Unmasking HiddenFace

MirrorFace’s most complex backdoor yet

Dominik Breitenbacher

Malware Researcher

Digital Security
Progress. Protected.

Unmasking NOOPDOOR

MirrorFace’s most complex backdoor yet

Dominik Breitenbacher

Malware Researcher

Digital Security
Progress. Protected.

Dominik Breitenbacher

- Malware researcher @ ESET since 2019
- Research focus
  - MirrorFace – LODEINFO
  - Kimsuky

dominik.breitenbacher@eset.com

@dbreitenbacher

dbreitenbacher

Agenda

- **MirrorFace overview**
- **HiddenFace (NOOPDOOR)**
  - Introduction
  - Execution chain
  - Technical details

MirrorFace

MirrorFace

- **China-aligned threat actor**
- **Active at least since 2019**
  - Activity often attributed to APT10
- **LODEINFO malware unique for the group**
- **Exclusively targeting Japanese entities (?)**

Victimology

Media

Defense-related companies

Think tanks

Political entities

Academic institutes

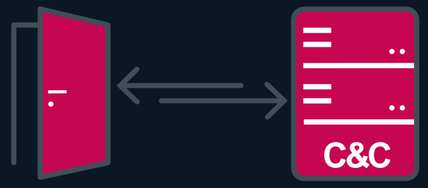
HiddenFace
(NOOPDOOR)

HiddenFace

shellcode

modular

evasive



active and passive communication

data categorization system

domain generation algorithm

Overall complexity and versatility surpasses LODEINFO

Victimology

Media

Defense-related companies

Think tanks

Political entities

Academic institutes

How we discovered HiddenFace

- August 2023
- Japanese research institute
- Exploited a vulnerability in FortiOS/FortiProxy

