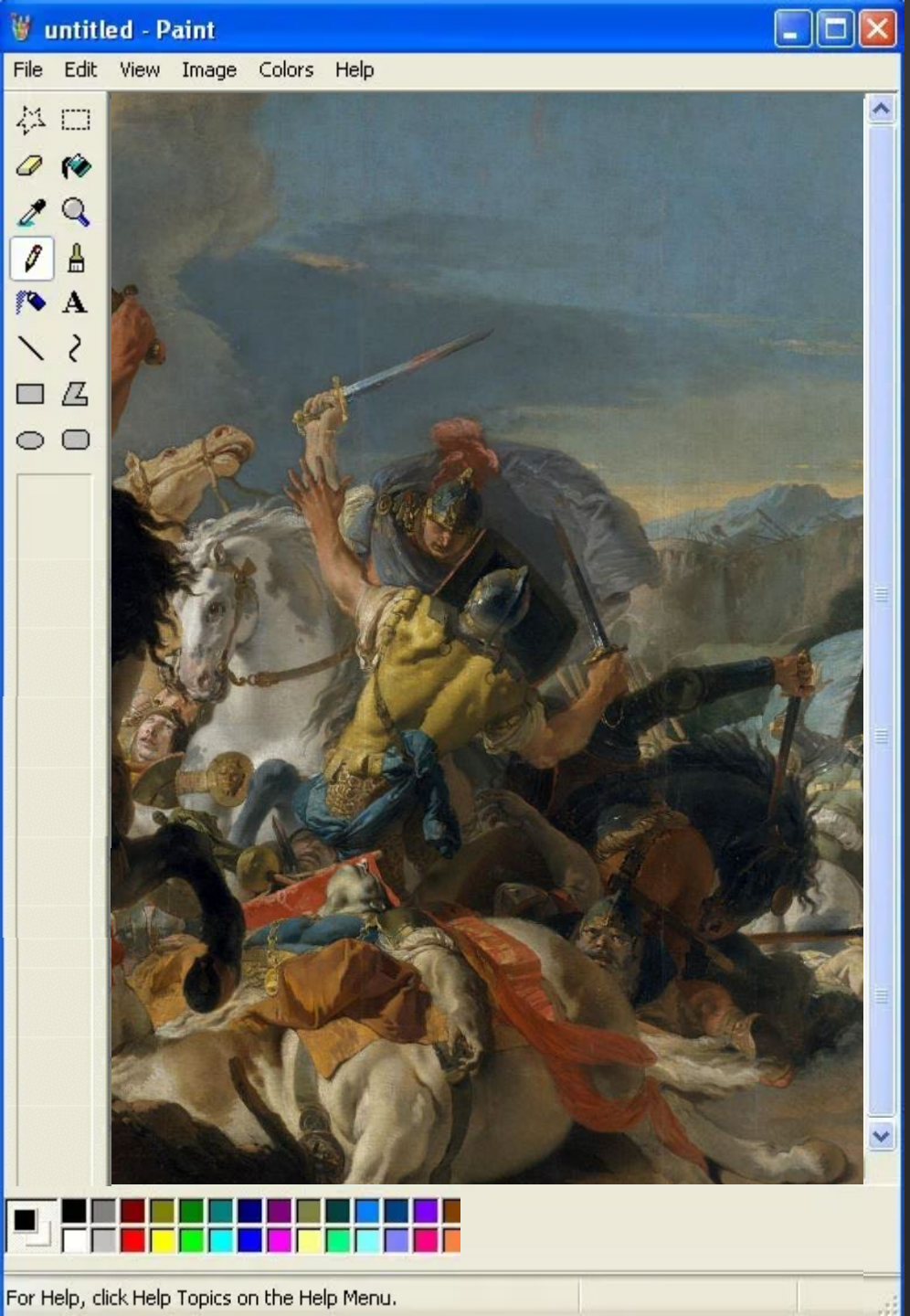
C2 Implants



- C2 communication
- AV/EDR evasion
- Command execution
  - Shell execution
  - PowerShell
  - .NET assemblies
  - Native applications
  - Beacon object files (BOF)
- ...

Red Teaming "History"

(in a nutshell)



