Jakob Friedl <> BSidesVienna0x7EA

BOFs in the Background

Async object file execution in modern C2 frameworks

> whoami

- Jakob Friedl, MSc
- Offensive Security @ MM Group
- Certification Collector
- Covert Access
- C2 & Malware Development

https://github.com/jakobfriedl/conquest

Definitions

- **C2:** Command & Control; Techniques used by adversaries to communicate with systems under their control in a target network.
- **C2 Framework:** Platform used for managing and controlling multiple compromised systems remotely. Examples: Cobalt Strike, AdaptixC2, Havoc, Mythic, Conquest
- **Agent:** Payload that is executed on a compromised host and calls back to the C2 team server via **Beaconing**.
- **Sleepmask:** Agent memory is encrypted between callbacks.
- **OPSEC:** Operational security; Operate without getting detected by security solutions or blue teamers.
- **Post-Exploitation:** Offensive actions taken after getting an initial foothold on a target system.
- **Fork & Run:** Agent injects post-exploitation capability into a sacrificial process to execute it.

0x01

Beacon Object Files

BOF Basics (1)

Beacon Object File

Introduced in Cobalt Strike 4.1 (2020)

(usually) C2-agnostic

Executed directly in-memory

Usually written in C

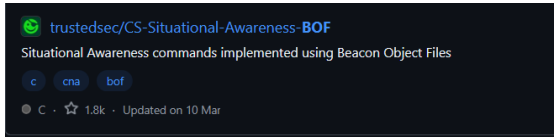
Position-independent COFF object files

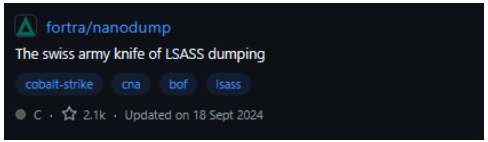
Compact size

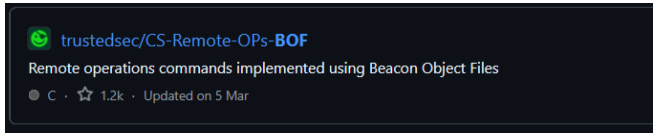
Provide support external post-ex capabilities to keep detection surface of the agent minimal.

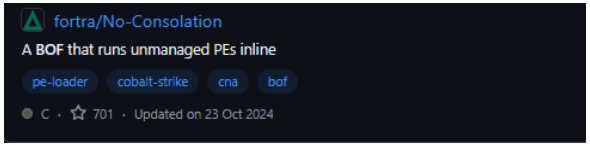
Stealthy | **Lightweight** | **Popular**

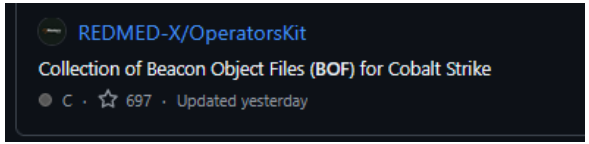
Examples

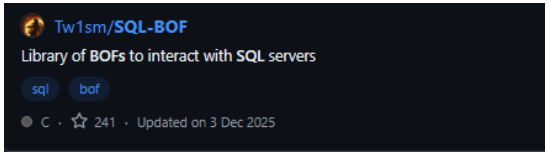

trustedsec/CS-Situational-Awareness-BOF
Situational Awareness commands implemented using Beacon Object Files
c
cna
bof
C · 1.8k · Updated on 10 Mar


fortra/nanodump
The swiss army knife of LSASS dumping
cobalt-strike
cna
bof
lsass
C · 2.1k · Updated on 18 Sept 2024

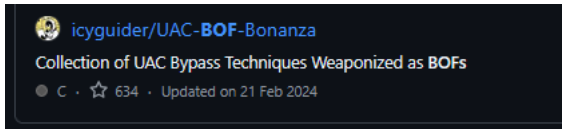

trustedsec/CS-Remote-OPs-BOF
Remote operations commands implemented using Beacon Object Files
C · 1.2k · Updated on 5 Mar

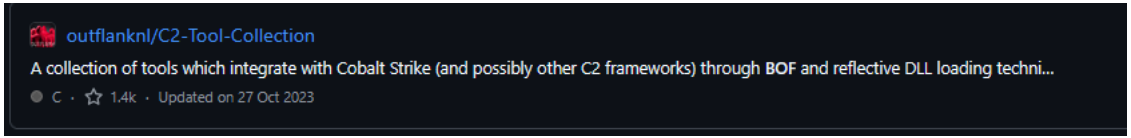

fortra/No-Consolation
A BOF that runs unmanaged PEs inline
pe-loader
cobalt-strike
cna
bof
C · 701 · Updated on 23 Oct 2024


REDMED-X/OperatorsKit
Collection of Beacon Object Files (BOF) for Cobalt Strike
C · 697 · Updated yesterday

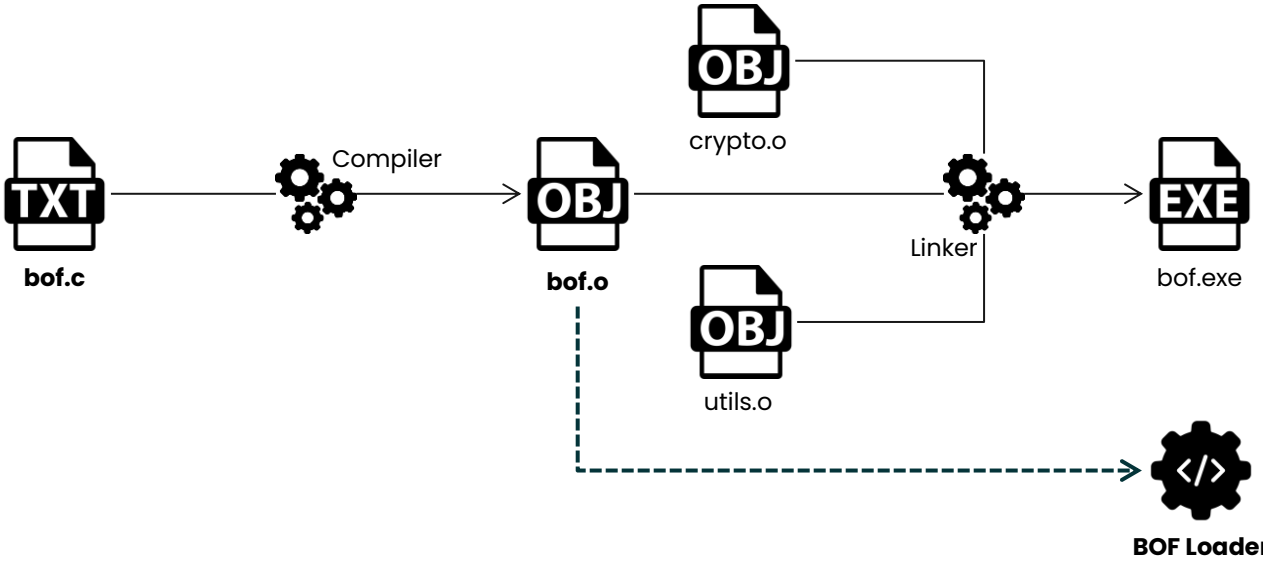

Tw1sm/SQL-BOF
Library of BOFs to interact with SQL servers
sql
bof
C · 241 · Updated on 3 Dec 2025


RalfHacker/Kerbeus-BOF
BOF for Kerberos abuse (an implementation of some important features of the Rubeus).
C · 592 · Updated on 23 Nov 2025


icyguider/UAC-BOF-Bonanza
Collection of UAC Bypass Techniques Weaponized as BOFs
C · 634 · Updated on 21 Feb 2024


outflanknl/C2-Tool-Collection
A collection of tools which integrate with Cobalt Strike (and possibly other C2 frameworks) through BOF and reflective DLL loading techni...
C · 1.4k · Updated on 27 Oct 2023

BOF Basics (2)


TXT
bof.c
Compiler
OBJ
bof.o
OBJ
crypto.o
OBJ
utils.o
Linker
EXE
bof.exe
</>
BOF Loader