  → NOT via spearphishing
- LODEINFO deployed

  → MirrorFace
- HiddenFace deployed

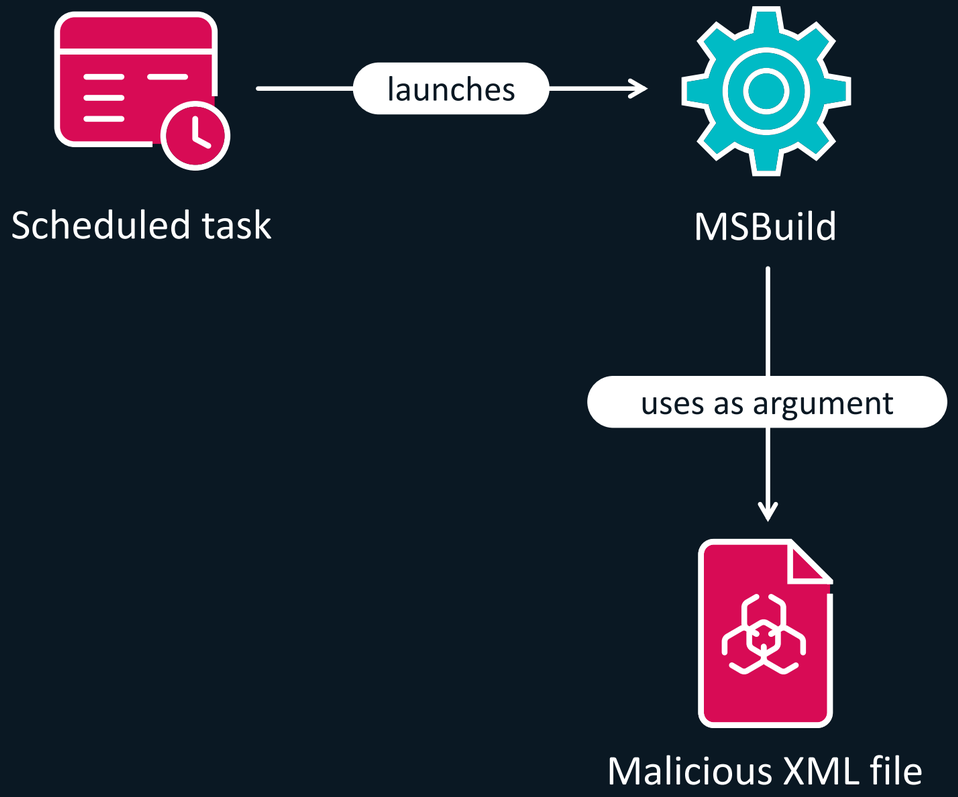
Execution Chain

Execution chain - Installation

Scheduled task

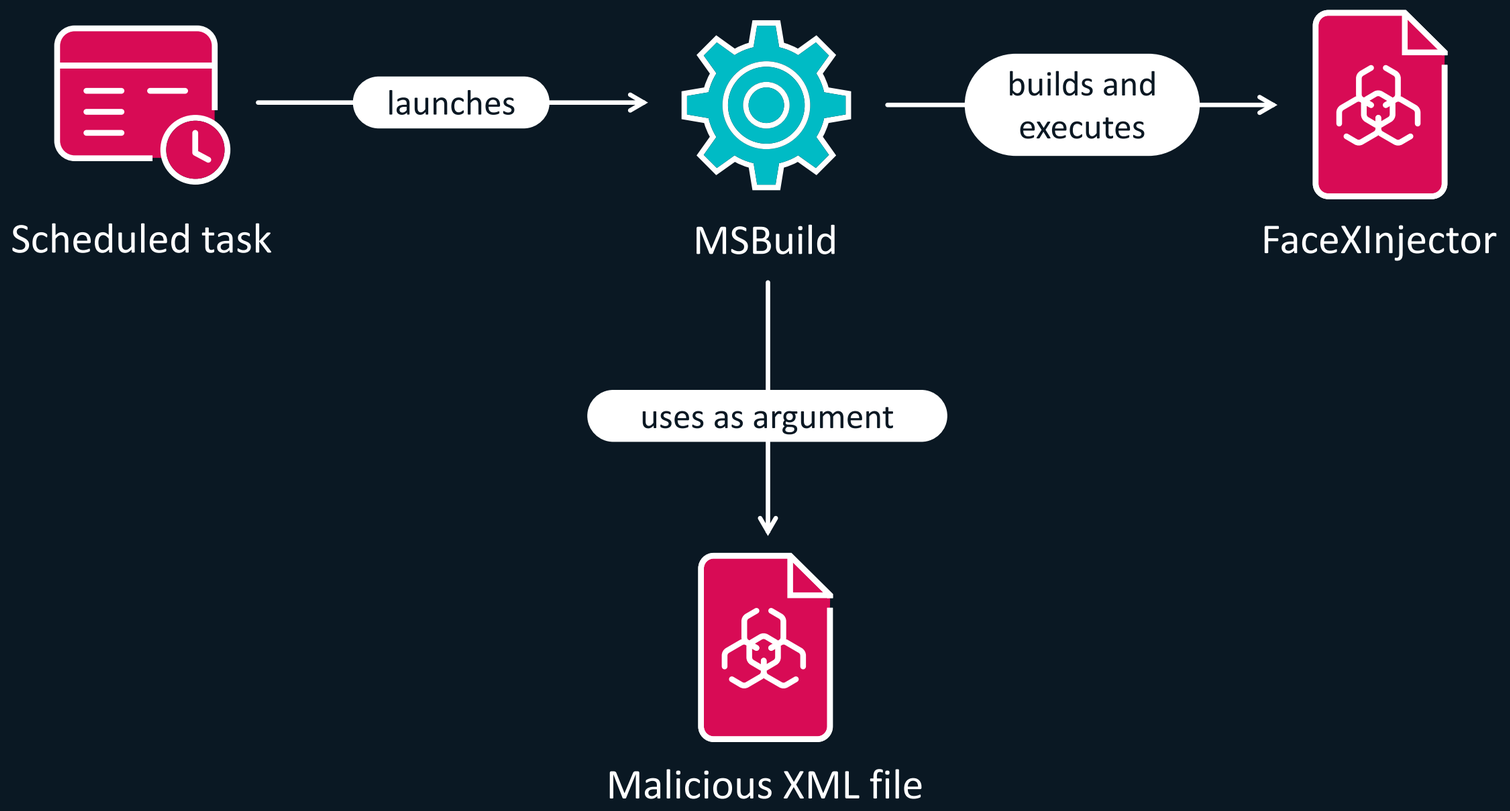
Example: automatic-device-check or createobject