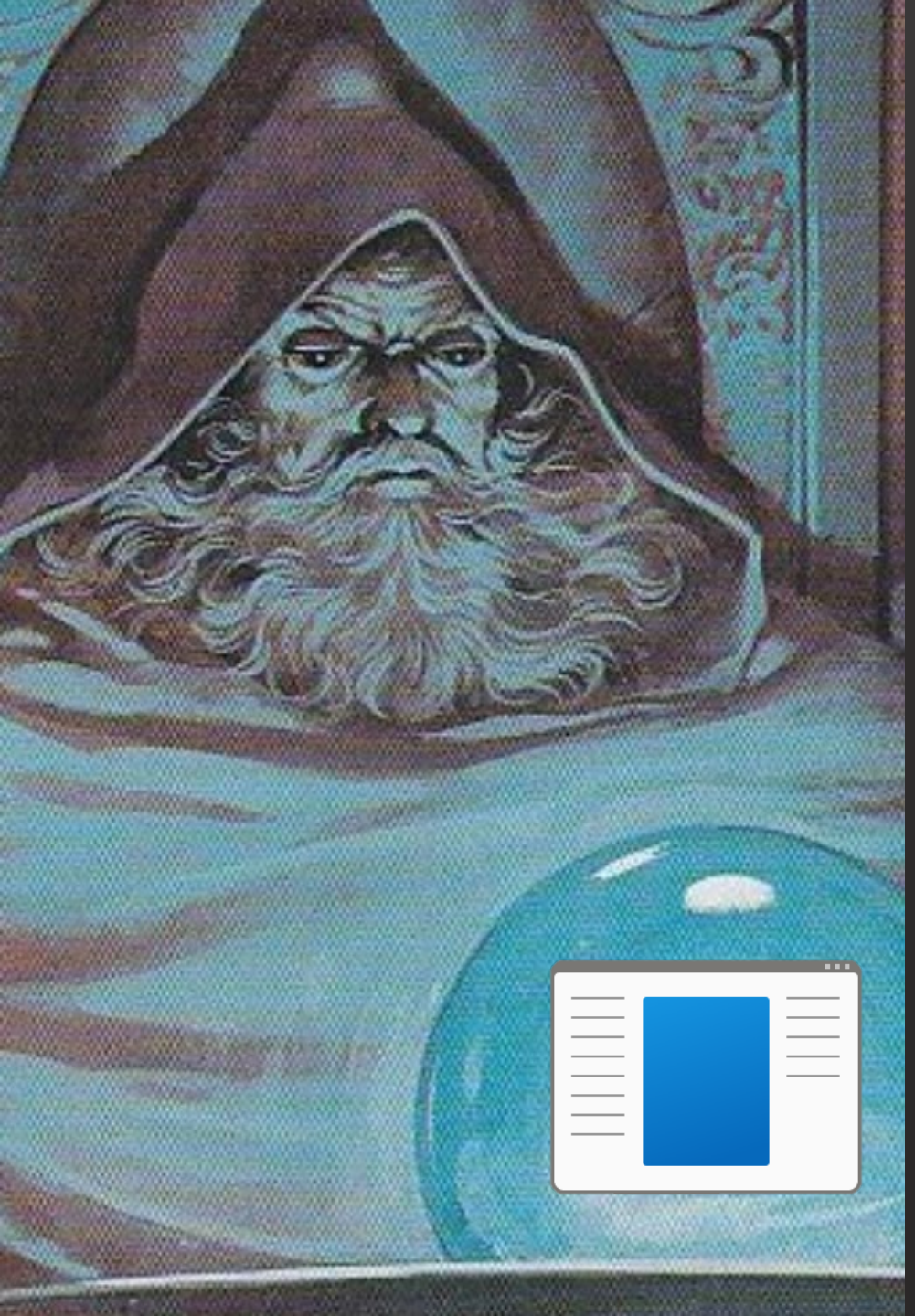
“Basically ancient times”

- Bypass AV’s simple signature detection
  - Packing
  - String obfuscation
- Drop tools onto target machine
- Execute tools from file system

2010s – Rise of the C2s

- In-memory execution of .NET assemblies
  - More complex, require loader/staging
  - Rise of commercial C2s
- Cobalt Strike (CS) emerges
  - $$$ vs stealth capabilities
  - Uses fork & run a bunch

https://cobaltstrike.com/

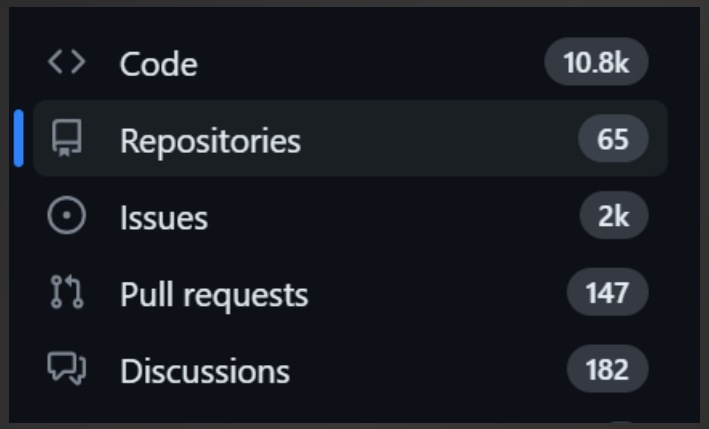


2020s – Rise of the BOFs

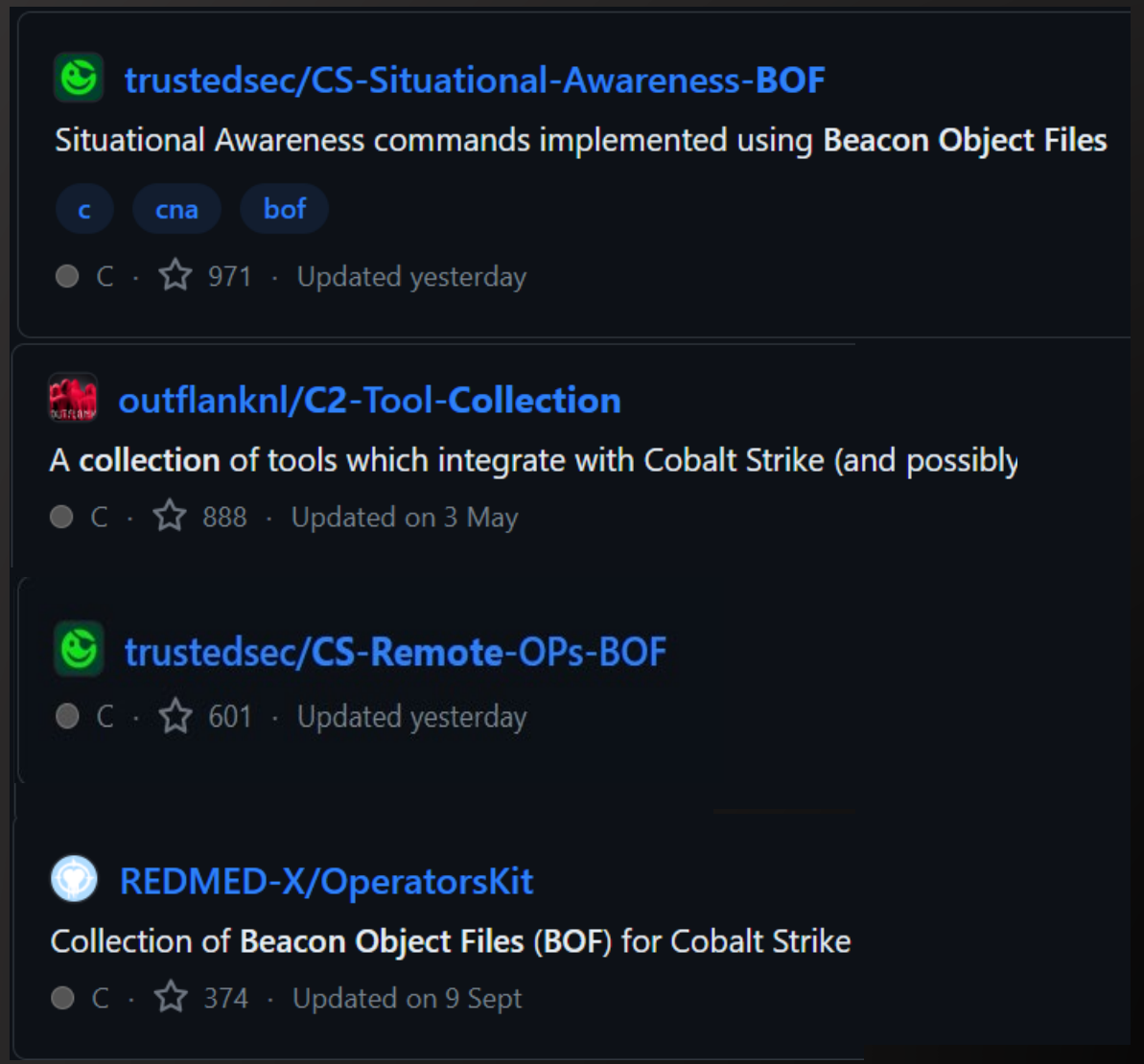
- Fork & Run gets detected a lot
  - CS introduces BOFs in 2020
- EDRs become an increasing challenge
  - Event monitoring & aggregation
  - Behaviour analysis
  - MDE steps up their capabilities
  - CS becomes hard to modify
- Brute Ratel C4 emerges in 2021
  - Security & OPSEC-safety first
  - Inferior feature set