BOF Basics (3)

```
root@AURA:/mnt/c/Users/jakob/Documents/Projects/conquest/data/modules/lateral-movement/scshell# objdump -t scshell.x64.o

scshell.x64.o:     file format pe-x86-64

SYMBOL TABLE:
[  0](sec  1)(fl 0x00)(ty   20)(scl   2) (nx 1) 0x0000000000000000 go
AUX tagndx 0 ttlsiz 0x0 lnnos 0 next 0
[  2](sec  1)(fl 0x00)(ty    0)(scl   3) (nx 1) 0x0000000000000000 .text
AUX scnlen 0x719 nreloc 72 nlnno 0
[  4](sec  4)(fl 0x00)(ty    0)(scl   3) (nx 1) 0x0000000000000000 .rdata
AUX scnlen 0x2a4 nreloc 0 nlnno 0
[  6](sec  5)(fl 0x00)(ty    0)(scl   3) (nx 1) 0x0000000000000000 .xdata
AUX scnlen 0xc nreloc 0 nlnno 0
[  8](sec  0)(fl 0x00)(ty    0)(scl   2) (nx 0) 0x0000000000000000 __imp_BeaconDataParse
[  9](sec  0)(fl 0x00)(ty    0)(scl   2) (nx 0) 0x0000000000000000 __imp_BeaconDataExtract
[ 10](sec  0)(fl 0x00)(ty    0)(scl   2) (nx 0) 0x0000000000000000 __imp_BeaconDataLength
[ 11](sec  0)(fl 0x00)(ty    0)(scl   2) (nx 0) 0x0000000000000000 __imp_BeaconPrintf
[ 12](sec  0)(fl 0x00)(ty    0)(scl   2) (nx 0) 0x0000000000000000 __imp_KERNEL32$CreateFileA
[ 13](sec  0)(fl 0x00)(ty    0)(scl   2) (nx 0) 0x0000000000000000 __imp_KERNEL32$GetLastError
[ 14](sec  0)(fl 0x00)(ty    0)(scl   2) (nx 0) 0x0000000000000000 __imp_KERNEL32$WriteFile
[ 15](sec  0)(fl 0x00)(ty    0)(scl   2) (nx 0) 0x0000000000000000 __imp_KERNEL32$CloseHandle
[ 16](sec  0)(fl 0x00)(ty    0)(scl   2) (nx 0) 0x0000000000000000 __imp_ADVAPI32$OpenSCManagerA
[ 17](sec  0)(fl 0x00)(ty    0)(scl   2) (nx 0) 0x0000000000000000 __imp_KERNEL32$DeleteFileA
[ 18](sec  0)(fl 0x00)(ty    0)(scl   2) (nx 0) 0x0000000000000000 __imp_ADVAPI32$OpenServiceA
[ 19](sec  0)(fl 0x00)(ty    0)(scl   2) (nx 0) 0x0000000000000000 __imp_ADVAPI32$CloseServiceHandle
[ 20](sec  0)(fl 0x00)(ty    0)(scl   2) (nx 0) 0x0000000000000000 __imp_ADVAPI32$QueryServiceConfigA
[ 21](sec  0)(fl 0x00)(ty    0)(scl   2) (nx 0) 0x0000000000000000 __imp_KERNEL32$GlobalAlloc
[ 22](sec  0)(fl 0x00)(ty    0)(scl   2) (nx 0) 0x0000000000000000 __imp_ADVAPI32$ChangeServiceConfigA
[ 23](sec  0)(fl 0x00)(ty    0)(scl   2) (nx 0) 0x0000000000000000 __imp_KERNEL32$GlobalFree
[ 24](sec  0)(fl 0x00)(ty    0)(scl   2) (nx 0) 0x0000000000000000 __imp_ADVAPI32$StartServiceA
```

BOF Loading

```
[*] Sleepmask settings: Technique: NONE, Delay: 5341s, Stack spoofing: false
[*] [12-06-2026 15:12:14] Checking in.
[*] Response contained 1 tasks for agent DC801830.
   [>] Executing object file scshell.x64.o.
[*] Virtual size of object file: 24576 bytes
[*] Virtual memory allocated for object file at 0x0000018C73C30000 (24576 bytes)
[*] Copying over sections.
    [>] .text @ 0x0000018C73C30000 (1824 bytes))
    [>] .data @ 0x0000018C73C31000 (0 bytes))
    [>] .bss @ 0x0000018C73C31000 (0 bytes))
    [>] .rdata @ 0x0000018C73C31000 (688 bytes))
    [>] .xdata @ 0x0000018C73C32000 (12 bytes))
    [>] .pdata @ 0x0000018C73C33000 (12 bytes))
    [>] /4 @ 0x0000018C73C34000 (32 bytes))
[*] Processing sections and performing relocations.
    [>] BeaconDataParse @ 0x00007FF7B98F9661
    [>] BeaconDataExtract @ 0x00007FF7B98F94D5
    [>] BeaconDataExtract @ 0x00007FF7B98F94D5
    [>] BeaconDataExtract @ 0x00007FF7B98F94D5
    [>] BeaconDataLength @ 0x00007FF7B98F92E0
    [>] BeaconDataExtract @ 0x00007FF7B98F94D5
    [>] BeaconPrintf @ 0x00007FF7B98F9D36
    [>] BeaconPrintf @ 0x00007FF7B98F9D36
    [>] KERNEL32$CreateFileA @ 0x00007FF99C6070D0
    [>] KERNEL32$GetLastError @ 0x00007FF99C5C8640
    [>] BeaconPrintf @ 0x00007FF7B98F9D36
    [>] KERNEL32$WriteFile @ 0x00007FF99C6075B0
    [>] KERNEL32$GetLastError @ 0x00007FF99C5C8640
    [>] BeaconPrintf @ 0x00007FF7B98F9D36
    [>] KERNEL32$CloseHandle @ 0x00007FF99C606E30
    [>] KERNEL32$CloseHandle @ 0x00007FF99C606E30
```

1. Calculate memory
2. Allocate memory
3. Copy over sections
4. Perform relocations
5. Execute entry point
6. Retrieve output

BOF API

- Cobalt Strike's de-facto standard for BOF development
- `#include "beacon.h"`
- Implemented by the BOF Loader

Data Parser API	Output API	Format API	Internal API
Extract and unpack BOF arguments	*Return output to the C2 agent*	*Build large or repeating output*	*Miscellaneous functionality*
BeaconDataParse BeaconDataExtract BeaconDataInt BeaconDataShort BeaconDataLength BeaconDataPtr	BeaconPrintf BeaconOutput BeaconDownload	BeaconFormatAlloc BeaconFormatAppend BeaconFormatFree BeaconFormatPrintf BeaconFormatReset BeaconFormatToString	BeaconUseToken BeaconRevertToken BeaconIsAdmin BeaconInjectProcess toWideChar BeaconAddValue BeaconGetValue …

https://hstechdocs.helpsystems.com/manuals/cobaltstrike/current/userguide/content/topics/beacon-object-files_bof-c-api.htm

BOF Arguments (1)

BOF Arguments need to be **packed** into a specific format		
z	NULL-terminated string (ANSI)	`BeaconDataExtract`
Z	NULL-terminated wide string (UTF-16)	`(wchar_t*)BeaconDataExtract`
i	Integer	`BeaconDataInt`
s	Short	`BeaconDataShort`
b	Raw bytes	`BeaconDataExtract`

```
PS C:\Users\jakob\Documents\Projects> python3 .\beacon_generate.py
Beacon Argument Generator
Beacon>help

Documented commands (type help <topic>):
========================================
addString  addWString  addint  addshort  exit  generate  help  reset

Beacon>addString arg1
Beacon>addWString arg2
Beacon>addint 15
Beacon>generate
b'1b0000000500000061726731000a0000006100720067003200000000f000000'
Beacon>
```

https://github.com/trustedsec/COFFLoader/blob/main/beacon_generate.py

Example: LDAPSearch (1)

```
VOID go(
    IN PCHAR Buffer,
    IN ULONG Length
)
{

    datap  parser;
    char * ldap_filter;
    char * ldap_attributes;
    char * hostname;
    char * domain;
    ULONG results_count;
    ULONG scope_of_search;
    ULONG ldaps;

    BeaconDataParse(&parser, Buffer, Length);
    ldap_filter = BeaconDataExtract(&parser, NULL);
    ldap_attributes = BeaconDataExtract(&parser, NULL);
    results_count = BeaconDataInt(&parser);
    scope_of_search = BeaconDataInt(&parser);
    hostname = BeaconDataExtract(&parser, NULL);
    domain = BeaconDataExtract(&parser, NULL);
    ldaps = BeaconDataInt(&parser);

    ldap_attributes = *ldap_attributes == 0 ? NULL : ldap_attributes;
    hostname = *hostname == 0 ? NULL : hostname;
    domain = *domain == 0 ? NULL : domain;


    if(!bofstart())
    {
        return;
    }

    ldapSearch(ldap_filter, ldap_attributes, results_count, scope_of_search, hostname, domain, ldaps==1);
```

z z i i z z i

https://github.com/trustedsec/CS-Situational-Awareness-BOF/blob/master/src/SA/ldapsearch/entry.c

