FORTRA™

Outflank OST Modules Technical Overview

December 2024

Outflank Security Tooling (OST) is a broad set of evasive tools that cover every step in the attacker kill chain to effectively emulate real-world attack scenarios. OST enables red teams to bypass advanced defensive measures and assess organizational resilience. It not only simplifies complex tasks, but also fosters a vibrant community for sharing and discussing the latest tradecraft, ensuring red teams remain at the forefront of offensive security.

While this document focuses on the various *tools* available in Outflank OST (33 at this moment), two other important areas of OST are:

- *tradecraft knowledge*: over 12 hours of video on Tech Deep Dives as well as over 13k lines of documentation on tools, EDR evasion strategies and technique and red teaming tactics.
- the ever-growing *Outflank community*: a vetted community with nearing 1000 red teamers in the private Slack channel where they can safely interact with other operators and the Outflank team. It serves as a hub that facilitates collaboration, shared learning, and knowledge exchange with fellow red teamers.

The Outflank team regularly presents at various industry events and has provided trainings to the community – most recently an MS Office Tradecraft session with over 1,000 registered attendees. A selection of their public research is on the Outflank Blog and on the Presentation repository.

The Outflank tools allow you to simulate similar techniques to what some APTs and Organized Crime Groups apply but are not available in public tools. They also help your team members to easily perform deep technical and difficult tasks without hassle, while being OPSEC safe. OST is explicitly developed to enable skilled operators to bypass defensive measures and detection tools.

The Outflank tools are presented to customers in their private web download portal. From there an easy interface guides users in generating payloads and downloading tools to be used within their own infrastructure.

More detailed information can be found below, but also on https://outflank.nl/ost

The tools of OST are categorized along the phases of a typical attack kill chain:

- In Phase (initial access)
- Through Phase (C2, lateral movement, privilege escalation)
- Out Phase (actions on objectives)

In addition, other categories of tools are provided:

- EDR Evasion
- Support (scripts and tools to support the red teaming process)
- Cloud (ROADtune, PhisherPrice, O365TokenExtractor)
- Cobalt Strike Enhancer: Beacon Booster
- Miscellaneous (various smaller BOFs and attack tools)

In the remainder of this document, a high-level technical overview of the tools in OST is provided.

EDR Evasion

Background and Concepts

Endpoint Detection and Response (EDR) products typically consist of a centralized server and a collection of endpoint agents. Agents comprise multiple sensory components, including a DLL loaded into each process, a kernel driver, or a startup service for uploading telemetry and directing the agent. Each agent collects information from the following telemetry sources and acts based on local logic or as instructed by the server.

Techniques that bypass all EDRs in one go are now increasingly less common. It is becoming more important for red teamers to know which exact technique to use (or not use) for a specific EDR. Up-to-date knowledge is important—particularly knowledge on how a specific EDR gets its data, how it works under the hood and where its blind spots and bypass opportunities lie. For OST we aimed on better equipping our customers with exactly that: knowledge. We started with what we call “EDR Presets” in our Payload Generator. Payload Generator allows red team operators to

easily create payloads with a wide range of decoy tricks, binary transformation options (such as prepending random shellcode), payload transformation options (such as encoding), encryption and compression, encryption possibilities, process creation techniques (such as Write Hook, KernelCallbackTable, Earlybird, in-process thread), as well as OPSEC tricks, guardrails and many more. The total set of option combinations easily surpasses a hundred, and it is critical to select the proper combination to bypass specific EDRs.

EDR Presets are configuration sets of these 100+ settings in the Payload Generator. Each EDR Preset has demonstrated the ability to evade detection by a certain EDR system at specific points in time. Though presets are incredibly useful and are a significant time saver for red team operators, they do have a limited shelf life. In order to maintain an up-to-date library of effective presets, we obviously perform ongoing lab testing. But as OST has a large community of active users, we decided to use that power of the community. We have implemented a secure way where OST users can contribute such presets they discovered during their engagements. This collaborative effort highlights the benefit of being a part of a strong user community, providing access to a collective pool of knowledge from which everyone can benefit.

In Phase

PE Payload Generator

Payload generator is a binary payload builder/transformer focused on OPSEC safety, operational traceability, and anti-forensics. It can generate various output formats useful for many red teaming use cases, including:

- Phishing
- Dropper
- Persistency
- Privilege Escalation
- Lateral movement
- Legup provider

- UAC bypass
- AppLocker bypass

The features provided should work on arbitrary payloads but are tested on Cobalt Strike and Stage1 payloads. The build process is always performed using OPSEC safe parameters (release, no debug, etc.) and all outputs, together with an IOC overview are provided as output.

