FORTRA OUTFLANK

# Lessons learned from 10 years of red teaming

*+ a sneak peek into Outflank Security Tooling*

**Marc Smeets**
**Red Team manager @ Outflank**

**Ekoparty, Nov 2024**

FORTRA

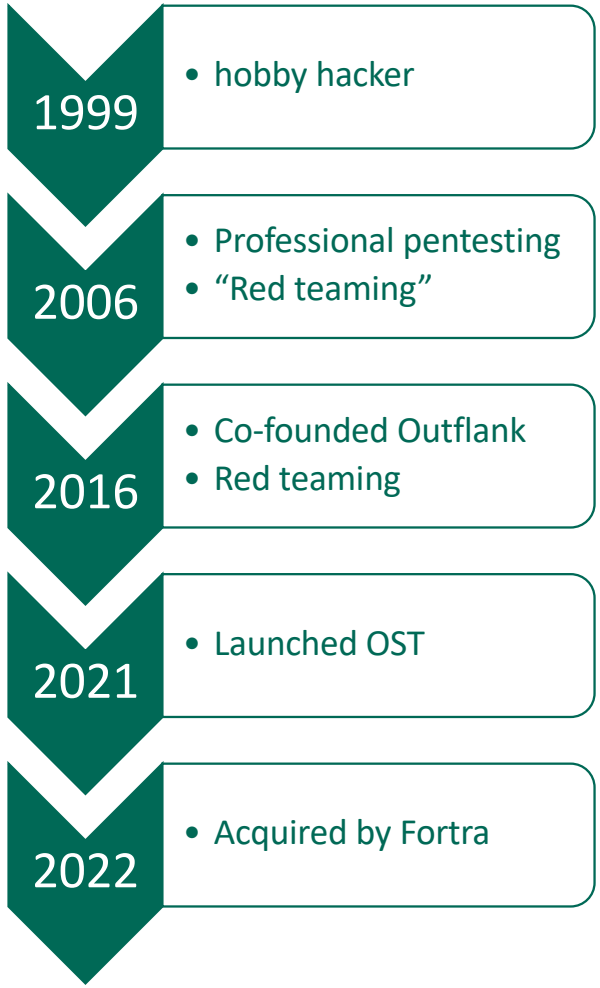
# About Me & Fortra's Outflank

## Marc Smeets - @MarcOverIP

- Manager in the Outflank team
- Amsterdam, The Netherlands based
- Free time: drift car instructor

## Outflank

- Red Teaming done right: services and tooling
- Makers of 'Outflank Security Tooling'
- Public tools and blogs via:
  - https://outflank.nl/blog
  - https://github.com/OutflankNL
- Part of *Fortra* since 2022 – together with *Core Impact* and *Cobalt Strike*



FORTRA

# Let’s talk red teaming

FORTRA

# Red Teaming

- Attack simulation to improve resilience of the organization and the blue team
- Mimic tactics, techniques and procedures of advanced real-life attackers at the highest level
- Value for the customer
  - Exceptional learning experience – be prepared for real incidents
  - Irrefutable proof of the state of IT security within the organization
- More than pentesting++

FORTRA

# Selection of experience from 50+ red team gigs

Buy me a beer and I'll tell you:

- Hacked a 'unhackable' nuclear power plant
- When €2 million went missing
- How we know blue teams are investigate us before they know they are
- Have bypass knowledge of all top EDR products
- When escalation went wrong, and we nearly ended up on Europol's cyber actor list
- "just 4 hackers at their kitchen table completely hacked us after we spent many million euros. We need to change."

