



# Alan

A powerful and professional  
Post-Exploitation Framework

<https://github.com/enkomio/AlanFramework/>

@s4tan - aparata@gmail.com



# Problem Statement

Red-Team operators need a tool that:

- Allows the operator to use his/her preferred tools
- It is reliable
- It is easy to use
- It has a low system footprint
- It has a good operational security
- Can be extended in an easy way (no C/ASM development required)
- It is affordable
- Can help to simulate real-world threats to test the controls in place

# Alan - Next Generation Post-Exploitation Framework

```

  ALAN
  ==[ Post Exploitation Framework ]==
  Copyright (c) 2021-2022 Enkomio

[INFO] 2022-02-20 18:08:49 - Start listeners
[INFO] 2022-02-20 18:08:49 - Web listener started on: 0.0.0.0:8080
[INFO] 2022-02-20 18:08:49 - Using certificate: E=alan@localhost, C=Italy, S=IT, L=IT, O=AlanCA, OU=AlanFramework, CN=Enkomio. Expires: 5/1/2022 7:56:19 PM
[INFO] 2022-02-20 18:08:49 - Web listener started on: 0.0.0.0:8443
[INFO] 2022-02-20 18:08:49 - Host address: 192.168.56.1
[INFO] 2022-02-20 18:08:49 - Host address: 192.168.174.1
[INFO] 2022-02-20 18:08:49 - Host address: 192.168.1.61
[INFO] 2022-02-20 18:08:49 - Host address: 192.168.88.17
$> join
17160@http://127.0.0.1> ?

[+] Help:
? or help          Show this help.
agents             List the currently active agents.
exec <cmd> [&]      Execute the command on the remote host (& run the process in background).
shell [<cmd>] [&]   Execute the shell command on the remote host.
                   If no command is specified, a command shell is started
                   on the remote host (& run the process in background).
                   In memory execution of a local binary. If a <pid> is
                   specific the file is injected into that process, otherwise
                   a default one is chosen. & run the process in background.

run <cmd> [<pid>] [<x86|x64>] [&]
                   Terminate the specified process.
                   Get information on the host system.
                   Get extended information on the host system.
                   Download the agent config to the specified file.
                   Detach from the agent session without terminating the agent.
                   Show a list of the current running processes.
                   Select the specified agent as the currently active one.
                   Send a new configuration to the agent.
                   Migrate the agent session to the specified process ID.
                   Locally download the file(s) from the agent host.
                   Upload a local file(s) to the agent host.
                   Set the agent sleep timeout. A variance integer can be specified.
                   Termination the agent process.

kill <pid>          Terminate the specified process.
info               Get information on the host system.
info++            Get extended information on the host system.
get-config         Download the agent config to the specified file.
detach            Detach from the agent session without terminating the agent.
ps                Show a list of the current running processes.
join <agent ID>    Select the specified agent as the currently active one.
update <config file> Send a new configuration to the agent.
migrate <process ID> <x86|x64> Migrate the agent session to the specified process ID.
download <remote> [<local>]   Locally download the file(s) from the agent host.
upload <local> <remote>      Upload a local file(s) to the agent host.
sleep <msec> [<variance>]    Set the agent sleep timeout. A variance integer can be specified.
exit              Termination the agent process.

17160@http://127.0.0.1>
```

Download: <https://github.com/enkomio/AlanFramework/>



# Alan - Features (1/2)

Currently supported features:

- Various kind of artefact formats are supported, such as: Executable, DLL, Shellcode, PowerShell (all types are provided in both x86 and x64 version).
- Execution of commands received from a Command-and-control server via HTTP/HTTPS (the certificate is automatically generated and can be customized).
- All traffic is encrypted in a strong way. A network dump or the reverse engineering of the binary is not enough to the decrypt the traffic.
- The Alan server can be executed on any OS supporting .NET core (such as Linux).
- Low network footprint and AV resistant.
- All code executed in memory (event third party programs).
- SOCKS5 proxy for Network Pivoting



## Alan - Features (2/2)

Currently supported features:

- Execution of JavaScript script file to extend the Alan agent capabilities.
- The agent can be easily configure through JSON profiles.
- Fully customizable (eg. it is possible to change the used communication protocol at runtime).
- No dependency on third-party tools/software (DB, Web Server, TLS Certificate generation, ...).
- Server can be customized to mimic a legitimate one.
- The operator sends command to the Alan agent by using a clean Command-line-interface.
- Fully documented.



# Roadmap 2022

- **Web UI for the Server (Only available in Alan Pro Edition)**
- Execution of PowerShell script
- New C2 channel: DNS
- Create JavaScript files Knowledge Base to emulate a real-word adversary
- Additional commands (download from web url, agent customization, ...)



# Alan Early-Adopters

Who is an Alan early-adopter?

- Perform red-team activities
- Use post-exploitation tools but want more from them
- Willing to share TTPs used during the red-team activities



# Alan Early-Adopters

What they receive:

- Alan agent source-code
- Early access to the new Alan version (included the Pro version when ready)
- Strict collaboration with the development team for the suggestion of new features
- “Real-time” support
- Sync-meeting (one meeting each one or two months)





# Alan Early-Adopters

What they provide:

- Feedbacks
- Spread the word
- Support Alan development with a yearly contribute of 2k € (+tax). The support includes no limitation on its usage or number of users.



# Become an Alan Early-Adopter

How to become an Alan early-adopter?

1. Send an email to [aparata@gmail.com](mailto:aparata@gmail.com) specifying why you are interested in becoming an Alan early-adopter and your company name.
2. Receive the Alan binary in its latest version (this might include a release not yet published).
3. Schedule an introduction meeting.
4. Test Alan for 2 weeks.
5. Schedule a feedback meeting.
6. Become an effective Alan early-adopter by sending your contribute.
7. Receive the Alan agent source code.