The general workflow for an operator is to:

- Select the right project
- Upload a payload (raw, x86/x64) generated by Cobalt Strike, Stage1 or a custom payload, or alternatively point to a staging location.
- Configure output formats and transformations

Payload generator offers the latest offensive R&D to help bypassing AV and EDR products, such as direct system calls, techniques to blend in with TI ETW, ROP gadgets, sleep masking, stack spoofing and much more.

Once the configuration is complete the operator can start a build. This will generate the final payload in the format requested and provide various metadata for payload management (e.g. tracking exact options configured in this specific payload).

In-Phase Builder

The In-phase builder can be used for transforming a binary payload in various output formats including script based formats (e.g. HTA, VBScript and JScript) as well as more complex file formats and payload-wrapper formats (e.g. CHM, ClickOnce). Intent of builder is to make sophisticated, multilayered phishing samples with a couple of clicks.

Using Builder, an operator can convert into various other formats. Shellcode can be converted into various formats, and each format can allow for new conversions. For example:

- Shellcode can be converted into ClickOnce, this is deemed a 'final format' and does not allow further conversions. ClickOnce is a deployment technology that allows users to run (and optionally install) applications from a web server.
- Shellcode can be converted into JScript or VBScript, which allows for new conversions (e.g. CHM, HTA, SCT, WSF, MSC).

Stego Loader

Stego loader allows the operator to use steganography techniques as used by certain nation state actors. Stego loader embeds a payload into an image file using Least Significant Bit Steganography. After the payload is embedded into the image, the image is still valid and viewable. Host the image with the hidden payload on a redirector and have a stager retrieve it. The result will be decoded (and optionally decrypted) before execution/dropping/loading. Encoding is just a means to obfuscate to bypass automated detections. This does not provide any safety, only obscurity.

Office Intrusion Pack

The Office Intrusion Pack generates VBA macros based on various templates and settings. Currently it generates a .txt file with VBA code. The main use case is generating malicious macros to be used in maldocs for initial access or persistence (e.g. by backdooring normal.dotm). The templates used for generating the macro payloads are based on various techniques observed in the wild, including techniques used by APTs (such as UUID encoding of payloads and callback thread execution). Some of these include:

- Bypassing Mark of the web restrictions in Office - Certain customers have hardened Office and enabled a policy which blocks macro execution for downloaded files. We have strategies to bypass these restrictions.
- Bypass macro signing - approaches that can be taken when you encounter a target that solely accepts signed macros.
- Lateral movement via Office macros - Various companies struggle with securing Office macros. Specifically in the financial industry, companies are quite risk averse in hardening macros as they are often widely used in their business. This writeup shows various attack patterns to exploit weak Office settings.

- Macroless Office reconnaissance - Sometimes you want to obtain information on a workstation without running an Office macro. Specifically, you may need an exact Office version so that an EvilClippied macro can be delivered later.

Language Panda

Language Panda changes language-artifacts within your office document to make it appear as if it was created using a Russian / Hebrew / Ukrainian / etc. MS Office installation.

- Change the language characteristics of a Word, Excel, or PowerPoint document.
- Extendable with various languages: Russian released in first version. Later Hebrew, Ukrainian, etc..
- Support for all Office Open XML (OpenXML or OOXML) Word, Excel and PowerPoint filetypes: docx, docm, dotx|m, xlsx|m, xlam, xlsb, pptx|m, etc.

Planting these false flags in your MS Office payload may help making your red teaming operation to appear to be more realistic to a responding blue team.

Through Phase

Outflank C2

Outflank C2 is Outflank's custom Command-and-Control framework. Outflank C2 (formerly named Stage 1 C2) is a light C2 framework that is aiming to be as OPSEC-safe as possible. Another core feature of Outflank C2 is that it's multi implant platform support: it has full native implant support for implants running on Windows, macOS and Linux. This implant support is fully native, meaning no need for python or other interpreters, it is all handled in C/C++.

Outflank C2 has support for loading Beacon Object Files, linking of implant – even cross platforms, SOCKS tunneling, different ways of loading shellcode so you can start another C2 implant straight from Outflank C2, and many different OPSEC features for EDR evasion. Communication channels supported are HTTP(S), raw TCP, SMB and file-based.