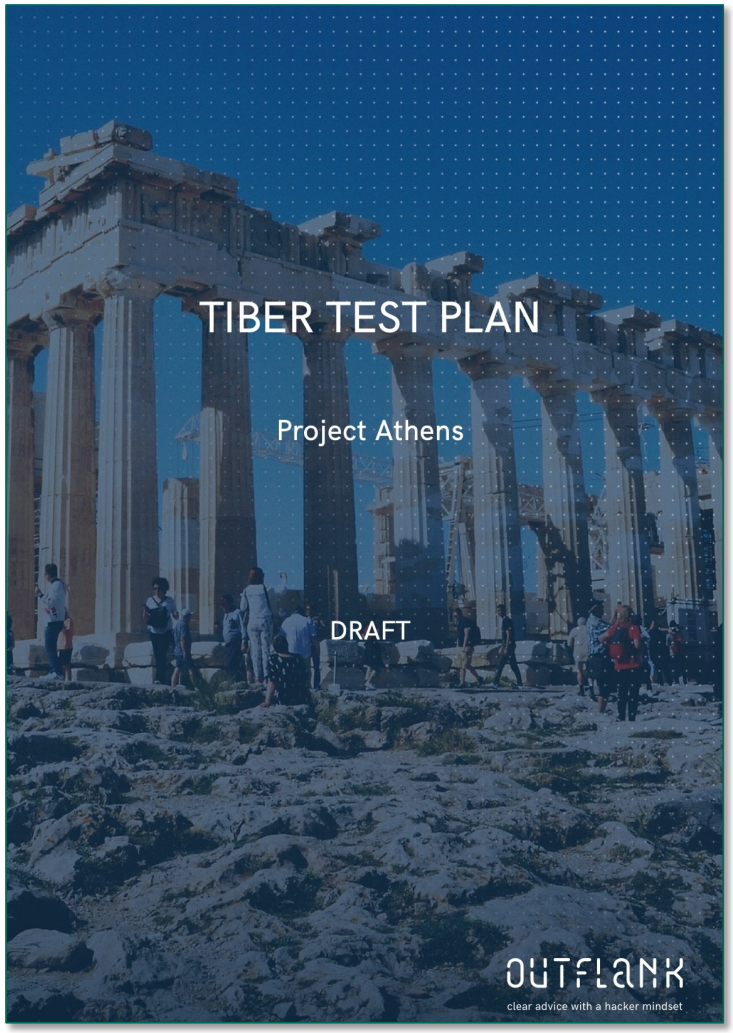
FORTRA

# Lessons learned

FORTRA

# Lesson 1 – a framework is important

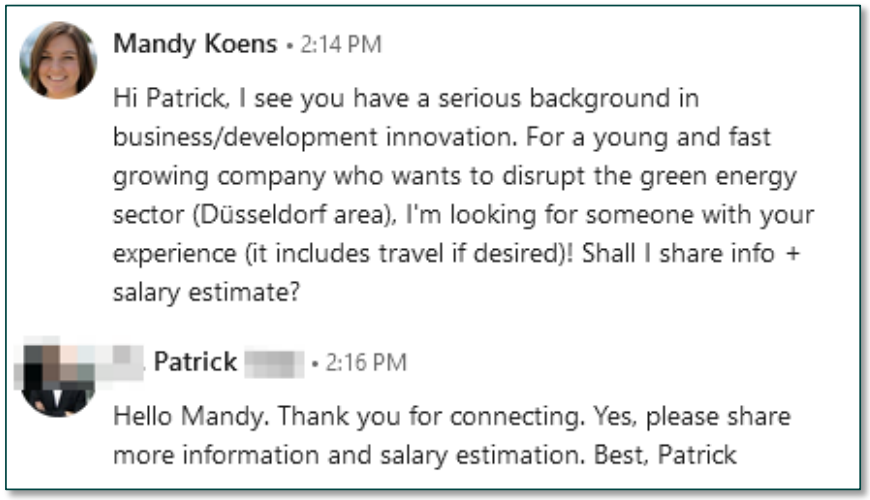
- Defines what ‘red teaming’ is
- **TIBER**: *T*hreat *I*ntelligence *B*ased *E*thical *R*ed teaming
  - Or CBEST, AASE, etc
- Defines:
  - Testing organization, communication, blue/red/white team
  - Realistic testing scenarios, milestones and planning
  - Incident lifecycle: appetite and escalation paths
- Overall: gives clarity to all parties involved



FORTRA

## Lesson 2 – networks are like coconuts

- Getting your payload to land and execute can be hard.
  But once inside, internal networks are most often wide open
- Hacking websites and DMZ systems is not realistic.
- Spear fishing still gets the job done
- Cloud is a new frontier



FORTRA

# Lesson 3 – Evasion has become hard in the current EDR time

## 2010

A trivial malware sample created works for years against many AVs

- Malware executables on disk
- Exploits era:
  - ActiveX
  - Flash
  - Java

## 2016

One complex sample created works for years against many AVs

- Staging and fileless malware
- In-memory process creation / migration techniques
- Office macro's
- Living off the land

## 2022

Evasion of EDRs requires unique complex examples, custom TTP's

- (In)direct systemcalls
- Unhooking EDRs
- Custom sleep masks
- Thread stack spoofing
- Blending in the environment

**Making tailor-made malware to bypass EDRs has become a dedicated specialism. Bad guys have a solution. We had to find a solution for this as well.**

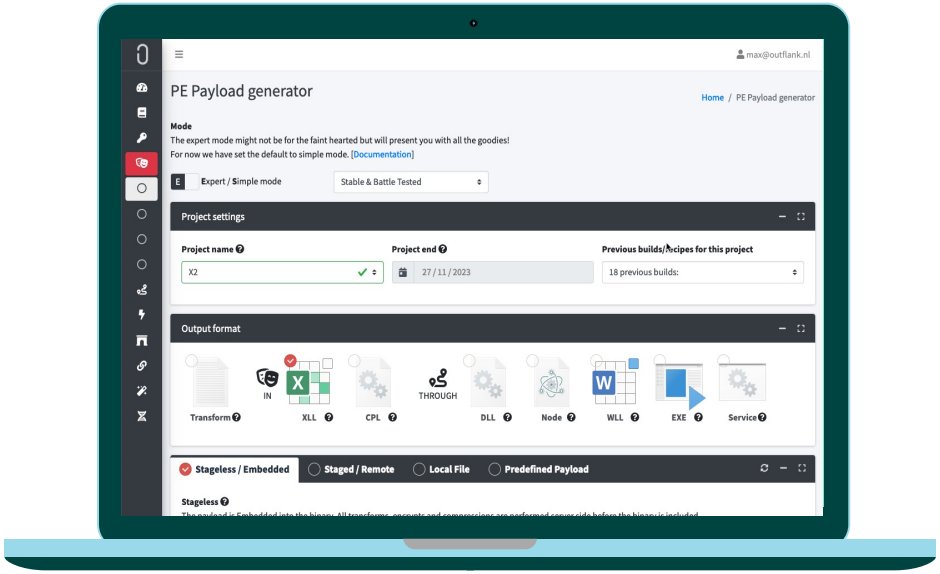
FORTRA

# Outflank Security Tooling (OST)

Tools and tradecraft straight from the Outflank red team

OST is not a C2 product but a **collection of offensive tools and tradecraft**, offering:

- A broad arsenal
- OPSEC safety
- Evasion of security controls
- Intuitive interface
- Knowledge hub and active community
- Constantly evolving and cloud delivery model