Example: LDAPSearch (2)

```
cmd_ldapsearch = (
    conquest.createCommand(name="ldapsearch", description="Execute a LDAP query.",
                           example="ldapsearch \"(objectClass=user)\" --attributes *,ntsecuritydescriptor --dn DC=conquest,DC=local --dc dc01.conquest.local",
                           message="Tasked agent to execute a LDAP query.", mitre=["T1087.002", "T1069.002", "T1482", "T1018"])
            .addArgString("query", "LDAP filter query.", True)
            .addFlagString("--attributes", "attributes", "Attributes to retrieve, comma-separated (default: *).", False, "*")
            .addFlagInt("--count", "count", "Maximum number of results (default: 0 = unlimited).")
            .addFlagString("--scope", "scope", """Search scope.
Available options:
  - base
  - level
  - subtree (default)""", False, "subtree")
            .addFlagString("--dc", "hostname", "Hostname or IP of domain controller (default: default domain controller).")
            .addFlagString("--dn", "dn", "LDAP query base DN (default: current domain).")
            .addFlagBool("--ldaps", "Use LDAPS on port 636 instead of LDAP on port 389.")
            .addFlagBool("--raw", "Return raw ntSecurityDescriptor field (base64) instead of human readable output.")
            .setHandler(_ldapSearch)
            .setOutputHandler(_ldapSearchOutput)
).registerToGroup("situational awareness")
```

Example: LDAPSearch (3)

```python
LDAP_SCOPE = {
    "base": 1,
    "level": 2,
    "subtree": 3
}
def _ldapSearch(agentId, cmdline, args):
    query = conquest.get_string(args, 0)
    attributes = conquest.get_string(args, 1)
    count = conquest.get_int(args, 2)
    scope = conquest.get_string(args, 3)
    hostname = conquest.get_string(args, 4)
    dn = conquest.get_string(args, 5)
    ldaps = conquest.get_bool(args, 6)

    # Use a global variable to set the option for the output handler
    global RAW
    RAW = conquest.get_bool(args, 7)

    ldapScope = LDAP_SCOPE.get(scope, LDAP_SCOPE["subtree"])

    bof = os.path.join(SCRIPT_DIR, "CS-Situational-Awareness-BOF/SA/ldapsearch/ldapsearch.x64.o")
    params = conquest.bof_pack("zziizzi", [
        query,                  # z: LDAP filter query
        attributes,             # z: Attributes (comma-separated or "*")
        count,                  # i: Max results (0 = unlimited)
        ldapScope,              # i: Scope (1=base, 2=level, 3=subtree)
        hostname,               # z: DC hostname (auto-discover if empty)
        dn,                     # z: Domain DN (auto-detect if empty)
        int(ldaps)              # i: Use LDAPS
    ])

    if os.path.exists(bof):
        conquest.execute_alias(agentId, cmdline, f"bof {bof} {params}")
    else:
        conquest.error(agentId, f"Failed to open object file: {bof}", cmdline)
```

Example: LDAPSearch (4)

```python
# Output handler
def _ldapSearchOutput(agentId, output):
    if not IMPACKET_AVAILABLE:
        conquest.warn(agentId, "Missing pip dependency: impacket. Install impacket to convert ntSecurityDescriptor into human-readable output.")

    result = []
    for line in output.split('\n'):
        # Process group memberships
        if line.lower().startswith('memberof:'):
            key, _, groups = line.partition(':')
            result.append(key + ':')
            for group in groups.strip().split(', '):
                result.append('  ' + group)

        # Process ntSecurityDescriptor if python3-impacket is installed
        # Does not parse the field if the global RAW variable was set by the command handler
        elif not RAW and IMPACKET_AVAILABLE and line.lower().startswith('ntsecuritydescriptor:'):
            key, _, sd_b64 = line.partition(':')
            result.append(key + ':')
            result.append(_decode_security_descriptor(sd_b64.strip()))

        else:
            result.append(line)

    conquest.output(agentId, '\n'.join(result))
```

Raw

```
Listeners | [332A9326] CONQUEST\Administrator @ D... | Script Manager
CONQUEST\Administrator@DC01.conquest.local | 192.168.168.10 | 6588/monarch.http_x64.exe
[*] Distinguished name: DC=conquest,DC=local
[*] Binding to dc01.conquest.local
[+] Successfully bound to LDAP server
[*] Filter: (samaccountname=julius)
[*] Scope of search value: 3
[*] Returning specific attribute(s): *,ntsecuritydescriptor

-------------------
objectClass: top, person, organizationalPerson, user
cn: Julius Caesar
sn: Caesar
givenName: Julius
distinguishedName: CN=Julius Caesar,CN=Users,DC=conquest,DC=local
instanceType: 4
whenCreated: 20251029171820.0Z
whenChanged: 20260602134528.0Z
displayName: Julius Caesar
uSNCreated: 16430
memberOf:
  CN=Domain Admins,CN=Users,DC=conquest,DC=local
uSNChanged: 262230
nTSecurityDescriptor: AQAEnIgEAACkBAAAAAAAABQAAAAEAHQEGAAAAAUAPAAQAAAAAwAAAABCFkzAINARp2gAqgBuBSkUzChINxS8RZsHrW8B
name: Julius Caesar
objectGUID: 699a2503-601c-4847-b26a-9d4545291e88
userAccountControl: 66048
badPwdCount: 0
codePage: 0
countryCode: 0
badPasswordTime: 134151445280536596
lastLogoff: 0
lastLogon: 134249430006885684
pwdLastSet: 134062327616547497
primaryGroupID: 513
objectSid: S-1-5-21-2062779629-1118245407-2952427100-1108
adminCount: 1
accountExpires: 9223372036854775807
logonCount: 132
sAMAccountName: julius
sAMAccountType: 805306368
userPrincipalName: julius@conquest.local
objectCategory: CN=Person,CN=Schema,CN=Configuration,DC=conquest,DC=local
dSCorePropagationData: 20251029172818.0Z, 16010101000000.0Z
lastLogonTimestamp: 134248815286709629
```

Parsed

```
[DF811E3F] CONQUEST\Administrator @ D...
CONQUEST\Administrator@DC01.conquest.local | 192.168.168.10 | 3232/monarch.http_x64.exe
[*] Distinguished name: DC=conquest,DC=local
[*] Binding to dc01.conquest.local
[+] Successfully bound to LDAP server
[*] Filter: (samaccountname=julius)
[*] Scope of search value: 3
[*] Returning specific attribute(s): *,ntsecuritydescriptor

-------------------
objectClass: top, person, organizationalPerson, user
cn: Julius Caesar
sn: Caesar
givenName: Julius
distinguishedName: CN=Julius Caesar,CN=Users,DC=conquest,DC=local
instanceType: 4
whenCreated: 20251029171820.0Z
whenChanged: 20260602134528.0Z
displayName: Julius Caesar
uSNCreated: 16430
memberOf:
  CN=Domain Admins,CN=Users,DC=conquest,DC=local
uSNChanged: 262230
nTSecurityDescriptor:
  Owner: Domain Admins
  Group: Domain Admins
  DACL:
    [Allowed] BUILTIN\Pre-Windows 2000 Compatible Access: ReadProperty
    [Allowed] BUILTIN\Pre-Windows 2000 Compatible Access: ReadProperty
    [Allowed] BUILTIN\Pre-Windows 2000 Compatible Access: ReadProperty
    [Allowed] BUILTIN\Pre-Windows 2000 Compatible Access: ReadProperty
    [Allowed] BUILTIN\Pre-Windows 2000 Compatible Access: ReadProperty
    [Allowed] BUILTIN\Pre-Windows 2000 Compatible Access: ReadProperty
    [Allowed] BUILTIN\Pre-Windows 2000 Compatible Access: ReadProperty
    [Allowed] BUILTIN\Pre-Windows 2000 Compatible Access: ReadProperty
    [Allowed] BUILTIN\Pre-Windows 2000 Compatible Access: ReadProperty
    [Allowed] BUILTIN\Pre-Windows 2000 Compatible Access: ReadProperty
    [Allowed] Cert Publishers: WriteProperty, ReadProperty
    [Allowed] BUILTIN\Windows Authorization Access Group: ReadProperty
    [Allowed] BUILTIN\Terminal Server License Servers: WriteProperty, ReadProperty
    [Allowed] BUILTIN\Terminal Server License Servers: WriteProperty, ReadProperty
    [Allowed] BUILTIN\Pre-Windows 2000 Compatible Access: ReadControl, ListObject, ListChildren, ReadProperty
    [Allowed] BUILTIN\Pre-Windows 2000 Compatible Access: ReadControl, ListObject, ListChildren, ReadProperty
    [Allowed] Everyone: UserChangePassword
    [Allowed] Self: UserChangePassword
```