Automation if the C2 server and implant actions can be performed with python.

Lateral Pack - ShovelNG

ShovelNG is our most OPSEC safe toolkit for lateral movement in Windows networks. It abuses different techniques for remote code execution such as WMI, WSMan and DCOM. It also allows flexibility to the red team operator to select relevant LOLBAS and select the loader to be usedIt integrates with Cobalt Strike and Stage1 but can be used with any Command and Control framework with BOF support.

ShovelNG's functionality can be extended for bypassing UAC or perform Local Privilege Escalation, by using the added SCMUseKerberos tool.

Lateral Pack – pyShovel

pyShovel is a a python implementation of some of the ShovelNG functionality. It has added use in cases where target networks are tightly firewalled and only WMI/RPC and SMB protocol functionality is available. It leverages the Impacket libraries to do so. In order to ensure proper functioning this tool is embedded in a docker environment that will install all the needed libraries.

SharpFuscator

SharpFuscator is a .NET obfuscator which compiles C# repositories, source code or binaries making the following change to source code:

- Names of namespaces, classes, delegates and interfaces will be replaced with randomly generated names
- Strings will be encrypted and a decryption function will be added to main class
- For some projects, code will be patched. I.e. Rubeus is rewritten so that all Rubeus LDAP queries communication is encrypted.

- Array initializations can be reversed (optional)
- Check for debugger presence can be added (optional)
- Check for sandbox artifacts presence can be added (optional)

For some tools we have noticed that even the XORed version of specific strings is flagged by AV/EDR. In these cases it is possible to use AES for strings encryption.

If the use of Costura is detected, it will be reimplemented in the output assembly. For this task Mono.Cecil | Mono is used to inject a module initializer to be able to resolve references from embedded resources Module initializers in C#.

ShapFuscator is designed for AV/EDR static signature evasion. By analyzing the assembly with tools such as dnspy it is relatively easy to identify the encryption keys, the assembly features or the original tool.

Credential Pack

Credential Pack is a collection of tools that can be used with any Command and Control framework with BOF support and allows the red team operators to extract (dump) and obtain credentials.

The toolset currently consists of the following tools:

- **DumpertNG**: an evolved and improved version of the original public Dumpert tool from outflank. DumpertNG uses a `PROCESS_CREATE_PROCESS` handle to create a process snapshot and uses direct system calls and API unhooking to dump LSASS memory from the snapshot handle.
- **ProcessDupMiniDump**: LSASS dump tool that uses a `PROCESS_CREATE_PROCESS` handle to create a forked LSASS process and uses direct system calls and API unhooking to dump LSASS memory from the forked process handle.

- **HandleDupMiniDump**: LSASS dump tool that enumerates active processes on a system for existing LSASS handles and "when found" duplicates an existing handle and use direct system calls and API unhooking to dump LSASS memory from the existing process handle.
- **DecompreZz**: the produced LSASS memory dump files from the "userland" LSASS dump tools above are compressed using native API calls and saved to a random filename within the Windows temp folder. An additional decompress tool is provided to decompress and obtain the original memory dump file.
- **KernelKatz**: LSASS dump tool which uses a technique called "Bring Your Own Vulnerable Driver" (BYOVD) to install a "vulnerable" signed kernel driver on the system, which we can abuse to dump credentials from the LSASS process memory within kernelspace.
- **PasswordSpy++**: Our implementation of NPLogonNotify with AES encryption and SACLS on the output file. Every password is encrypted using a different key written to a file in `C:\Windows\Temp\` with a read|del everyone, write admin SACL. A python script allows decryption.

DLL Hijack Library – Local Privilege Easy (LPEasy)

Local Privilege Easy (LPEasy) is a DLL hijacking project to facilitate Local Privilege Escalation (LPE) attacks.

This project helps you in abusing DLL hijacks on a system. It solely helps in the abuse, not the detection process of a DLL hijack.

Out Phase

Hidden Desktop

Hidden Desktop is Outflank's implementation of 'hVNC' functionality of top tier financial malware families like Gozi and TrickBot, plus a few extra tricks. It provides a full graphical desktop experience to a remote system, without the victim knowing. It is essentially an extra desktop

session on the target system, in the remote user's logon context. Have you ever required access to a fat-client application, locally connected hardware tokens and/or coockies of the user that is running on the victim's system in an OUT phase? Using Hidden Desktop you can spawn a new desktop on a user's system and control that desktop as if it was RDP. However, the data is transferred via your C2 channel of choice. You can use it to open the Explorer, or for example any browser installed on the system.