FORTRA

# Case Study

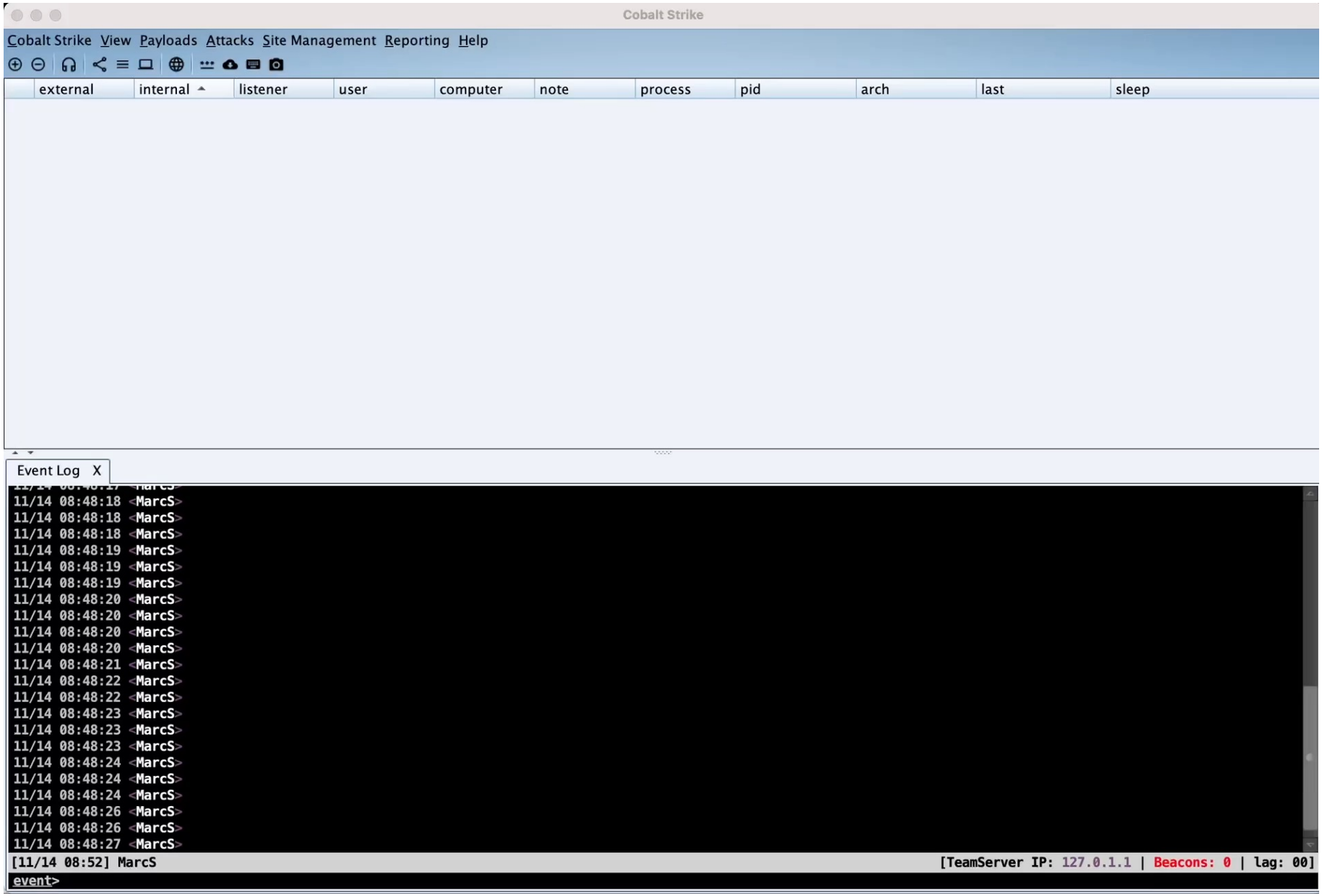
## Building an advanced Cobalt Strike payload into a phishing payload

FORTRA

# Building advanced malware with OST

Make your Cobalt Strike payload more OPSEC safe

1. Create a raw Cobalt Strike payload from your team server


Cobalt Strike
Cobalt Strike View Payloads Attacks Site Management Reporting Help
external internal listener user computer note process pid arch last sleep
Event Log X
11/14 08:48:18 <MarcS>
11/14 08:48:18 <MarcS>
11/14 08:48:18 <MarcS>
11/14 08:48:19 <MarcS>
11/14 08:48:19 <MarcS>
11/14 08:48:19 <MarcS>
11/14 08:48:20 <MarcS>
11/14 08:48:20 <MarcS>
11/14 08:48:20 <MarcS>
11/14 08:48:20 <MarcS>
11/14 08:48:21 <MarcS>
11/14 08:48:22 <MarcS>
11/14 08:48:22 <MarcS>
11/14 08:48:23 <MarcS>
11/14 08:48:23 <MarcS>
11/14 08:48:23 <MarcS>
11/14 08:48:24 <MarcS>
11/14 08:48:24 <MarcS>
11/14 08:48:24 <MarcS>
11/14 08:48:26 <MarcS>
11/14 08:48:26 <MarcS>
11/14 08:48:27 <MarcS>
[11/14 08:52] MarcS
[TeamServer IP: 127.0.1.1 | Beacons: 0 | lag: 00]
event>