https://bruteratel.com/

Why BOFs?



- whoami
- sc_enum
- env
- ldapsearch
- adcs_request
- ProcessListHandles
- procdump
- office_tokens



Listener ID	Listener Host	External IP	Internal IP	ID	Host	UID	Last Seen (Local)	Last Seen (sec)	PID
https	https://10.20.0.42:8080	10.20.0.69	10.20.0.69,0.0.0.0	b-1	DESKTOP-NB2CP1F	Tim Vic	Tue Sep 12 11:36:59 2023	6	5664

x64 | 5664@b-1 | DESKTOP-NB2CP1F

Command $

Search Text ...

Sentinel $ Perform a quick LDAP query in the current domain or forest, eg.: objectClass=user

Domain $

Wordwrap

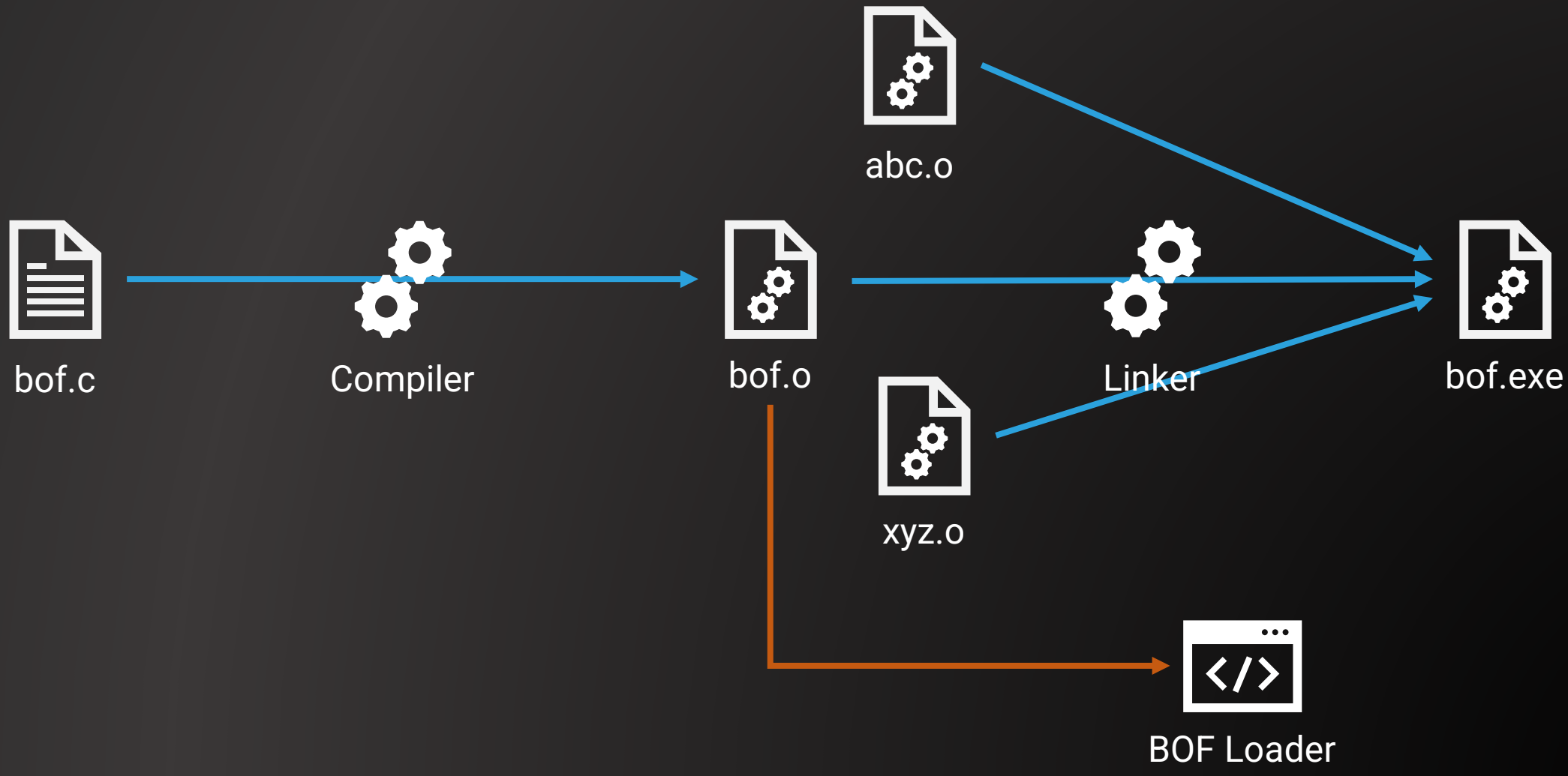
```
wmiexec                    :Creates a new process on local or remote host using WMI with the wminamespace, username and password configured from 'set
wmiconfig' command. Default configuration is 'ROOT\CIMV2'. This command does not return any output

wmiquery                   :Runs a WMI query while using the wminamespace, username and password configured from 'set wmiconfig' command. Default
configuration is 'ROOT\CIMV2'

[ End ] ::::::::::::::::::::::::::::::::
```