- Hidden Desktop is a feature in top-tier financial malware
- Usually referred to as HVNC or HiddenVNC on darknet markets
- Despite the name Hidden Desktop is NOT relying on the VNC protocol but uses specific Windows APIs that are supported since Windows 2k.
- Malware family examples using similar functionality: Dridex, Gozi

Fake Ransom

FakeRansom is a tool developed and used to support during the out phase of a ransomware attack simulation. It is fake, yet real-life-like ransomware. It hijacks the screen and shows a full screen ransom notice combined with ongoing file listings of files of that computer. It also blocks common keyboard combinations (i.e. `ALT-TAB` and `CTRL-ALT-DEL`) of a user trying to exit the ransom notice. It creates the sense of urgency and a stress factor that is often overlooked in a simulation.

Support Phase

BlueCheck

BlueCheck is a collection of tools that help you identify possible Blue team activity. It integrates with RedELK. It is compatible with Cobalt Strike using reflective DLLs or by using Beacon Object Files (BOF). The reflective DLLs can also be used in other C2 frameworks, which support reflective DLL injection.

The toolset currently consists of the following tools:

- **CertCheck**: enables the RedTeam operator to check SSL certificate information (SSL interception e.g.) from a specified website.
- **PasswordChangeCheck**: enables the RedTeam operator to check specific Active Directory accounts for password changes.
- **SecurityToolCheck**: shows information about security tools and products running on the system.

BeaconBot

BeaconBot is a Cobalt Strike CNA script that performs all kind of administrative duties. You can consider this bot to be your Cobalt Strike secretary.

The most important functions include:

- Administration of beacons
    - Beacon notifications and digest
    - Track sleep and late beacons
- Triage of incoming beacons
    - Auto sleep beacons
    - Auto exit beacons
    - Auto shinject beacons

The triage commands are specifically of value when a 'user behaviour based persistence' (e.g. application startup) is applied and as such 'beacon storms' can occur when a user starts an application many times.

Cloud

ROADtune

ROADtune abuses non publicly known insecurities in Microsoft Intune with the goal of getting into a target network. It allows the red teamer to:

- Enroll devices into Intune
- Fake device state to get a device compliant
- View apps assigned to users and devices
- Download those applications locally (including decryption of the packages)

A full attack chain with ROADtune allows to enroll a fake complaint device into a target's Intune environment and download applications pushed to compliant devices. The red teamer can then analyse those applications, and use these as a next attack step, e.g. use a pushed VPN application to gain access to the network, or to gather and abuse credentials stored in those applications.

ROADtune was developed for Outflank by the external researcher Dirk-jan Mollema, so it is no surprise it works in tandem with his famous roadtx tool that handles the Azure AD interaction.

PhisherPrice

PhisherPrice allows the red teamer to easily abuse the EntraID Device Code Flow in order to obtain an authentication token of a phished user. In contrast to various other phishing approaches via device codes, PhisherPrice does not require a device code to be included in a phishing mail or the distribution of QR codes with all related timing complexities. Luring users to a phishing website should be sufficient.

With PhisherPrice a rogue phishing website is created. Upon visiting the website, a device code is requested from Microsoft. Via our phishing website, we lure the user in entering the code and

authenticating to their EntraID account. Upon a successful phish, PhisherPrice obtains a JWT token, and provides it to the operator by storing it on-disk or leveraging various notifiers such as SMTP, ntfy.sh, MS Teams webhook, etc. With the obtained JWT tokens, you can perform various actions on behalf of the regular user.

O365TokenExtractor

A BOF that dumps available O365 tokens stored for the current user by the Web Account Manager.

The WAM (Web Account Manager) component stores O365 access tokens in a cache that is accessible for the current user. Running applications that interact with Office365 (e.g. MS Teams) will store their tokens in this cache.

MSAL (Microsoft Authentication Library) can call WAM, a Windows 10 component that ships with the OS. This component acts as an authentication broker and allows applications to piggyback on the authenticated Windows session.

O365TokenExtractor can dump the available tokens for the current user.

Beacon Booster

Outflank Security Tooling is a standalone offering and does not rely on Cobalt Strike. However, we do have a tool called Beacon Booster that is applicable for improving only Cobalt Strike beacons.