FORTRA

# Building advanced malware with OST

Make your Cobalt Strike payload more OPSEC safe

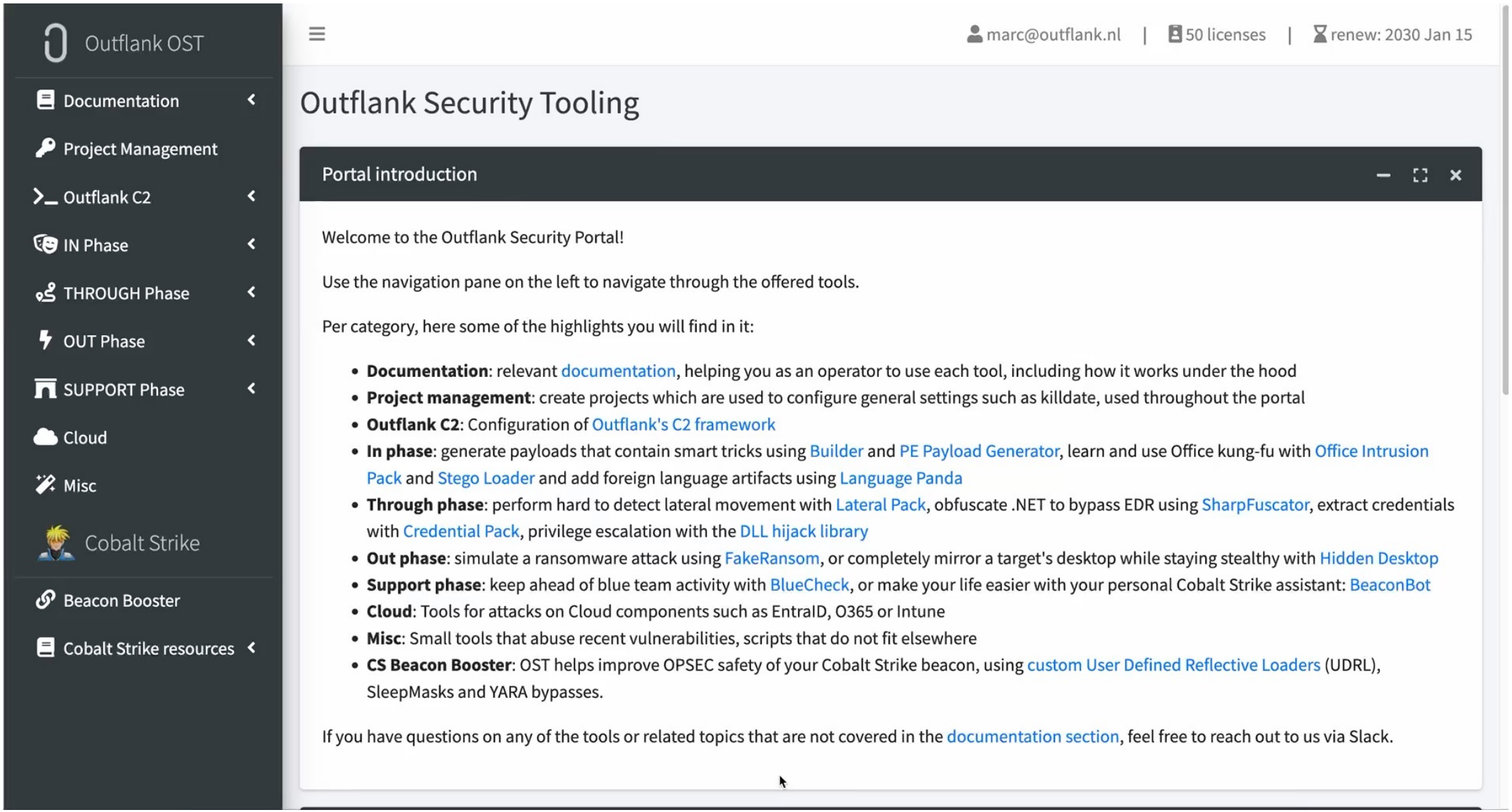
1. Create a raw Cobalt Strike payload from your team server
2. Go to the *Beacon Booster* in the OST portal