Execution chain - Installation



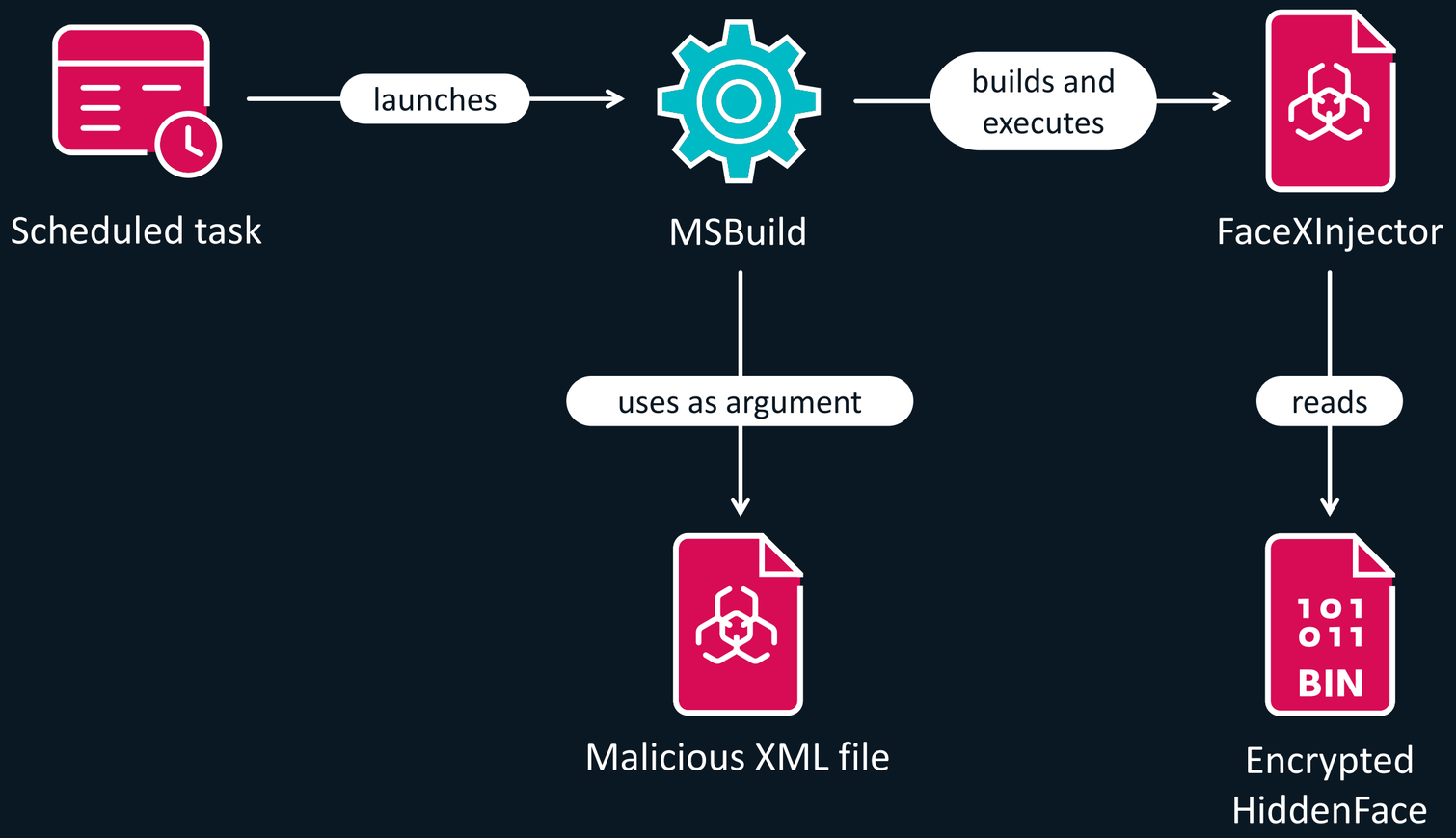
Example: diskmgmt.config, BrowserSettingSync.xml, or BluetoothDesktopHandlers.xml