Cobalt Strike is a very customizable adversary simulation tool. To improve evasion, it is recommended to customize the beacon implant to your situation. The beacon can be customized using the Aggressor script as well as the various kits in Arsenal kit.
From the Arsenal Kit, two elements stand out in importance for OPSEC safety for the beacon.bin;

the Reflective Loader (UDRL) and the Sleep Mask. Beacon Booster is an OST tool that helps the red teamer with:

1. changing the UDRL (currently 2 supported) plus bypasses for Windows Defender Exploit Guard, remove Guard Pages, Clear PI Callback and perform Stomping)
2. Insert a custom Sleep Mask (currently 2 supported).
3. Perform YARA rule analyzed pre and post Beacon Booster, so the operator understands the detection improvements and residue.

The outcome of Beacon Booster is an optimized beacon.bin, that can be used directly in your operation, or be used as input in other OST tools such as Payload Generator.

Miscellaneous

UntrustProcess

A BOF that downgrades a process token to Untrusted integrity level. Requires high integrity token (admin privileges). The principal you are executing as must have control over the target process token.

KerberosAsk

Implementation of various Rubeus-alike Kerberos functionality in BOF format. It uses a custom implementation for ASN1 encoding. This BOF focusses on ticket requests and renewals. It allows you to perform offensive Kerberos operations without having to do process injections and run Rubeus' .NET code.

Coercer

Collection of various methods to coerce a system to perform authentication towards another system (generally, a relaying server).

In many cases you will want your relaying server to relay the incoming authentication packets to an ADCS HTTP endpoint (in case of SMB connections) or LDAP for RBCD/ShadowCred abuse.

KernelTool

KernelTool is toolset that uses a kernel driver helper to interact with the system. The toolset includes commands for setting and removing process protections and modifying callbacks. It uses read/write primitives in the vulnerable the kernel driver to modify kernel memory to perform these actions. It comes with a vulnerable driver that is not on the Microsoft block list.

SideloadTrigger

SideloadTrigger is a collection of methods to trigger sideloads. E.g. to trigger a DLL sideload in a Windows service through a hijackable PATH.

RPC and Registry Tradecraft

Collection of miscellaneous scripts that we create during research time, ad-hoc during projects, or for talks/presentations. This collection may be extended at any time.

Dump Mstsc

Dump credentials of RDP client processes on the system. Can decrypt credentials of the current user's context when in medium integrity, and credentials of other users when in high integrity.

KernelKatz

KernelKatz allows reading LSASS memory through the kernel by leveraging a kernel driver and can dump hashes of login sessions. It comes with a vulnerable driver that is not on the Microsoft block list.

EvilClicky

EvilClicky is a Python-based ClickOnce payload generator that creates the necessary ClickOnce files for initial execution. It leverages a trusted executable to bypass SmartScreen controls.

Outflank – C2 Tool Collection

A ready to be used compiled version of our publicly shared repository of Red Teaming tools, the Outflank - C2 Tool Collection. The public repo does not contain ready to go compiled BOFs. New tools are released internally first for the OST community. This repository contains a collection of tools which integrate with Cobalt Strike (and possibly other C2 frameworks) through BOF, reflective DLL loading techniques, and .NET assemblies.

Ivanti Connect Secure VPN Privilege Escalation

This is a Beacon Object File (BOF) which exploits Ivanti Secure Access (previously Pulse Secure) VPN client (CVE-2023-35080). From a low-privileged user mode, it exploits a vulnerability in the kernel driver to overwrite the beacon process token privileges with a system process token privileges. The kernel driver symbolic link has an ACL defining that Everyone can open a handle to it. This tool came from an external researchers Tijme Gommers and Alex O from Northwave.

PowerShell Tradecraft

PowerShell Tradecraft is a collection of tools that support local and remote PowerShell execution. Remote PowerShell execution is performed by lesser-known techniques for lateral movement that PowerShell can leverage, tunneling over native Windows techniques. It can be used in Stage 1 C2, Cobalt Strike and there is also a standalone version for on-host cli usage.

Keyper

Keyper is a small keylogger written in .NET made for usage within your C2 implant (or standalone). It has its use case in specific scenarios despite it not being in BOF format.

Due to the rapid pace of updates to Outflank Security Tooling, this document may not fully represent the latest releases. For the most recent updates, please visit our OST Releases site - https://www.outflank.nl/services/outflank-security-tooling/releases/

Reference OST architectural diagram