BOF Limitations

- BOF crashes -> Agent crashes (not good)
- Long execution blocks the agent and renders it unresponsive (also not good)
- OPSEC risks since Sleepmask is not applied (again, not good)

- The main limitation of synchronous BOFs is… **they're synchronous**

- Work-arounds (without sacrificing the agent)
  - Post-ex DLLs (Fork & Run -> bad)
  - Process Injection (noisy)
  - Something new…

0x02

Async BOFs

Async BOFs (1)

- Major addition to the BOF standard through collaboration between **Outflank** (OutflankC2), **Fortra** (Cobalt Strike) and **MDSec** (Nighthawk) in 2025

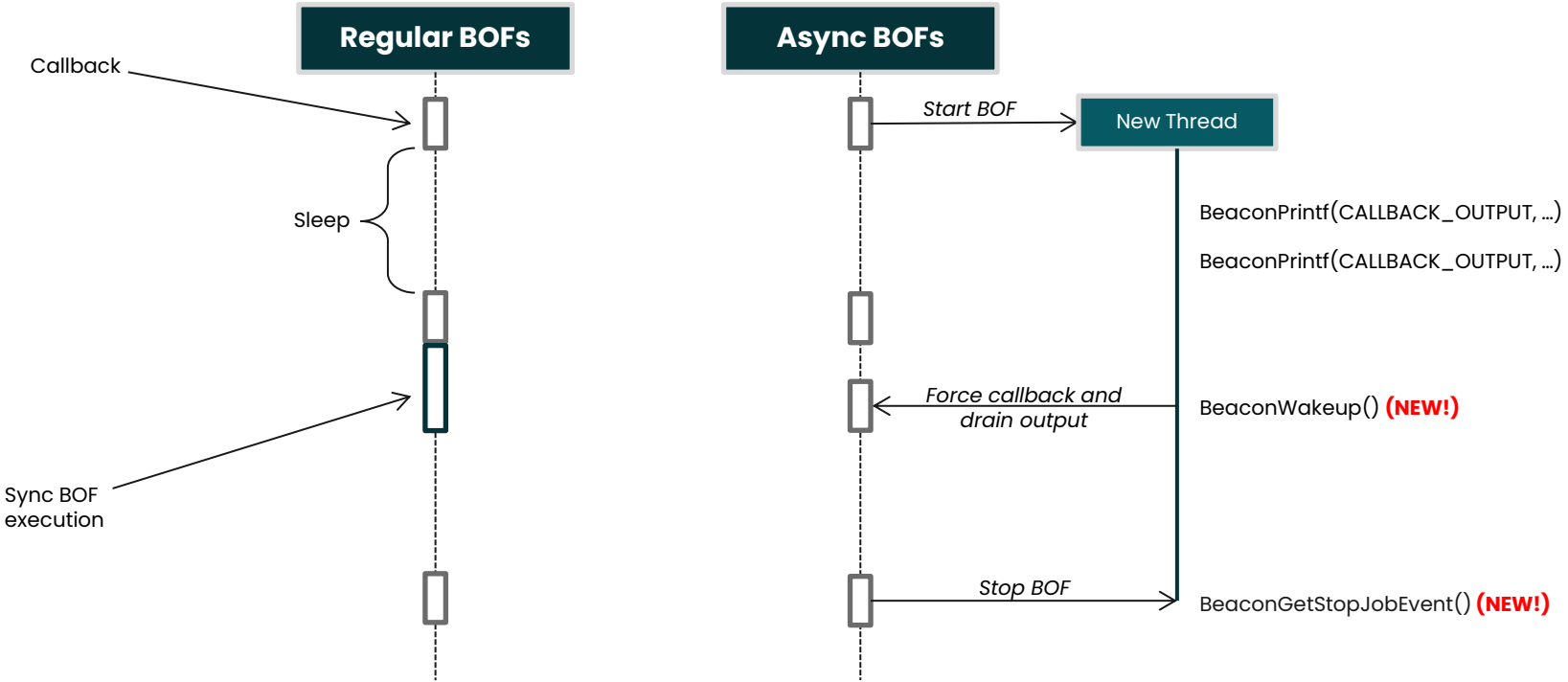
Async BOFs enable BOF execution in the background without blocking the agent's main execution thread.

Long-running Tasks	Event Monitoring	Parallel Execution	Better OPSEC

- **Downsides**
  - Almost exclusively implemented in commercial C2s ($$$)
  - Lack of public Async BOFs
  - Tricky to integrate into existing frameworks

https://www.outflank.nl/blog/2025/07/16/async-bofs-wake-me-up-before-you-go-go/

Async BOFs (2)



https://www.outflank.nl/blog/2025/07/16/async-bofs-wake-me-up-before-you-go-go/

Async BOFs (3)

Some things to consider:

- Synchronization
  - Wakeup
  - Output
  - Stop

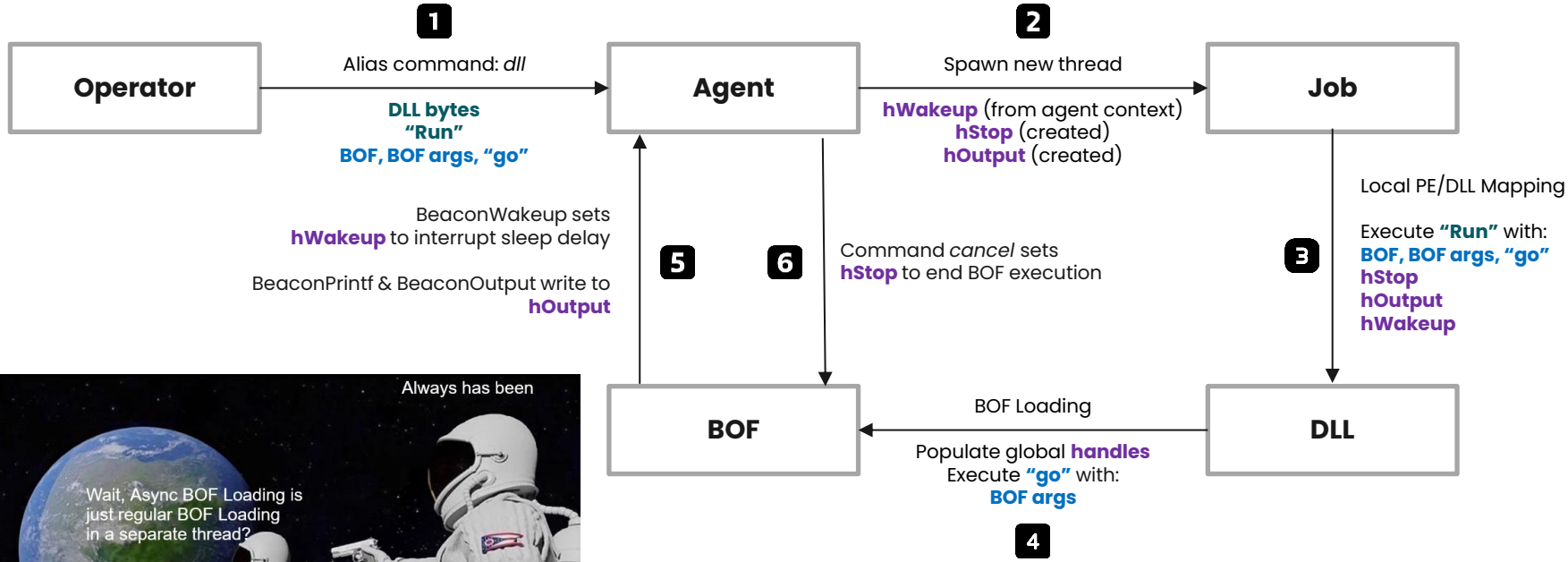
- Sleepmask problem
  - Beacon APIs live in agent memory
  - Agent memory is encrypted during sleep
  - Async BOF tries to access Beacon APIs
  - Crash (not good)

Implementation Approach

The Conquest solution:

1. Create a **background job system** for executing long-running tasks in a separate thread
2. Create a **DLL loading** command that uses the job system
3. Create a fully **self-contained BOF loader** as a Windows DLL
4. Setup **inter-thread communication** between BOF and Agent
5. Profit

Async BOF Loading



0x03

Async BOF Use Cases

Discovery

Async Port Scanning

- Discover open ports on target system(s)
- Synchronous version exists, but sucks when scanning large ranges
- Cobalt Strike injects port scanner into remote process