user: Tim Vic

sleep: 5:0


Improvise. Adapt. Overcome

About BOFs



About BOFs

bof.c

```c
#include <windows.h>
#include "beacon.h"

void go(char * args, int alen) {
    BeaconPrintf(CALLBACK_OUTPUT, "Hello World: %s", args);
}
```

beacon.h

```c
// [...]
#define CALLBACK_OUTPUT      0x0
#define CALLBACK_OUTPUT_OEM  0x1e
#define CALLBACK_ERROR       0x0d
#define CALLBACK_OUTPUT_UTF8 0x20

DECLSPEC_IMPORT void   BeaconPrintf(int type, char * fmt, ...);
DECLSPEC_IMPORT void   BeaconOutput(int type, char * data, int len);
// [...]
```

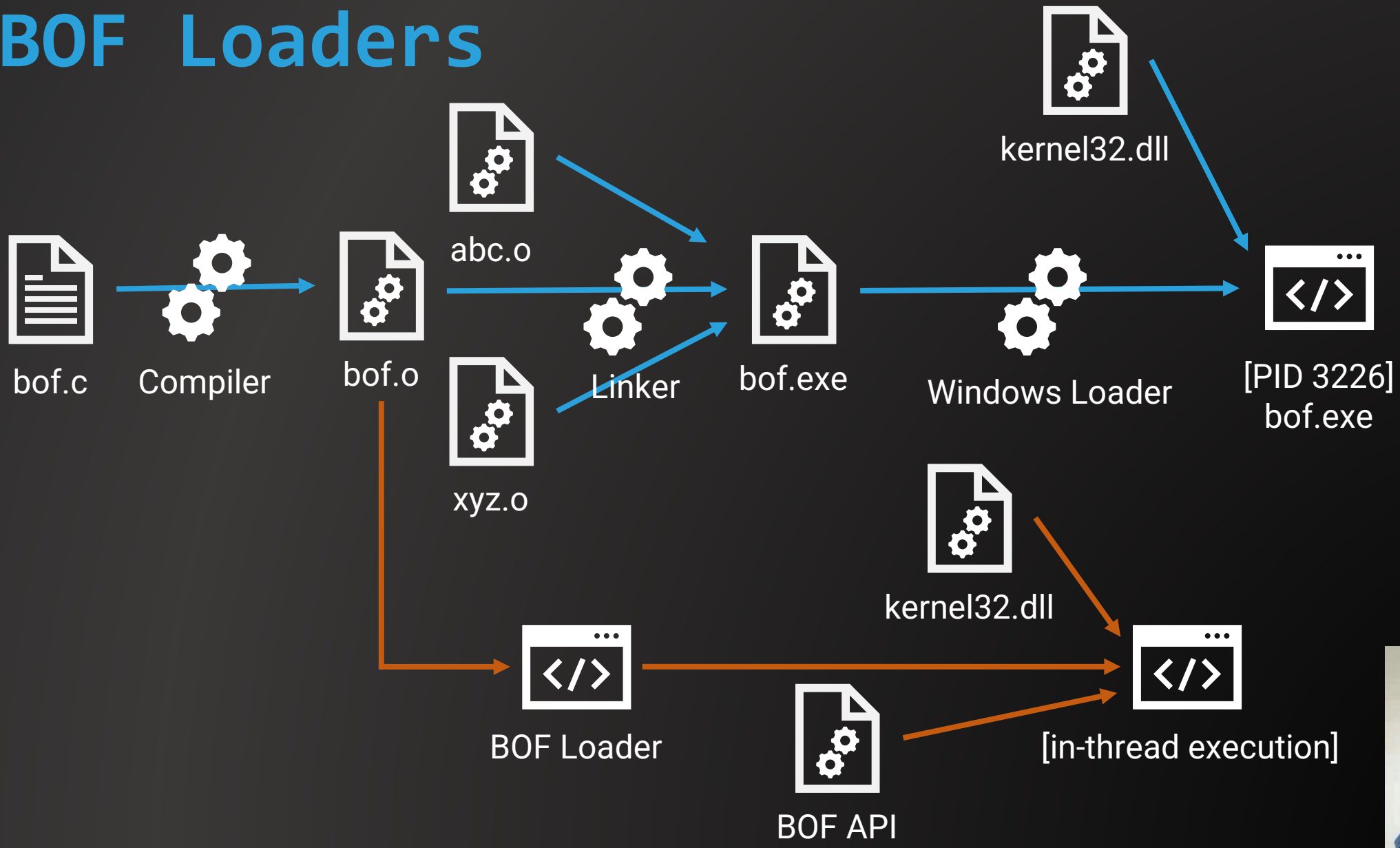
Symbols

bof.o

```
BOOL WINAPI KERNEL32$AttachConsole(DWORD dwProcessId);
BOOL WINAPI KERNEL32$SetConsoleTitleA(LPCSTR lpConsoleTitle);

KERNEL32$AttachConsole(0);
KERNEL32$SetConsoleTitle(“BOF Console”);
BeaconPrintf(CALLBACK_OUTPUT, "Hello World: %s", args);
```