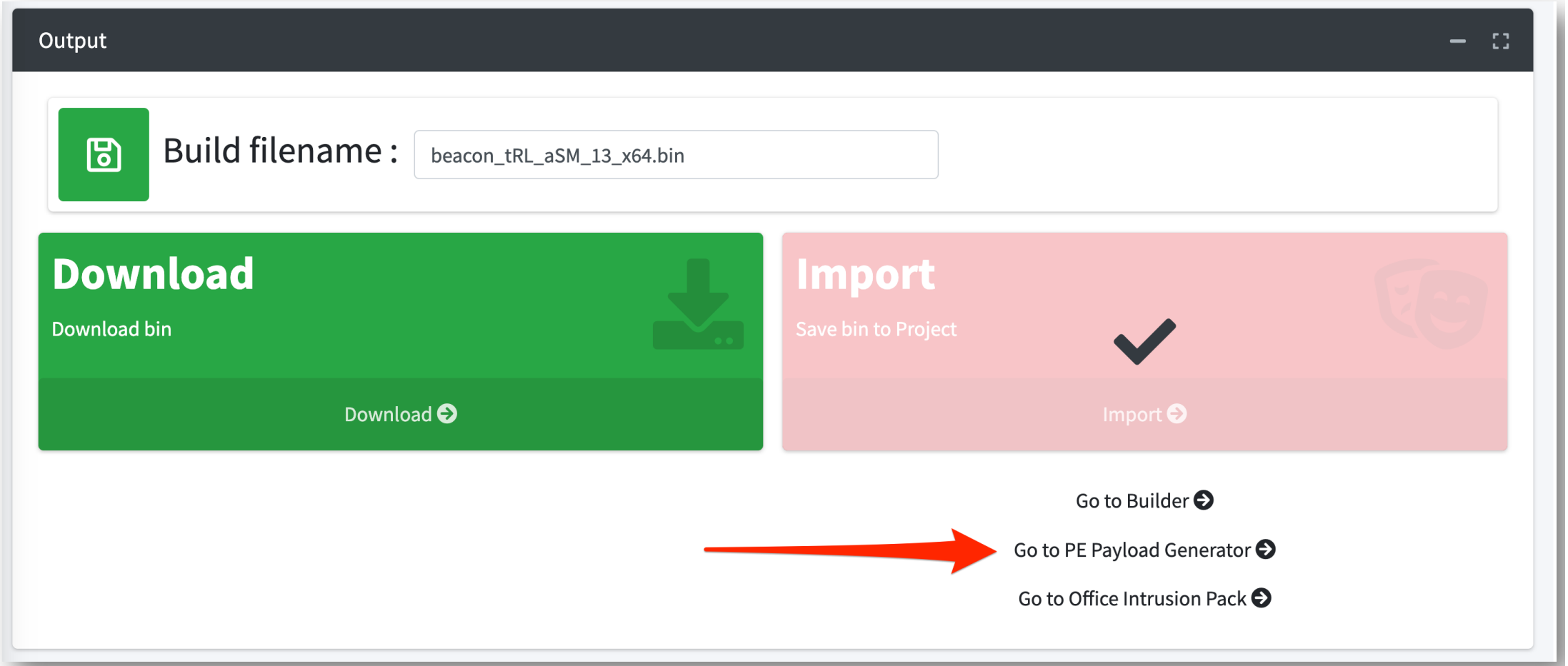
FORTRA

# Building advanced malware with OST

Make your Cobalt Strike payload more OPSEC safe

1. Create a raw Cobalt Strike payload from your team server
2. Go to the *Beacon Booster* in the OST portal
3. Upload the raw Cobalt Strike payload to OST, select OPSEC options and build


Outflank OST
Documentation
Project Management
Outflank C2
IN Phase
THROUGH Phase
OUT Phase
SUPPORT Phase
Cloud
Misc
Cobalt Strike
Beacon Booster
Cobalt Strike resources
marc@outflank.nl | 50 licenses | renew: 2030 Jan 15
Outflank Security Tooling
Portal introduction
Welcome to the Outflank Security Portal!
Use the navigation pane on the left to navigate through the offered tools.
Per category, here some of the highlights you will find in it:
Documentation: relevant documentation, helping you as an operator to use each tool, including how it works under the hood
Project management: create projects which are used to configure general settings such as killdate, used throughout the portal
Outflank C2: Configuration of Outflank's C2 framework
In phase: generate payloads that contain smart tricks using Builder and PE Payload Generator, learn and use Office kung-fu with Office Intrusion Pack and Stego Loader and add foreign language artifacts using Language Panda
Through phase: perform hard to detect lateral movement with Lateral Pack, obfuscate .NET to bypass EDR using SharpFuscator, extract credentials with Credential Pack, privilege escalation with the DLL hijack library
Out phase: simulate a ransomware attack using FakeRansom, or completely mirror a target's desktop while staying stealthy with Hidden Desktop
Support phase: keep ahead of blue team activity with BlueCheck, or make your life easier with your personal Cobalt Strike assistant: BeaconBot
Cloud: Tools for attacks on Cloud components such as EntraID, O365 or Intune
Misc: Small tools that abuse recent vulnerabilities, scripts that do not fit elsewhere
CS Beacon Booster: OST helps improve OPSEC safety of your Cobalt Strike beacon, using custom User Defined Reflective Loaders (UDRL), SleepMasks and YARA bypasses.
If you have questions on any of the tools or related topics that are not covered in the documentation section, feel free to reach out to us via Slack.


Output
Build filename :
beacon_tRL_aSM_13_x64.bin
Download
Download bin
Download
Import
Save bin to Project
Import
Go to Builder
Go to PE Payload Generator
Go to Office Intrusion Pack