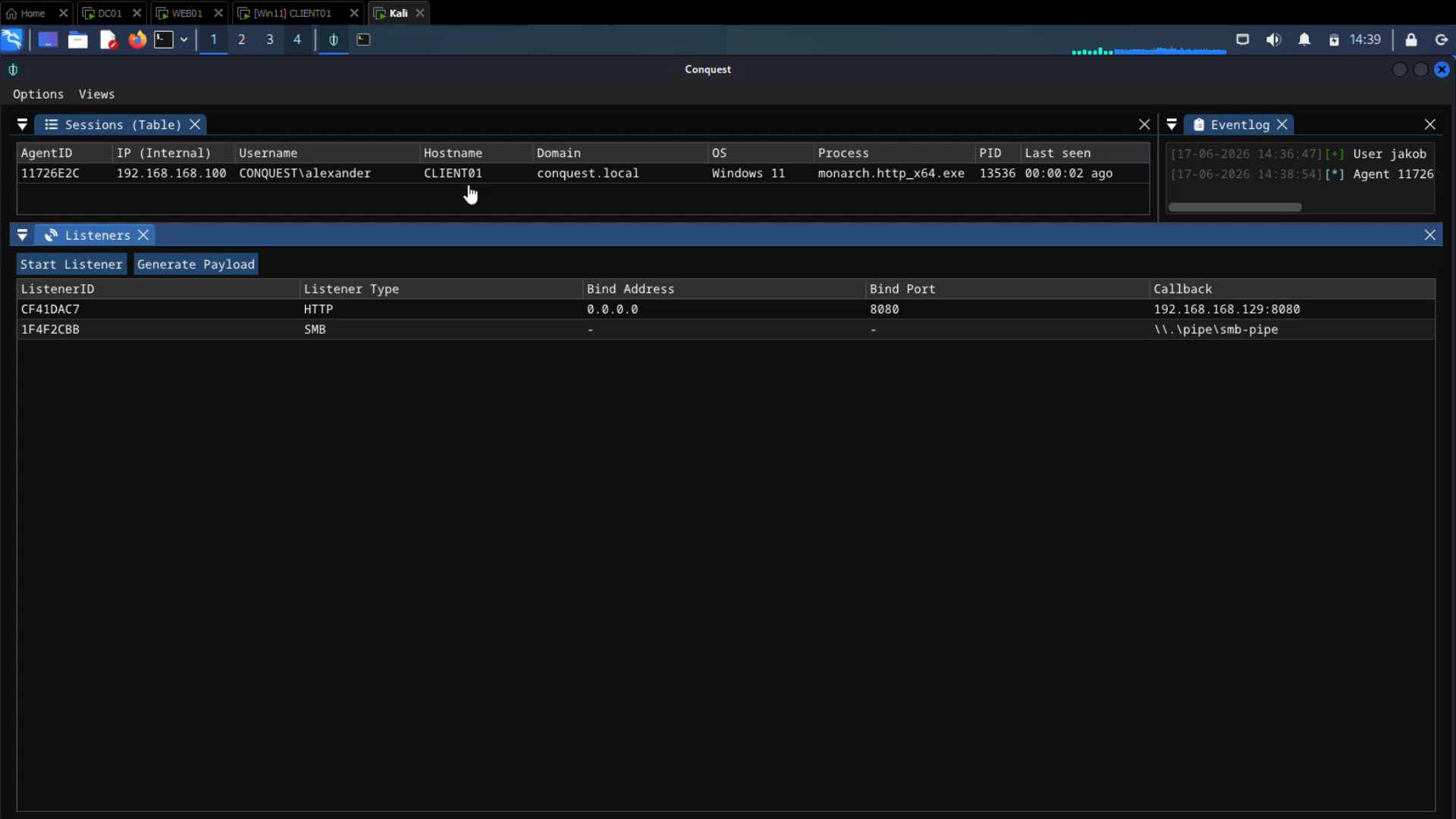
```
[17-06-2026 14:54:04][>>>>] help asyncscan
Scan target systems for open ports (async).
MITRE ATT&CK: T1046

Usage: asyncscan <targets> [ports] [--timeout timeout] [--max-conn max-conn] [--verbose]
Example: asyncscan 192.168.168.0/24 1-1000,8443 --verbose

Required arguments:
  targets                   STRING     Comma-separated list of targets to scan. Use `-` or CIDR notation to specify IP ranges (e.g. 192.168.1.0-128,192.168.1.200,10.0.1.0/24).

Optional arguments:
  ports                     STRING     Comma-separated list of ports to check. Use `-` to specify port ranges (default: nmap top 1000).
  --timeout timeout         INT        Maximum time to wait per poll cycle for connections to respond in ms (default: 500).
  --max-conn max-conn       INT        Number of concurrent connections the port scanner maintains at one time (default: 1024).
  --verbose                 BOOL       Report open ports as they are discovered (default: false).
```

https://github.com/jakobfriedl/asyncscan-bof/


Home
DC01
WEB01
[Win11] CLIENT01
Kali
1
2
3
4
14:39
Conquest
Options
Views
Sessions (Table)
AgentID
IP (Internal)
Username
Hostname
Domain
OS
Process
PID
Last seen
11726E2C
192.168.168.100
CONQUEST\alexander
CLIENT01
conquest.local
Windows 11
monarch.http_x64.exe
13536
00:00:02 ago
Eventlog
[17-06-2026 14:36:47][+] User jakob
[17-06-2026 14:38:54][*] Agent 11726
Listeners
Start Listener
Generate Payload
ListenerID
Listener Type
Bind Address
Bind Port
Callback
CF41DAC7
HTTP
0.0.0.0
8080
192.168.168.129:8080
1F4F2CBB
SMB
-
-
\\.\pipe\smb-pipe

Situational Awareness

User Logon Monitoring

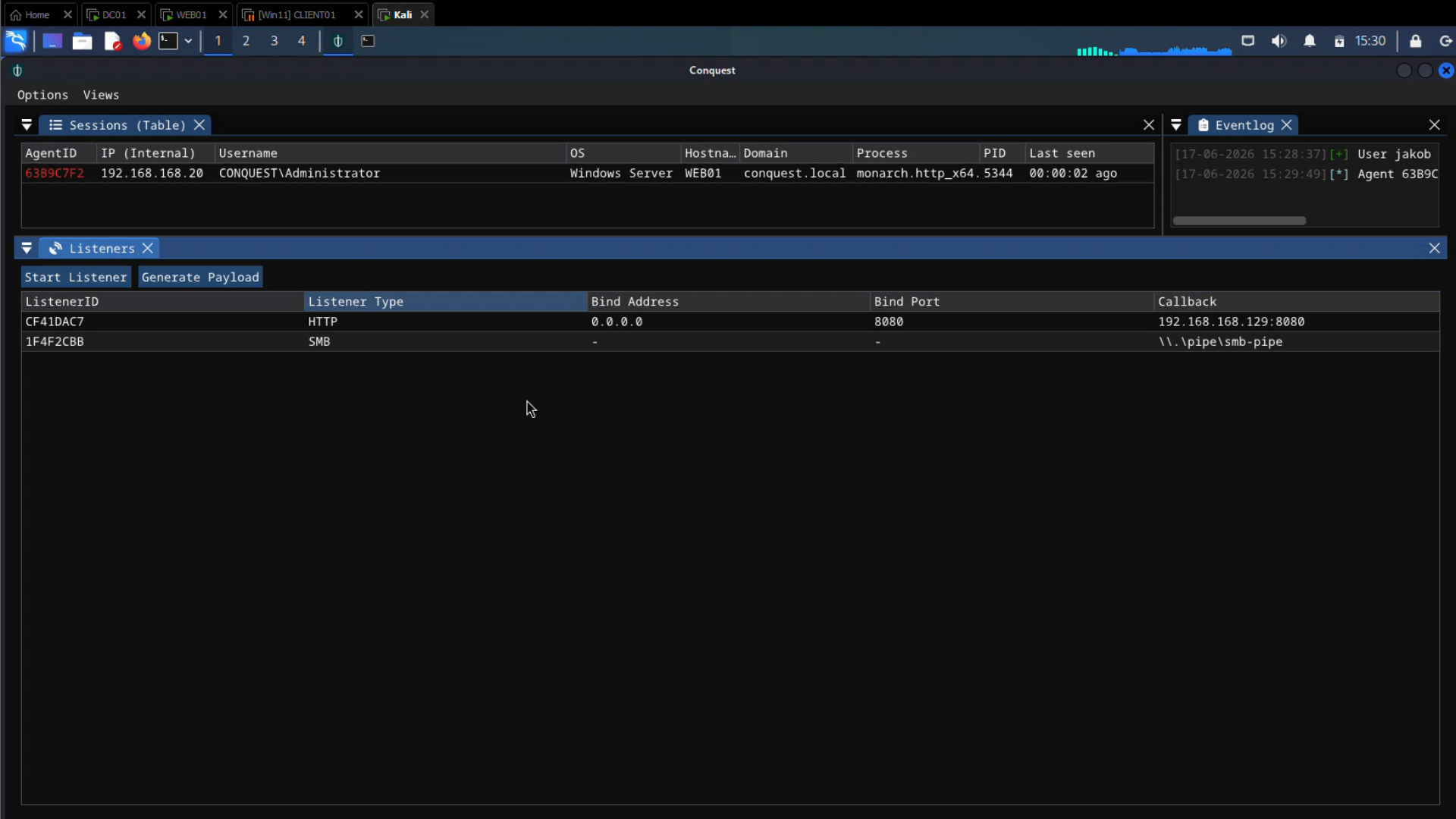
- Notify when a user session is created on a target machine (local or remote)
- Steal & store access token in token vault for later impersonation

```
[17-06-2026 14:54:09][>>>>] help logon-monitor
Notify when a user logs on to a target system (async).

Usage: logon-monitor [--interval interval] [--user user] [--server server] [--steal-token]
Example: logon-monitor --server web01 --user Administrator

Optional arguments:
  --interval interval       INT         Polling interval in seconds (default: 5).
  --user user               STRING      Comma-separated list of target users (default: all users).
  --server server           STRING      Target system (default: local computer). Requires Administrator or Remote Desktop privileges on the remote computer.
  --steal-token             BOOL        Steal and store the access token of the logged on user. Requires NT AUTHORITY\SYSTEM privileges.
```