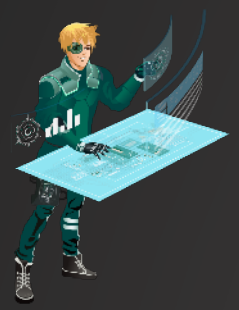
```
SYMBOL TABLE:
[...]
[  2](sec  1)(fl 0x00)(ty   20)(scl   2) (nx 1) 0x0000000000000000 go
AUX tagndx 0 ttlsiz 0x0 lnnos 0 next 0
[...]
[ 18](sec  0)(fl 0x00)(ty    0)(scl   2) (nx 0) 0x0000000000000000 __imp_KERNEL32$AttachConsole
[ 19](sec  0)(fl 0x00)(ty    0)(scl   2) (nx 0) 0x0000000000000000 __imp_KERNEL32$SetConsoleTitleA
[ 20](sec  0)(fl 0x00)(ty    0)(scl   2) (nx 0) 0x0000000000000000 __imp_BeaconPrintf
```

BOF Loaders


kernel32.dll
abc.o
bof.c
Compiler
bof.o
Linker
bof.exe
Windows Loader
[PID 3226]
bof.exe
xyz.o
kernel32.dll
BOF Loader
[in-thread execution]
BOF API


BUT WAIT
THERE'S MORE!

BOF APIs



23 total

Data Parser API	Output API	Format API	Internal APIs
BeaconDataExtract BeaconDataInt BeaconDataLength BeaconDataParse BeaconDataShort	BeaconPrintf BeaconOutput	BeaconFormatAlloc BeaconFormatAppend BeaconFormatFree BeaconFormatInt [+3]	BeaconUseToken BeaconRevertToken BeaconIsAdmin BeaconGetSpawnTo [+4]
	toWideChar		

12 total

“String API”	“Output API”	“Memory API”	BadgerAtoi
BadgerStrLen BadgerWcslen BadgerStrcmp BadgerWcscmp	BadgerDispatch BadgerDispatchW	BadgerAlloc BadgerFree BadgerMemcpy BadgerMemset	BadgerSetdebug

Brute Ratel Documentation

“The coffexec command parses the object file provided by the operator and patches the exported functions on the fly with the internal APIs of badger and windows DLLs.

This makes the port of existing Cobaltstrike BOFs to Brute Ratel extremely easy.”

- https://bruteratel.com/tabs/badger/commands/coffexec/

InPrivate

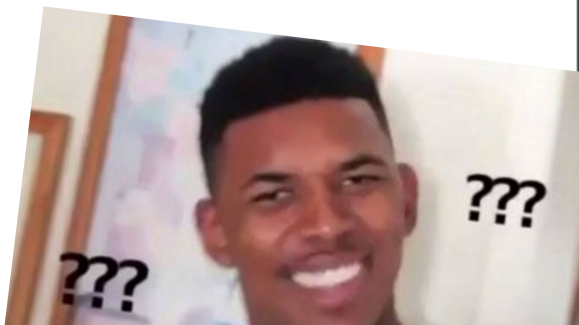
404 Not Found

https://bruteratel.com/tabs/badger/commands/coffexec/

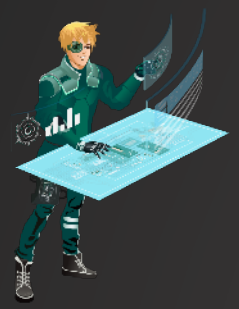
Not Found

The requested URL was not found on this server.

Apache/2.4.41 (Ubuntu) Server at bruteratel.com Port 443



Entrypoints #1



```c
#include <windows.h>
#include "beacon.h"

void go(char* args, int alen) {
    BeaconPrintf(CALLBACK_OUTPUT, "Hello World: %s", args);
}
```

```c
#include <windows.h>
#include "badger_exports.h"

void coffee(char** argv, int argc, WCHAR** dispatch) {
    BadgerDispatch(dispatch, “Hello World: %s\n", argv);
}
```

1. Rewrite entrypoint: go → coffee
2. Replace header files: beacon.h → badger_exports.h
3. Replace uses of CS BOF APIs with BR BOF APIs
4. ???
5. PROFIT

PoC Port Iteration #1

```
2023/01/27 09:12:07 UTC [input] admin => coffexec /home/kali/tools/badger-bofs/sample/getdcname.o

2023/01/27 09:12:15 UTC [sent 1908 bytes]

[*] Task-00 [Thread: 5304]

[*] Coffexec Output:

ecorp.local

+------------------------------------------------------------------+
```

PoC Port Iteration #1

```
2023/02/06 10:35:57 UTC [input] admin => coffexec /home/kali/tools/badger-bofs/C2-Tool-Collection/BOF/Winver/Winver.o

2023/02/06 10:35:57 UTC [sent 4580 bytes]

[*] Task-00 [Thread: 12208]
```

PoC Port Iteration #1

```
_RtlInitUnicodeString RtlInitUnicodeString = (_RtlInitUnicodeString)
        GetProcAddress(GetModuleHandleA("ntdll.dll"), "RtlInitUnicodeString");
if (RtlInitUnicodeString == NULL) {
        return 0;
}
```

https://github.com/outflanknl/C2-Tool-Collection/blob/31eeb66e/BOF/Winver/SOURCE/Winver.c#L38

Dynamic Function Resolution

GetProcAddress, LoadLibraryA, GetModuleHandle, and FreeLibrary are available within BOF files. You have the option to use these to resolve Win32 APIs you wish to call. Another option is to use Dynamic Function Resolution (DFR).

https://hstechdocs.helpsystems.com/manuals/cobaltstrike/current/userguide/content/topics/beacon-object-files_dynamic-func-resolution.htm

1. Rewrite entrypoint: go → coffee
2. Replace header files: beacon.h → badger_exports.h
3. Replace uses of CS BOF APIs with BR BOF APIs
4. Hardcode default Kernel32 imports
5. ???
6. PROFIT