FORTRA

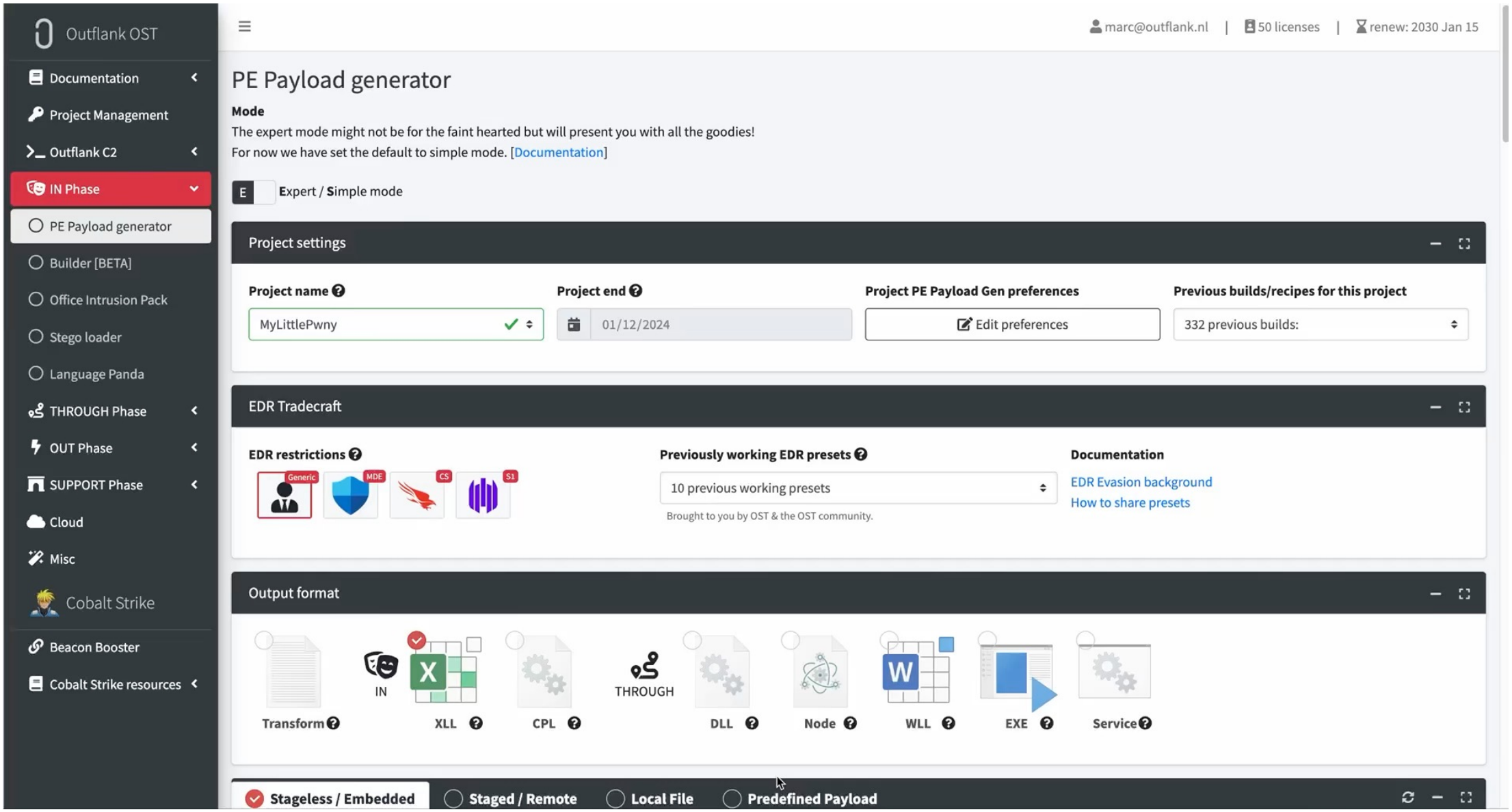
# Building advanced malware with OST

Make your Cobalt Strike payload more OPSEC safe

1. Create a raw Cobalt Strike payload from your team server
2. Go to the *Beacon Booster* in the OST portal
3. Upload the raw Cobalt Strike payload to OST, select options and build

Weaponize the ‘boosted’ Cobalt Strike payload

1. Using *PE Payload Generator* for compiling an actual phishing payload


Outflank OST
Documentation
Project Management
Outflank C2
IN Phase
PE Payload generator
Builder [BETA]
Office Intrusion Pack
Stego loader
Language Panda
THROUGH Phase
OUT Phase
SUPPORT Phase
Cloud
Misc
Cobalt Strike
Beacon Booster
Cobalt Strike resources
marc@outflank.nl | 50 licenses | renew: 2030 Jan 15
PE Payload generator
Mode
The expert mode might not be for the faint hearted but will present you with all the goodies!
For now we have set the default to simple mode. [Documentation]
E Expert / Simple mode
Project settings
Project name
MyLittlePwny
Project end
01/12/2024
Project PE Payload Gen preferences
Edit preferences
Previous builds/recipes for this project
332 previous builds:
EDR Tradecraft
EDR restrictions
Generic
MDE
CS
S1
Previously working EDR presets
10 previous working presets
Brought to you by OST & the OST community.
Documentation
EDR Evasion background
How to share presets
Output format
Transform
IN
XLL
CPL
THROUGH
DLL
Node
WLL
EXE
Service
Stageless / Embedded
Staged / Remote
Local File
Predefined Payload

**CompanyName**

Microsoft Corporation

**LegalCopyright**

© Microsoft Corporation. All rights reserved.

**FileVersion**

10.0.22621.1 (WinBuild.160101.0800)

**ProductVersion**

10.0.22621.1

**Build**

Output

| Build | Filename | | Size | Download |
|---|---|---|---|---|
| 1731506859 | build.md | | 5 kb | Download |
| 1731506859 | ioc.txt | | 464 b | Download |
| 1731506859 | k20241201_MyLittlePwny_beacon_tRL_aSM_12_x64.bin_Stage0x64_feda2e0088870048b1ddbba6d3812f13_3PdnZI.cpl | x64 | 589 kb | Download |
| 1731506859 | readme.md | | 1 kb | Download |