Execution chain - Installation



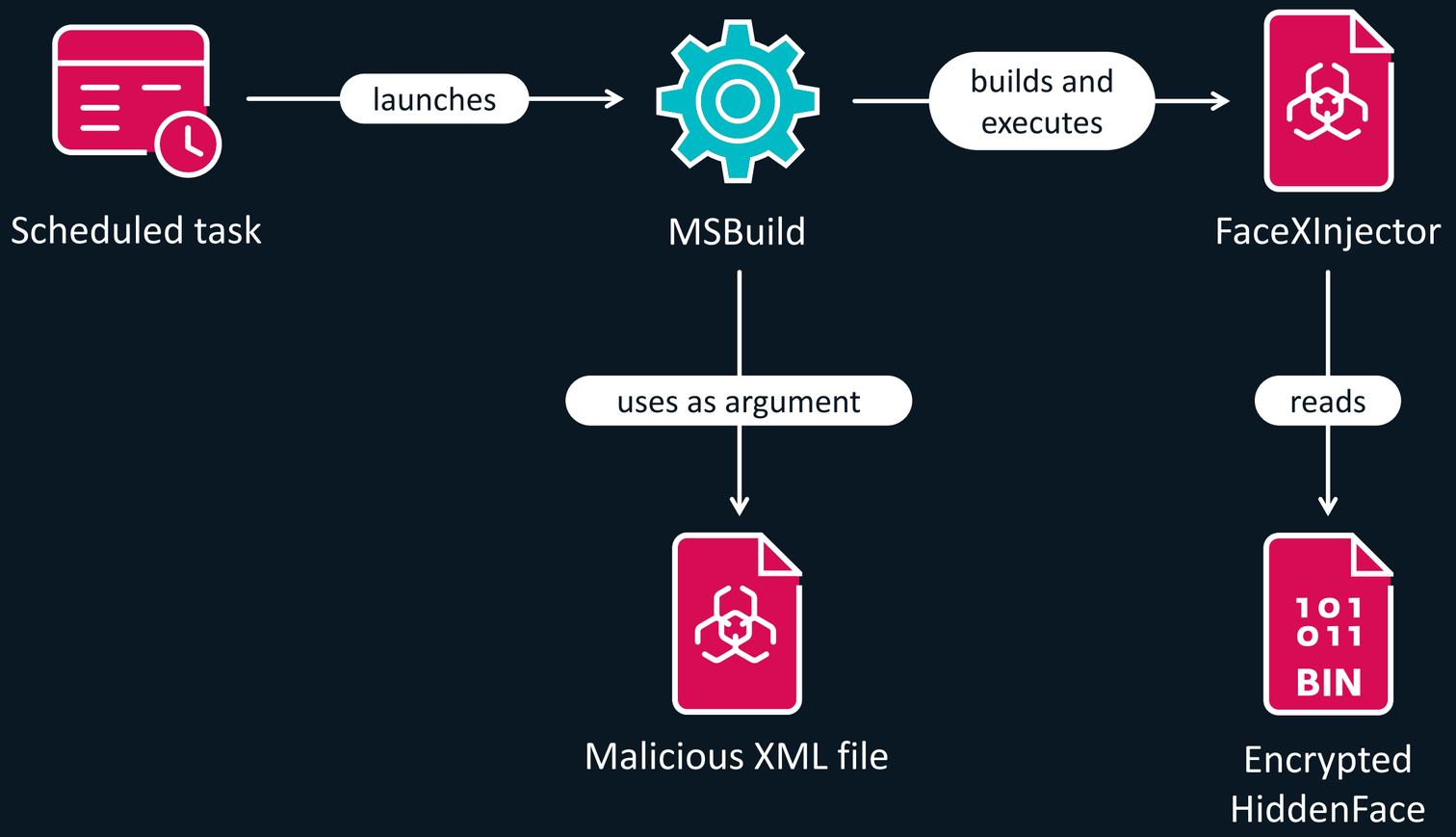
FaceXInjector = NOOPLDR

Execution chain - Installation

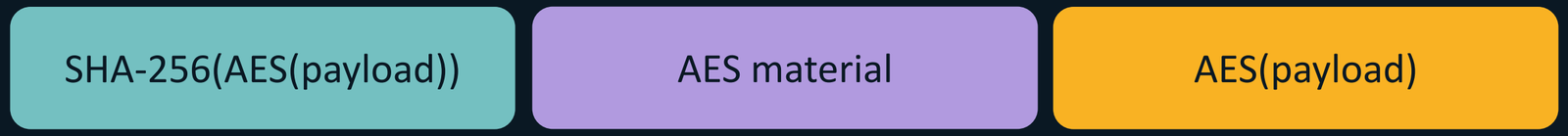