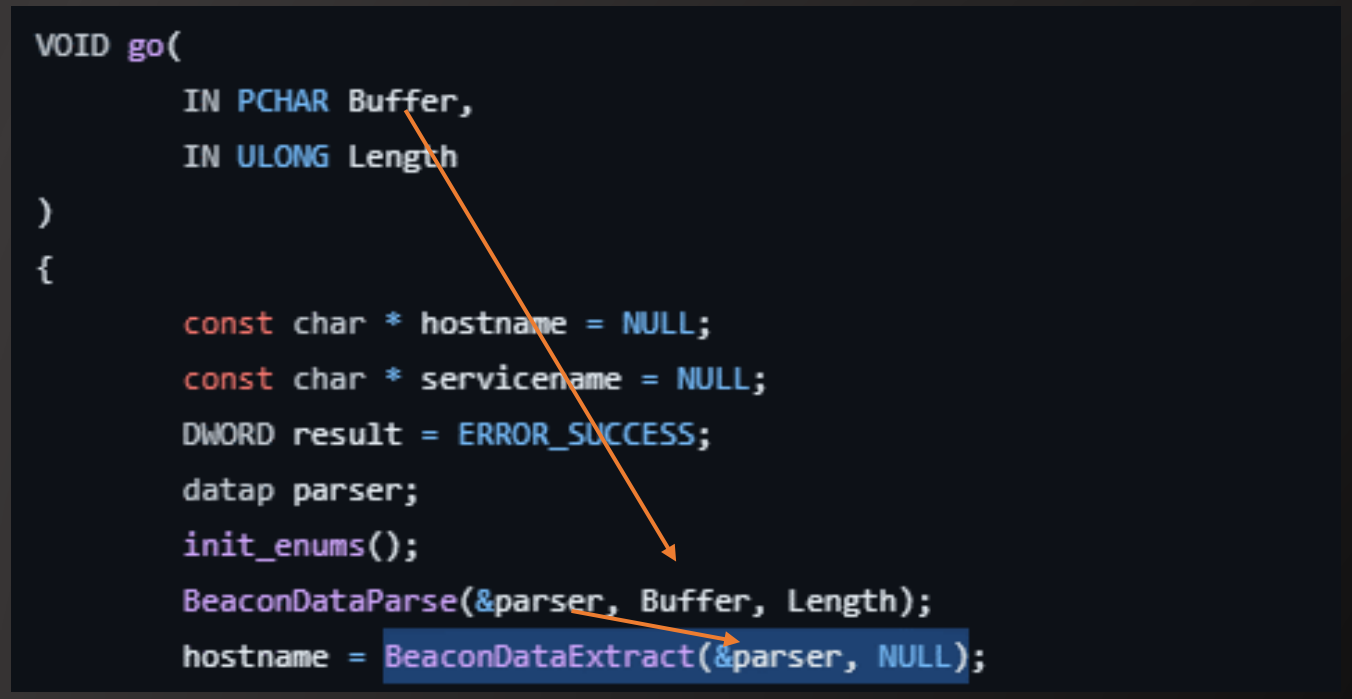
PoC Port Iteration #2

```
2023/02/06 11:04:07 UTC [input] admin => coffexec /home/kali/tools/badger-bofs/C2-Tool-Collection/BOF/Winver/Winver.o

2023/02/06 11:04:07 UTC [sent 4612 bytes]

[*] Task-00 [Thread: 13856]

[*] Coffexec Output:

Windows version: 10.0, OS build number: 19042.1706

+------------------------------------------------------------------+
```

PoC Port Iteration #2

```
2023/01/27 09:17:55 UTC [input] admin => coffexec /home/kali/tools/badger-bofs/SA/sc_enum/sc_enum.x64.o

2023/01/27 09:17:59 UTC [sent 18884 bytes]

[*] Task-00 [Thread: 9124]
```

PoC Port Iteration #2



```
VOID go(
        IN PCHAR Buffer,
        IN ULONG Length
)
{
        const char * hostname = NULL;
        const char * servicename = NULL;
        DWORD result = ERROR_SUCCESS;
        datap parser;
        init_enums();
        BeaconDataParse(&parser, Buffer, Length);
        hostname = BeaconDataExtract(&parser, NULL);
```

https://github.com/trustedsec/CS-Situational-Awareness-BOF/blob/b2bd1fb1/src/SA/sc_enum/entry.c#L404