EDR preset share form

Download Zip

FORTRA

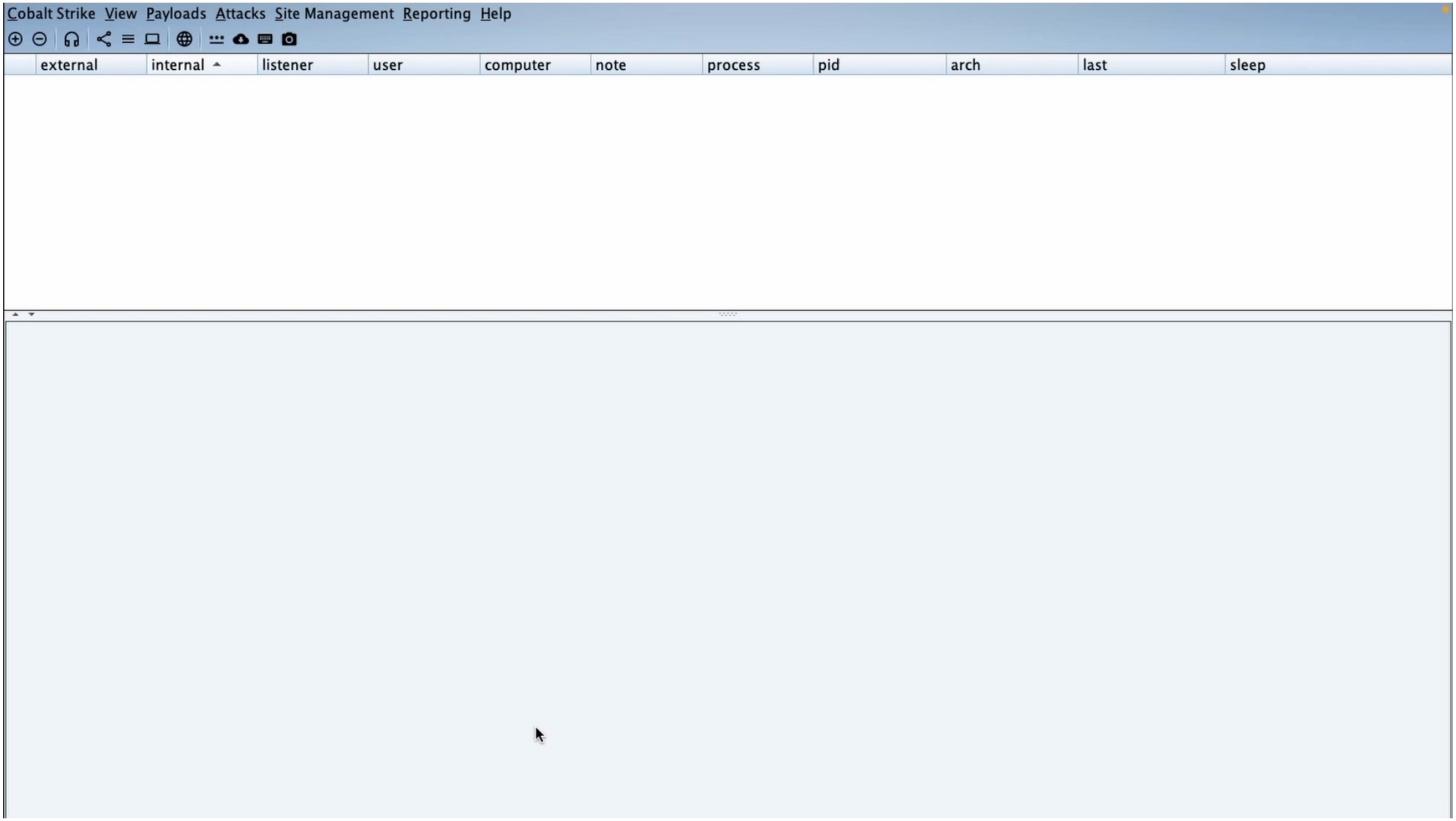
# Building advanced malware with OST

Make your Cobalt Strike payload more OPSEC safe

1. Create a raw Cobalt Strike payload from your team server
2. Go to the *Beacon Booster* in the OST portal
3. Upload the raw Cobalt Strike payload to OST, select options and build