https://github.com/jakobfriedl/logon-monitor-bof


Home
DC01
WEB01
[Win11] CLIENT01
Kali
1 2 3 4
15:30
Conquest
Options Views
Sessions (Table)
AgentID IP (Internal) Username OS Hostna… Domain Process PID Last seen
63B9C7F2 192.168.168.20 CONQUEST\Administrator Windows Server WEB01 conquest.local monarch.http_x64. 5344 00:00:02 ago
Eventlog
[17-06-2026 15:28:37][+] User jakob
[17-06-2026 15:29:49][*] Agent 63B9C
Listeners
Start Listener
Generate Payload
ListenerID Listener Type Bind Address Bind Port Callback
CF41DAC7 HTTP 0.0.0.0 8080 192.168.168.129:8080
1F4F2CBB SMB - - \\.\pipe\smb-pipe

Situational Awareness

USB Device Monitoring

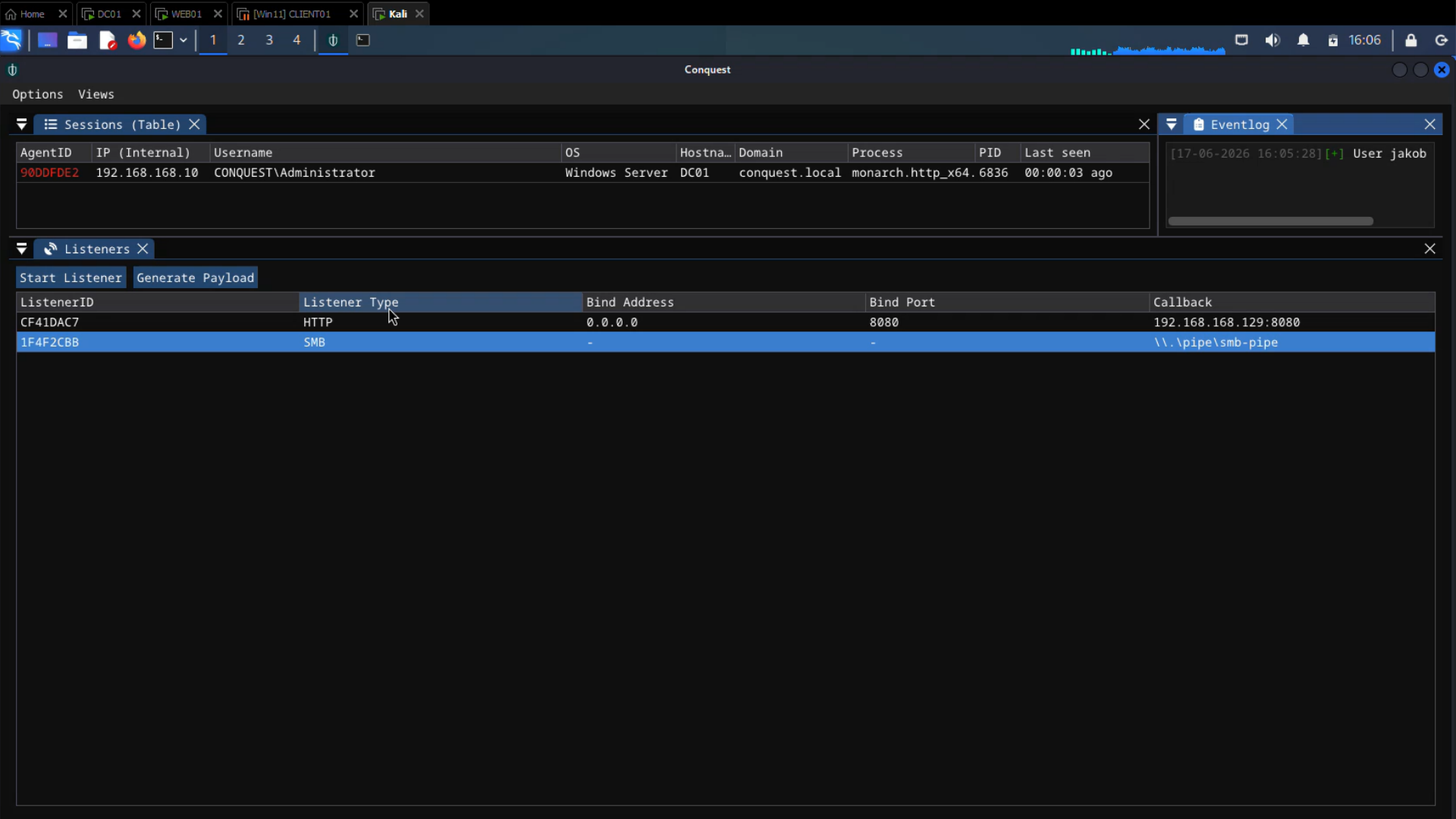
- Notify when USB device is connected/disconnected
- Automatically **list directory contents** of a connected drive or place coercion files to **force authentication** for NTLM relaying (even better with zero-click exploits)

```
[17-06-2026 14:55:44][>>>>] help usb-monitor
Notify when a USB device is connected/disconnected (async).
MITRE ATT&CK: T1120

Usage: usb-monitor [--list] [--put file] [--coerce] [--target target] [--name name] [--hidden]
Example: usb-monitor

Optional arguments:
  --list                  BOOL       List files and directories on the connected USB drive.
  --put file              STRING     Path to the file to upload to the connected USB drive.
  --coerce                BOOL       Upload a malicious .url file to force NTLM authentication to an attacker-controlled system (requires user interaction).
                                     Requires --target.
  --target target         STRING     Hostname or IP address of attacker-controlled system for coercion.
  --name name             STRING     Name of the uploaded file.
  --hidden                BOOL       Hide the uploaded file on the USB drive.
```

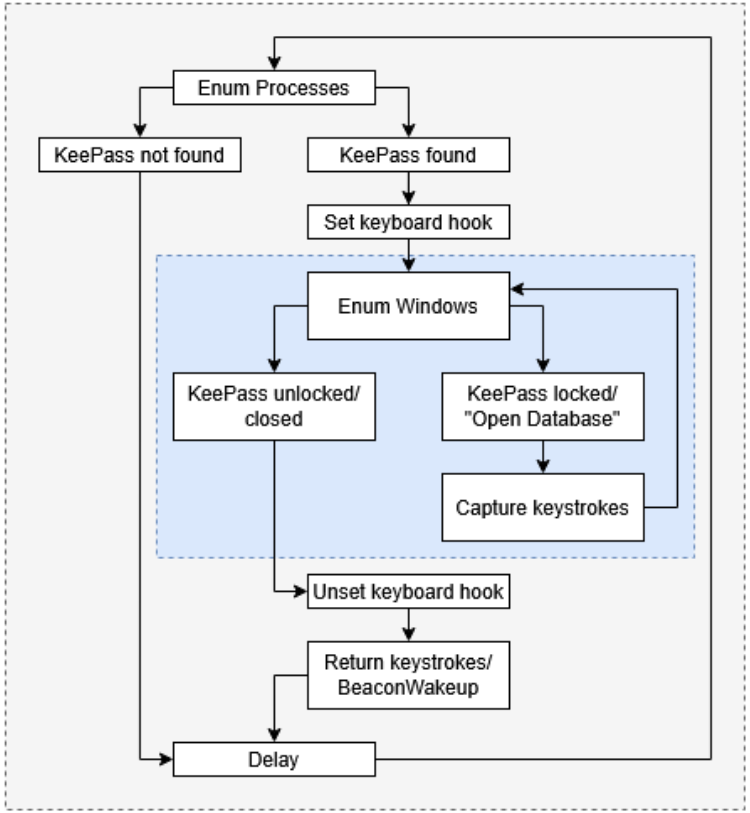
https://github.com/jakobfriedl/usb-monitor-bof