Entrypoints #2

Arguments: `-n` , `1` , `127.0.0.1`

```
void go(char* args, int alen);
```

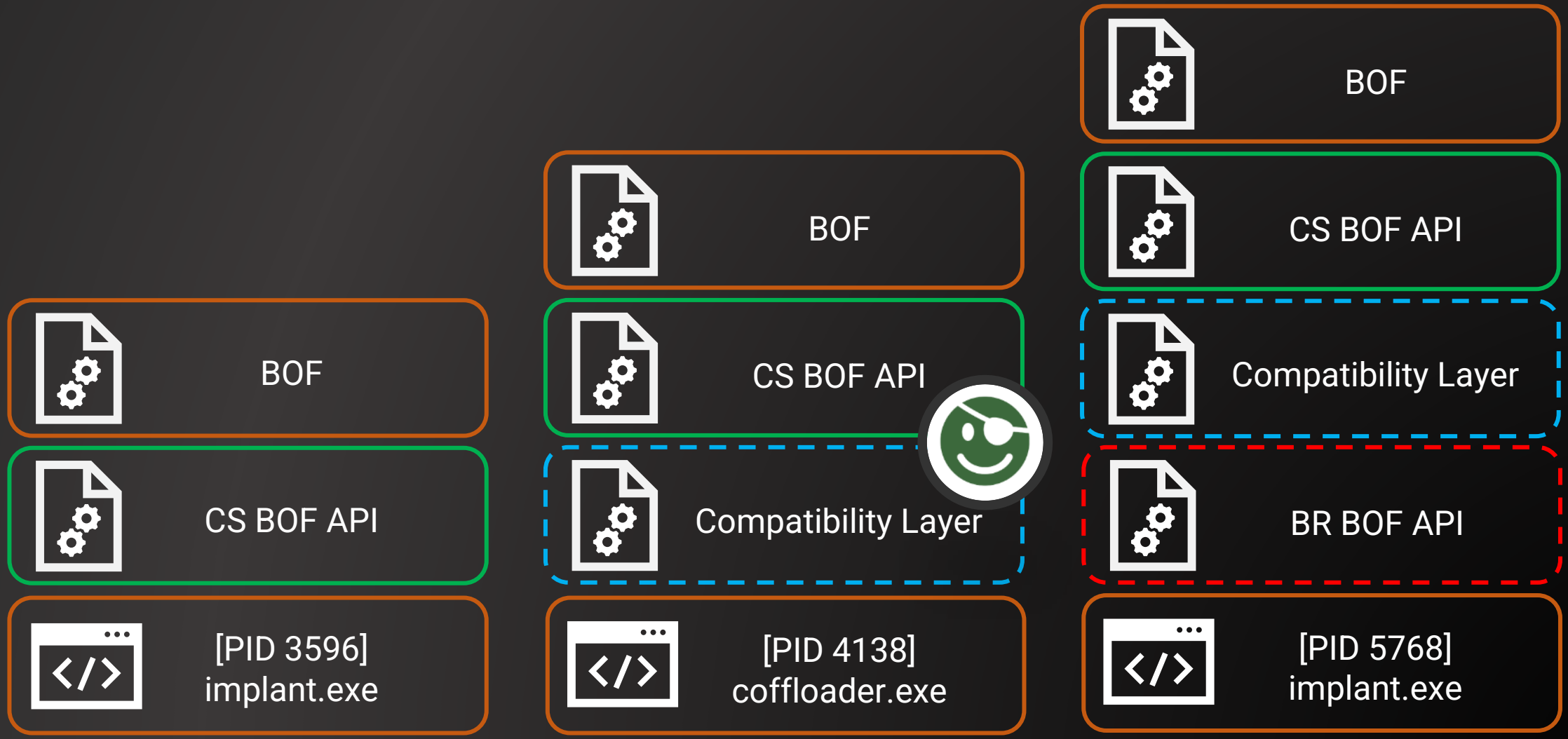
```
args:
03 00 00 00 2d 6e 00 02 00 00 00 31 00 0a 00 00  |....-n.....1....|
00 31 32 37 2e 30 2e 30 2e 31 00                 |.127.0.0.1.|

alen: 27
```

```
void coffee(char** argv, int argc, WCHAR** dispatch);
```

```
argv:
[0]: 0x002ef300487cb240 -> "-n"
[1]: 0x002ef30049a65020 -> "1"
[2]: 0x002ef30049cd7000 -> "127.0.0.1"

argc: 3
```


Me
CS2BR

CS2BR - Compatibility Layer



https://github.com/trustedsec/COFFLoader

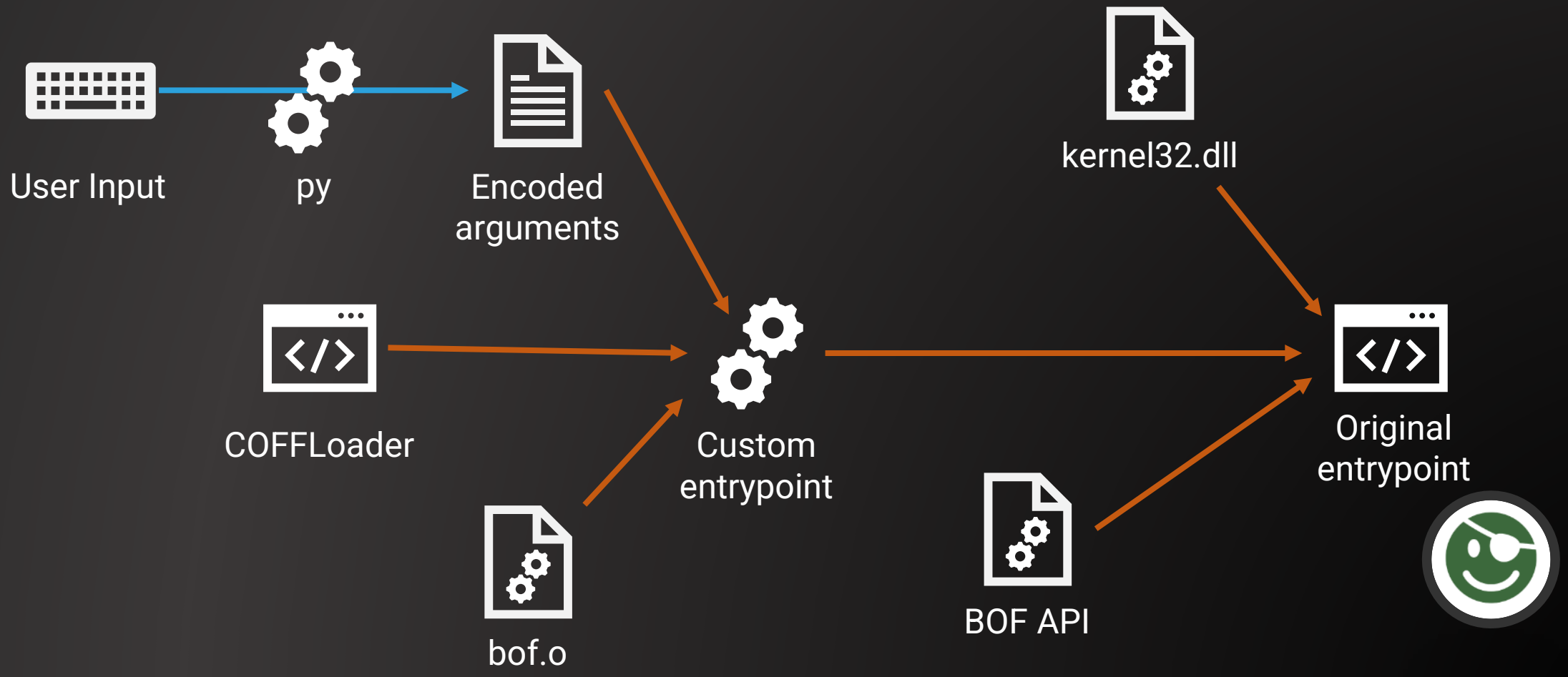
beacon_
compatibility.c

```c
void BeaconPrintf(int type, char* fmt, ...) {
    // [...]
    length = vsnprintf(NULL, 0, fmt, args);
    // [...]
    tempptr = realloc(beacon_compatibility_output, beacon_compatibility_size + length + 1);
    // [...]
    // [...]
    length = vsnprintf(beacon_compatibility_output + beacon_compatibility_offset, length +1,
fmt, args);
    // [...]
    return;
}
```

beacon_
wrapper.c

```c
void BeaconPrintf(int type, char* fmt, ...) {
    // [...]
    length = MSVCRT$vsnprintf(NULL, 0, fmt, args);
    // [...]
    buffer = (char*)BadgerAlloc(length + 1);
    // [...]
    (void)MSVCRT$vsnprintf(buffer, length, fmt, args);
    // [...]
    BadgerDispatch(_dispatch, buffer);
    BadgerFree((void**)&buffer);
    return;
}
```