Weaponize the ‘boosted’ Cobalt Strike payload

1. Using *PE Payload Generator* for compiling an actual phishing payload
2. Executing the phishing payload

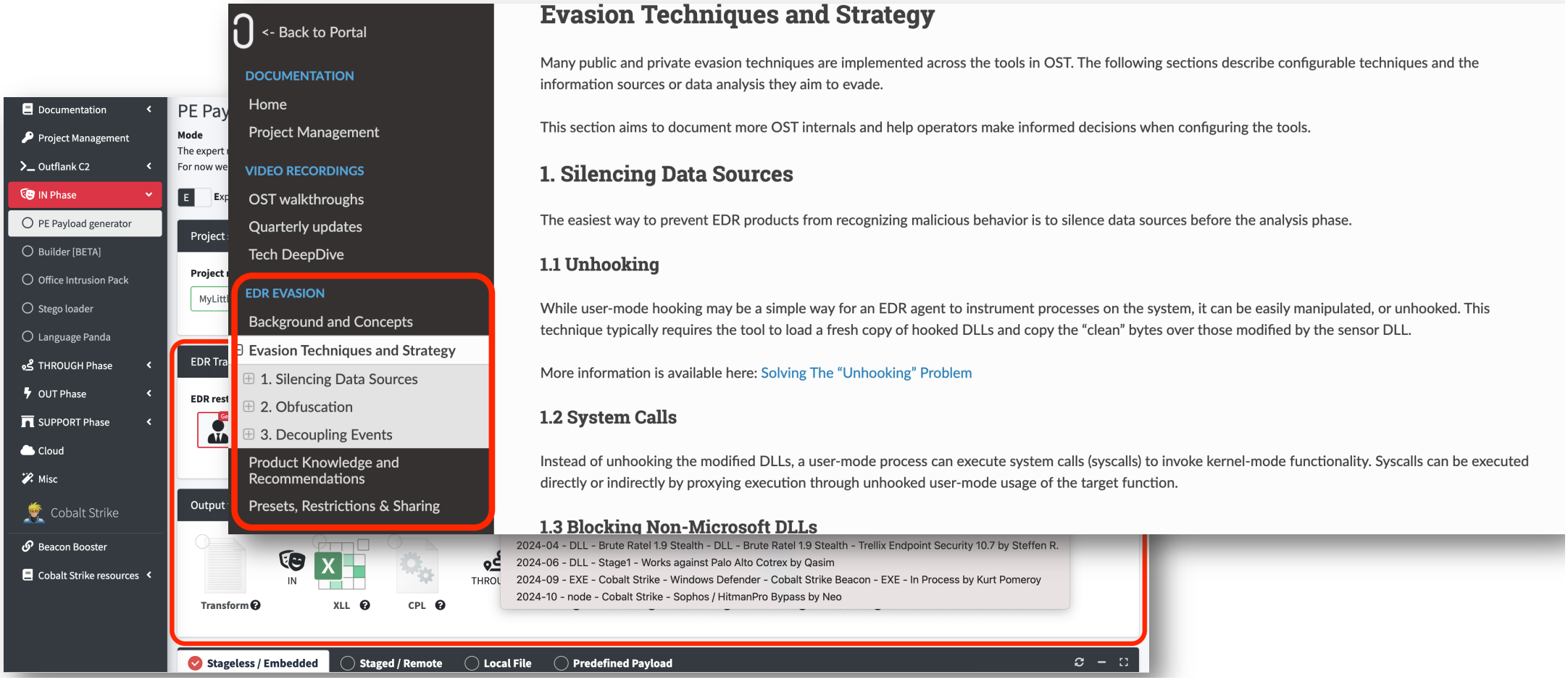

Cobalt Strike View Payloads Attacks Site Management Reporting Help
external
internal
listener
user
computer
note
process
pid
arch
last
sleep

FORTRA

# What’s More

FORTRA

# EDR specific knowledge




## Evasion Techniques and Strategy

Many public and private evasion techniques are implemented across the tools in OST. The following sections describe configurable techniques and the information sources or data analysis they aim to evade.

This section aims to document more OST internals and help operators make informed decisions when configuring the tools.

### 1. Silencing Data Sources

The easiest way to prevent EDR products from recognizing malicious behavior is to silence data sources before the analysis phase.

#### 1.1 Unhooking

While user-mode hooking may be a simple way for an EDR agent to instrument processes on the system, it can be easily manipulated, or unhooked. This technique typically requires the tool to load a fresh copy of hooked DLLs and copy the “clean” bytes over those modified by the sensor DLL.

More information is available here: Solving The “Unhooking” Problem

#### 1.2 System Calls

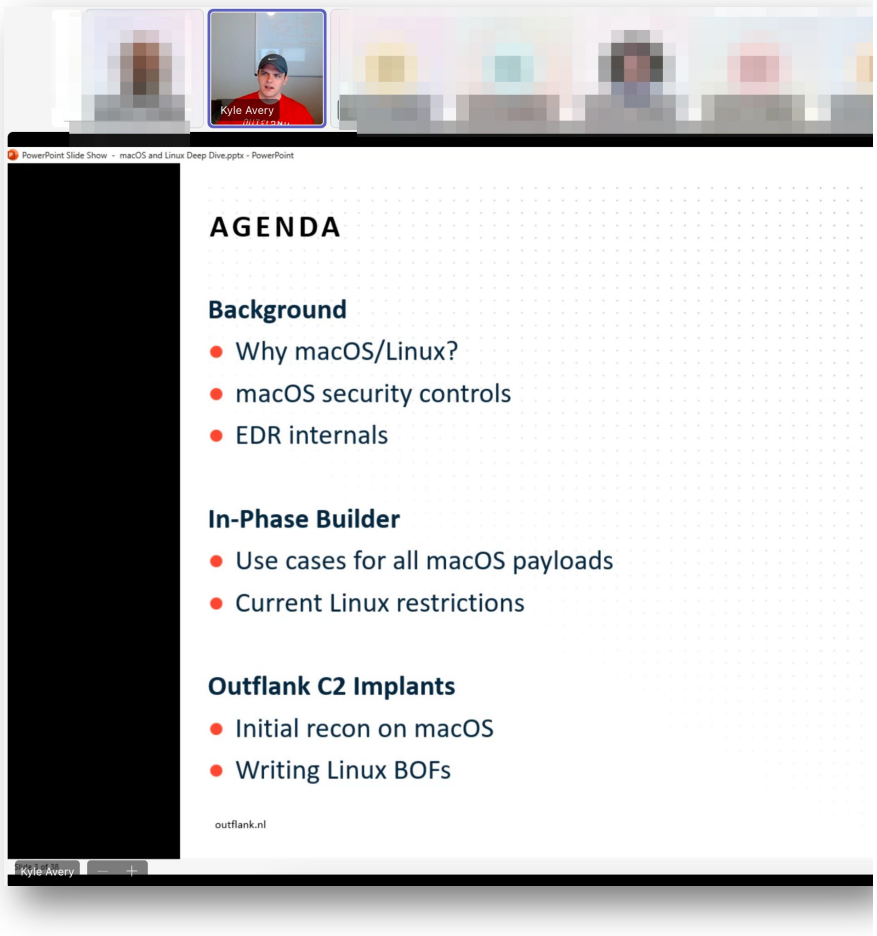
Instead of unhooking the modified DLLs, a user-mode process can execute system calls (syscalls) to invoke kernel-mode functionality. Syscalls can be executed directly or indirectly by proxying execution through unhooked user-mode usage of the target function.

#### 1.3 Blocking Non-Microsoft DLLs

FORTRA

# Collection of Evasive Tools and Tradecraft

- **Tools – 34 and increasing**
  - Outflank C2
  - CredentialPack
  - LateralPack
  - KernelTool
  - HiddenDesktop
  - …
- **Tradecraft**
  - Tech deep dive sessions
  - Extensive documentation
- **Strong development**
  - ~ Bi-weekly releases
  - 11 Outflank + external developers



FORTRA

# SUMMARY

- Red teaming framework needed
- Networks are like coconuts
- Modern red teams need a tailor-made malware toolset.
  Outflank Security Tooling can help.

FORTRA OUTFLANK

# Thank You

**Marc Smeets**
@MarcOverIP
marc@outflank.nl

www.outflank.nl/ost