Home
DC01
WEB01
[Win11] CLIENT01
Kali
1 2 3 4
16:06
Conquest
Options Views
Sessions (Table)
AgentID IP (Internal) Username OS Hostna… Domain Process PID Last seen
90DDFDE2 192.168.168.10 CONQUEST\Administrator Windows Server DC01 conquest.local monarch.http_x64. 6836 00:00:03 ago
Eventlog
[17-06-2026 16:05:28][+] User jakob
Listeners
Start Listener
Generate Payload
ListenerID Listener Type Bind Address Bind Port Callback
CF41DAC7 HTTP 0.0.0.0 8080 192.168.168.129:8080
1F4F2CBB SMB - - \\.\pipe\smb-pipe

Credential Access

KeePass Keylogging

- Monitor processes & windows
- Locked KeePass databases have the same window title: “**Open Database – *.kdbx**”
- Output handler reconstructs Master Password



https://github.com/jakobfriedl/keelog-bof

Credential Access

Clipboard Monitoring

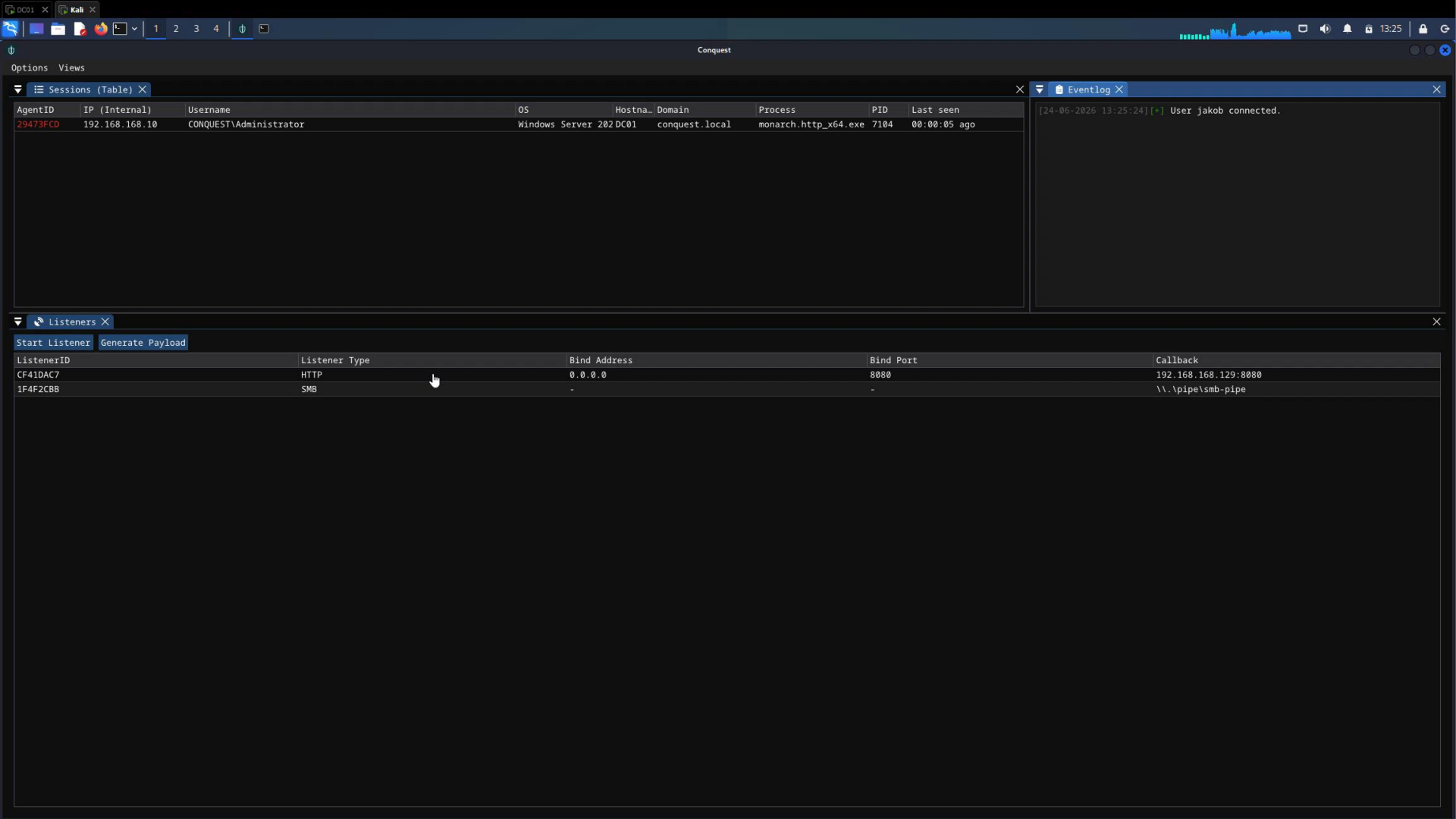
- Check `GetClipboardSequenceNumber` to determine if the clipboard content changed
- Limitation: Encoded screenshots are too large for the output pipe
- Solution: `get-clipboard` (synchronous)

```
[15-06-2026 20:33:40][>>>>] help clipboard-monitor
Monitor and output clipboard changes to the agent console (async).
MITRE ATT&CK: T1115

Usage: clipboard-monitor [--interval interval]
Example: clipboard-monitor --interval 2

Optional arguments:
  --interval interval       INT        Polling interval in seconds (default: 5).
```

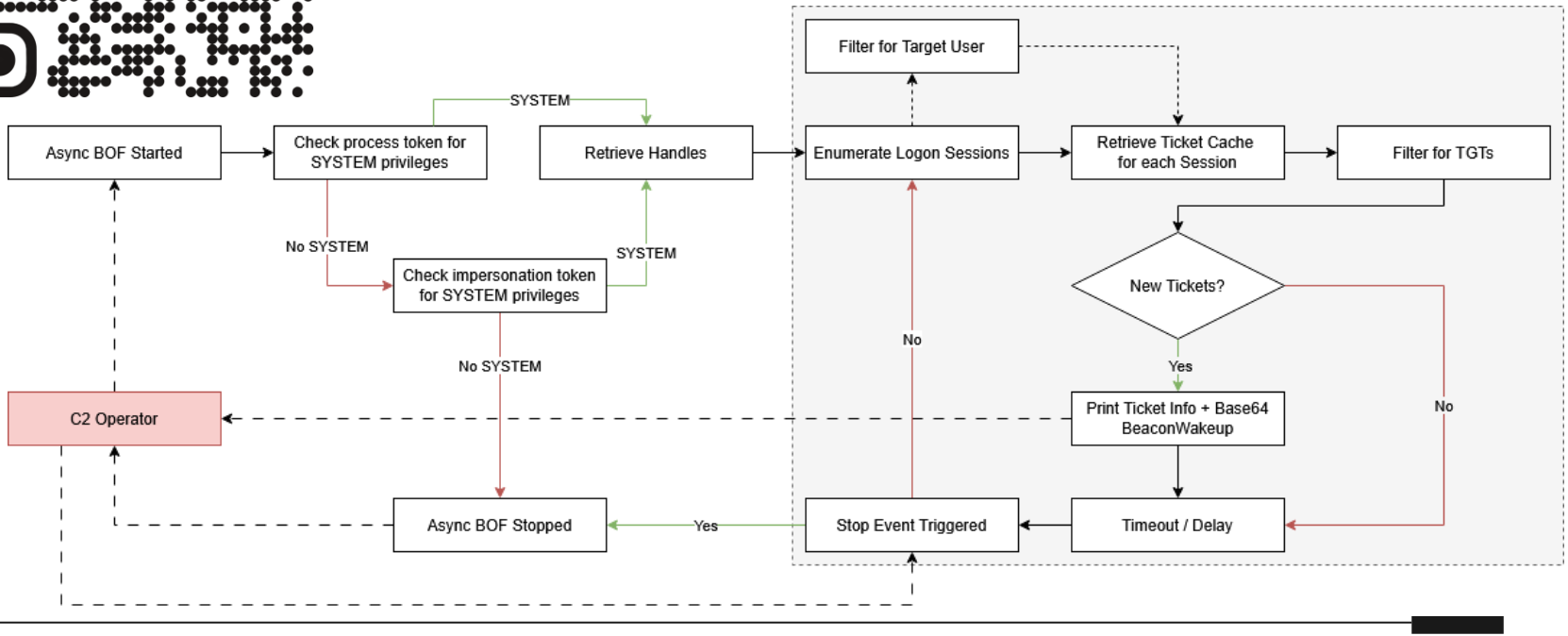
https://github.com/jakobfriedl/clipboard-monitor-bof