CS2BR - Entrypoint



CS2BR – Entrypoint wrapper

```
1 void coffee(char **argv, int argc, WCHAR **dispatch)
2 {
3   // [...]
4   // Set global dispatch variable to allow CS-wrappers to use the BR API's output methods
5   _dispatch = dispatch;
6   BadgerDispatch(dispatch, "[cs2br] Starting...\n");
7   // [...]
8   if (argc == 1)
9   {
10    // Decode base64 input
11    // [...]
12  }
13
14  BadgerDispatch(dispatch, "[cs2br] Invoking entrypoint...\n");
15  go(buffer, size);
16
17  BadgerDispatch(dispatch, "[cs2br] Done; exiting!\n");
18
19  if (buffer != NULL)
20     BadgerFree((PVOID *)&buffer);
21 }
```

Demo time!


1 2 3 4
18:54
kali@kali: ~/bofs
File Actions Edit View Help
(kali㉿kali)-[~/bofs]
$
KALI LINUX
"the quieter you become, the more you are able to hear"

x64 | 5664@b-1 | DESKTOP-NB2CP1F

Command $ Search Text ...

Sentinel $ Perform a quick LDAP query in the current domain or forest, eg.: objectClass=user Domain $ Wordwrap

```
userinfo                    :Prints current username, SID, privileges and groups

vault_remove                :Removes a token from Token Vault.

windowlist                  :Displays all hidden and visible windows

wmiexec                     :Creates a new process on local or remote host using WMI with the wminamespace, username and password configured from 'set
wmiconfig' command. Default configuration is 'ROOT\CIMV2'. This command does not return any output

wmiquery                    :Runs a WMI query while using the wminamespace, username and password configured from 'set wmiconfig' command. Default
configuration is 'ROOT\CIMV2'

[ End ] ::::::::::::::::::::::::::::::::

2023/09/12 11:42:43 EDT [input] admin => coffexec /home/kali/bofs/C2-Tool-Collection/BOF/Winver/Winver.x64.o

2023/09/12 11:42:46 EDT [sent 6440 bytes]

[*] Task-0 [Thread: 1616]
```

user: Tim Vic sleep: 5:0

Watchlist | x64 | 444@b-4 | DESKTOP-NB2CP1F | x64 | 5664@b-1 | DESKTOP-NB2CP1F

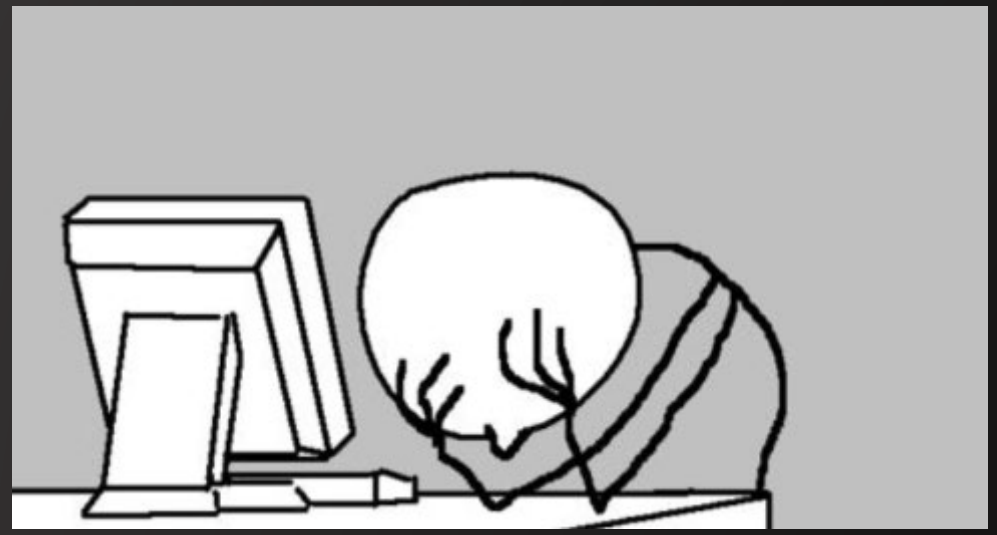
Badger: 2 Pivot: 0 Privileged: 0 Workstations: 1 Operators: 1 Ext IPs: 1

CS2BR – What now?

- ➕ Runs CS BOFs in BR!
- ➖ Requires source code
- ➖ Requires recompilation
- ➖ Requires reading the CNA
- ➖ Binary overhead
- ➖ Bad usability
- ➖ Incomplete API


```
ld --relocatable cs2br.o bof.o -o brc4bof.o
```

```
2023/03/27 08:55:12 UTC [input] admin => coffexec /home/kali/tools/badger-bofs/cs2br-bof-binpatch/minimal.BR_bin20.o

2023/03/27 08:55:12 UTC [sent 13904 bytes]

[*] Task-00 [Thread: 31616]
```



CS2BR - Outlook

Runs CS BOFs in BR!

Learned TONS

BOFs

Object files in general

Sharing is caring!

https://github.com/NVISOsecurity/cs2br-bof

https://blog.nviso.eu/series/introducing-cs2br/


nviso
Questions?
Moritz Thomas
@moritzlthomas
moritzlthomas
Patrick Eisenschmidt
@secdu_de
patrick-eisenschmidt