DC01
Kali
1 2 3 4
13:25
Conquest
Options Views
Sessions (Table)
AgentID IP (Internal) Username OS Hostna… Domain Process PID Last seen
29473FCD 192.168.168.10 CONQUEST\Administrator Windows Server 202 DC01 conquest.local monarch.http_x64.exe 7104 00:00:05 ago
Eventlog
[24-06-2026 13:25:24][+] User jakob connected.
Listeners
Start Listener
Generate Payload
ListenerID Listener Type Bind Address Bind Port Callback
CF41DAC7 HTTP 0.0.0.0 8080 192.168.168.129:8080
1F4F2CBB SMB - - \\.\pipe\smb-pipe

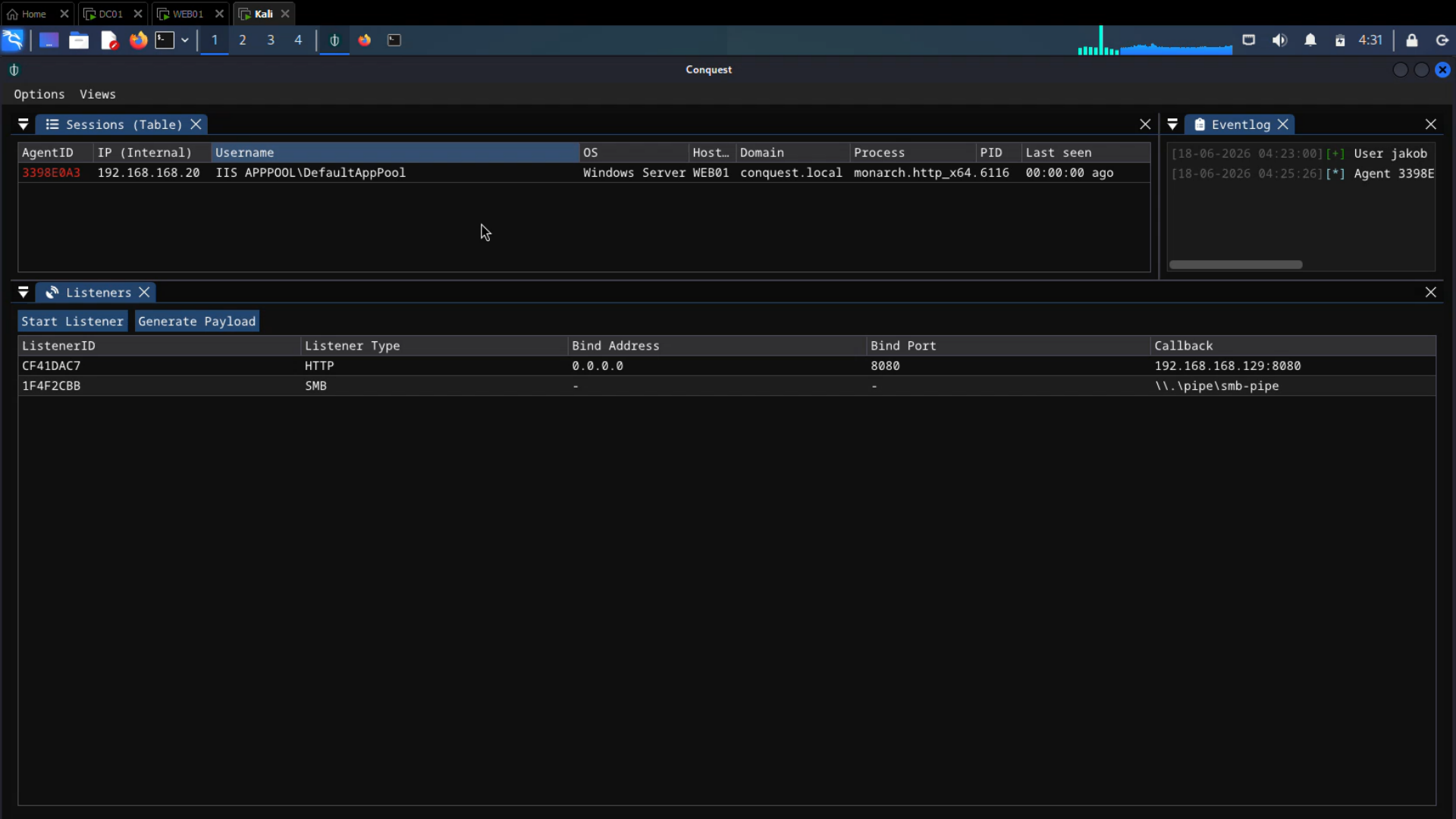
Credential Access + Lateral Movement

Kerberos TGT Monitoring

- `Rubeus monitor` functionality as a BOF (Unconstrained Delegation Abuse)



https://github.com/jakobfriedl/tgt-monitor-bof


Home
DC01
WEB01
Kali
1
2
3
4
4:31
Conquest
Options
Views
Sessions (Table)
AgentID
IP (Internal)
Username
OS
Host…
Domain
Process
PID
Last seen
3398E0A3
192.168.168.20
IIS APPPOOL\DefaultAppPool
Windows Server
WEB01
conquest.local
monarch.http_x64.6116
00:00:00 ago
Eventlog
[18-06-2026 04:23:00][+] User jakob
[18-06-2026 04:25:26][*] Agent 3398E
Listeners
Start Listener
Generate Payload
ListenerID
Listener Type
Bind Address
Bind Port
Callback
CF41DAC7
HTTP
0.0.0.0
8080
192.168.168.129:8080
1F4F2CBB
SMB
-
-
\\.\pipe\smb-pipe

Other Ideas

Event Monitoring	Long-running Tasks	OPSEC Risk Mitigation
- Automatic renewal of expiring TGTs (WIP) - Notify file/file share changes - Hijacking SSH sessions - NTLM hash capture (Responder/Inveigh as a BOF)	- LDAP enumeration (Bloodhound as a BOF) - Continuous screenshots - Audio/Microphone capture - Port forwarding/proxying	- Anti-debugging & anti-analysis - Self-delete on network isolation - Self-delete when debugger is detected - Monitor AV/EDR processes

Closing Thoughts

Key Takeaways

- BOFs are **lightweight third-party capabilities** that are executed within the C2 agent's process.
- The C2 agent contains less features, less code and **less detection surface**.
- Async BOFs enable background execution for **long-running tasks or monitoring** functionality
- Async BOFs operate completely independently from the C2 agent but can force them to wake up and return output.
- **The only limit is your imagination!**

Further Research/Next Steps

- **Automation!**
- Offensive SIEM
- Sleepmask for Async BOFs
- Async BOF loading without DLLs
- Position-independent code objects / PICOs (Crystal Palace + Tradecraft Garden)
- Async PICOs

Further Reading

- **Outflank Original Research**: https://www.outflank.nl/blog/2025/07/16/async-bofs-wake-me-up-before-you-go-go/
- **Nighthawk**: https://www.nighthawkc2.io/sivako/
- **Cobalt Strike**: https://www.cobaltstrike.com/blog/cobalt-strike-411-shh-beacon-is-sleeping
- **Havoc Professional**: https://infinitycurve.org/blog/release
- **BRC4**: https://bruteratel.com/release/2025/10/07/Release-Flux/
- https://kuwaitist.github.io/posts/ASYNC-BOFS/
- **Async PICOs**:
  - https://www.nccgroup.com/research/async-picos-and-custom-beacon-wakeups-in-cobalt-strike/
  - https://github.com/nccgroup/async-pico-hub
  - https://aff-wg.org/2026/06/10/a-long-running-bof-component-contract/
  - https://rastamouse.me/modular-pic-c2-agents/

Thanks!

Questions?

in/jakobfriedl

jakobfriedl

virtualloc

jakobfriedl.github.io

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon**, and infographics & images by **Freepik**