Example: ActivationManager.tlb, LaunchWinApp.dat, or Windows.Devices.Custom.dat

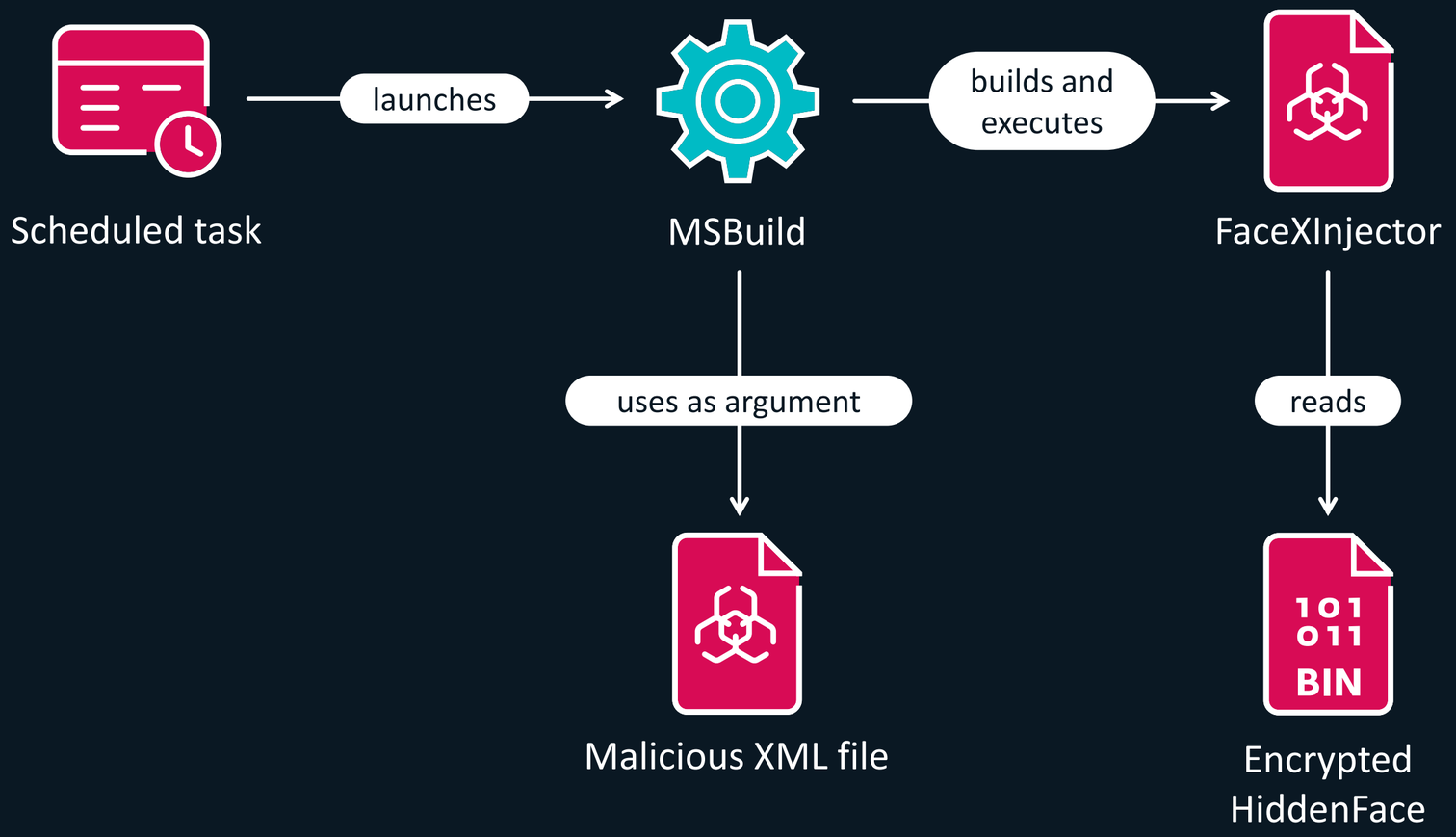
Execution chain - Installation



File content:



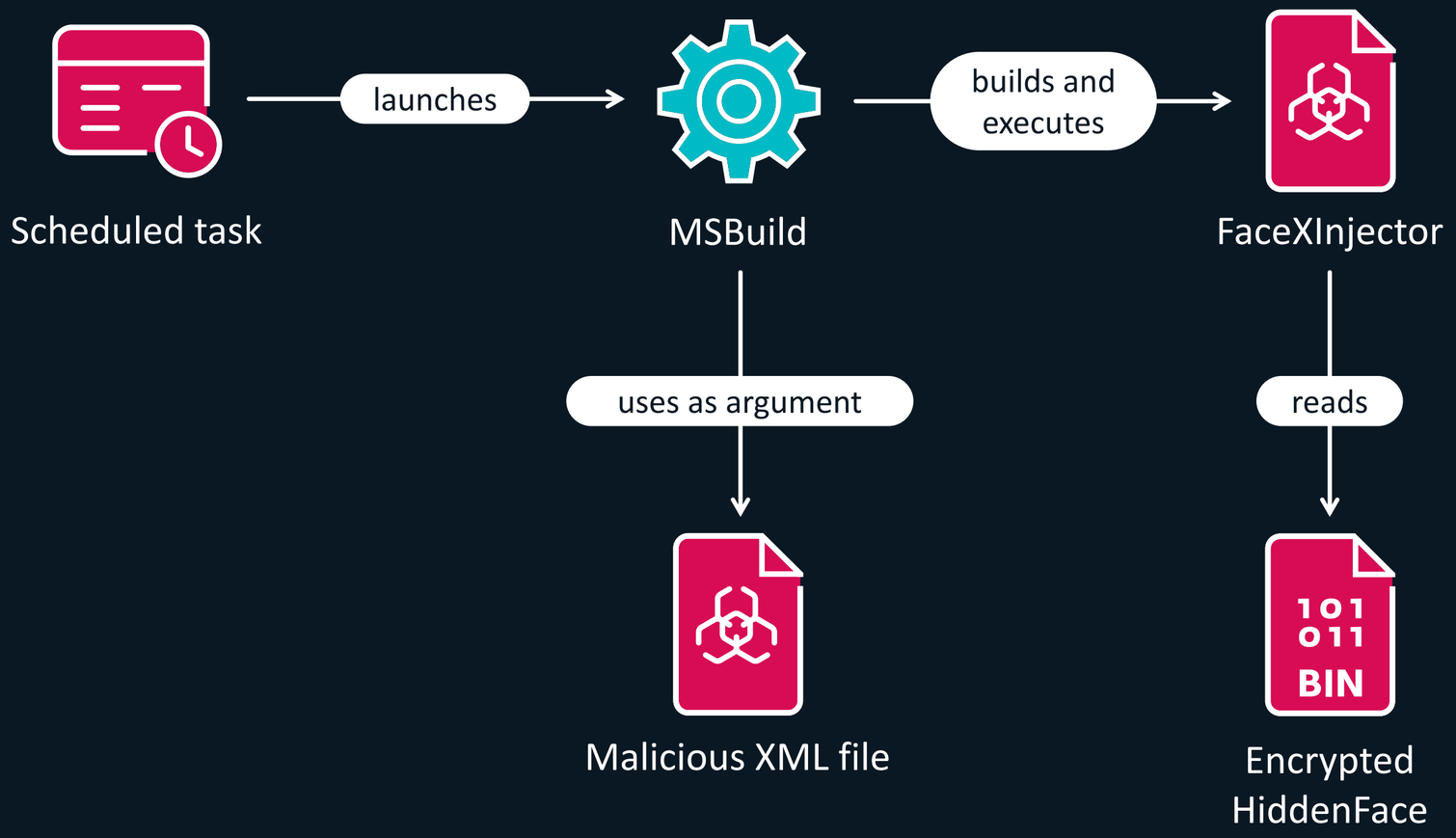
Execution chain - Installation



File content:

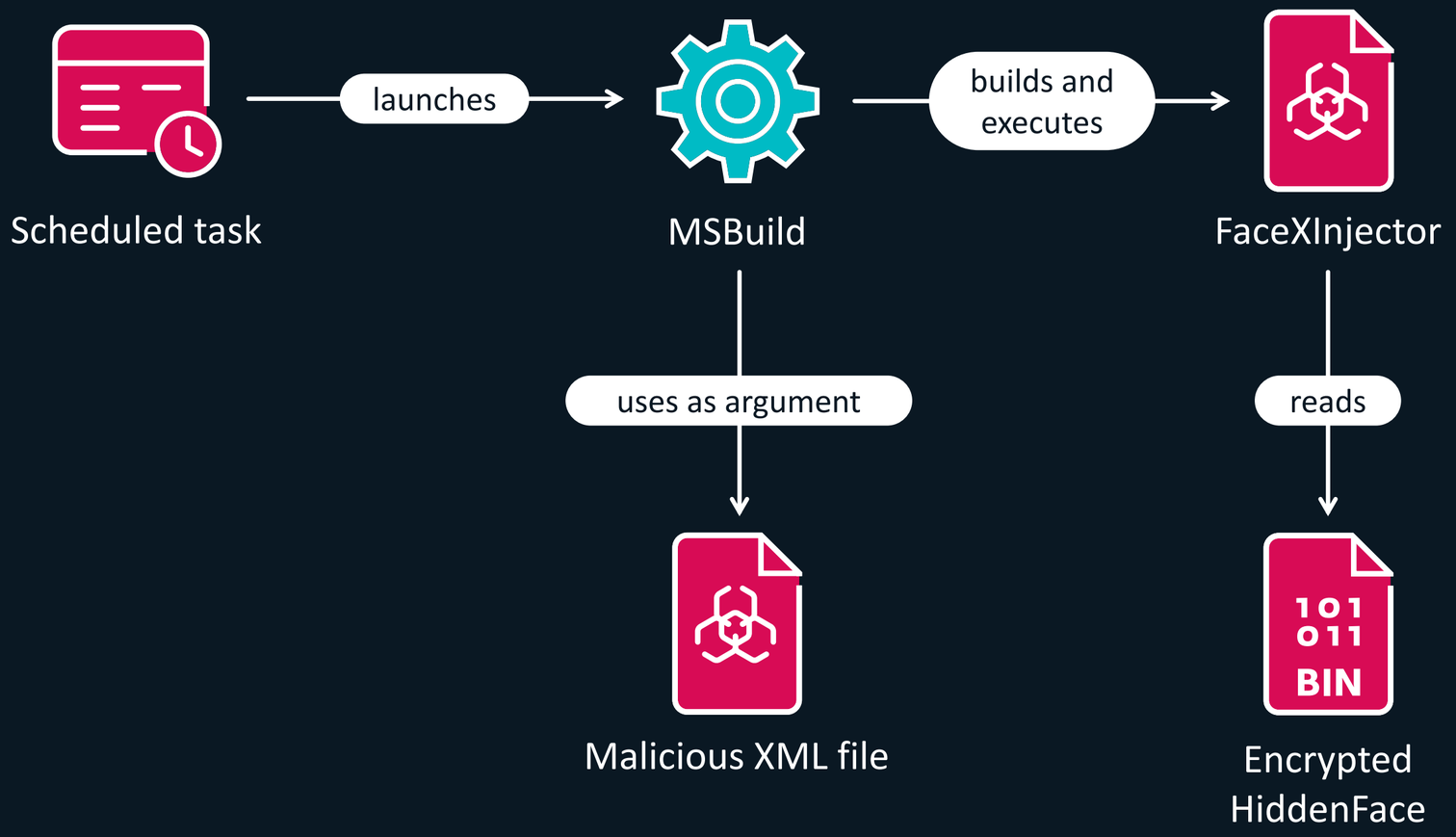
SHA-256(AES(payload))	AES material	AES(payload)

Execution chain - Installation



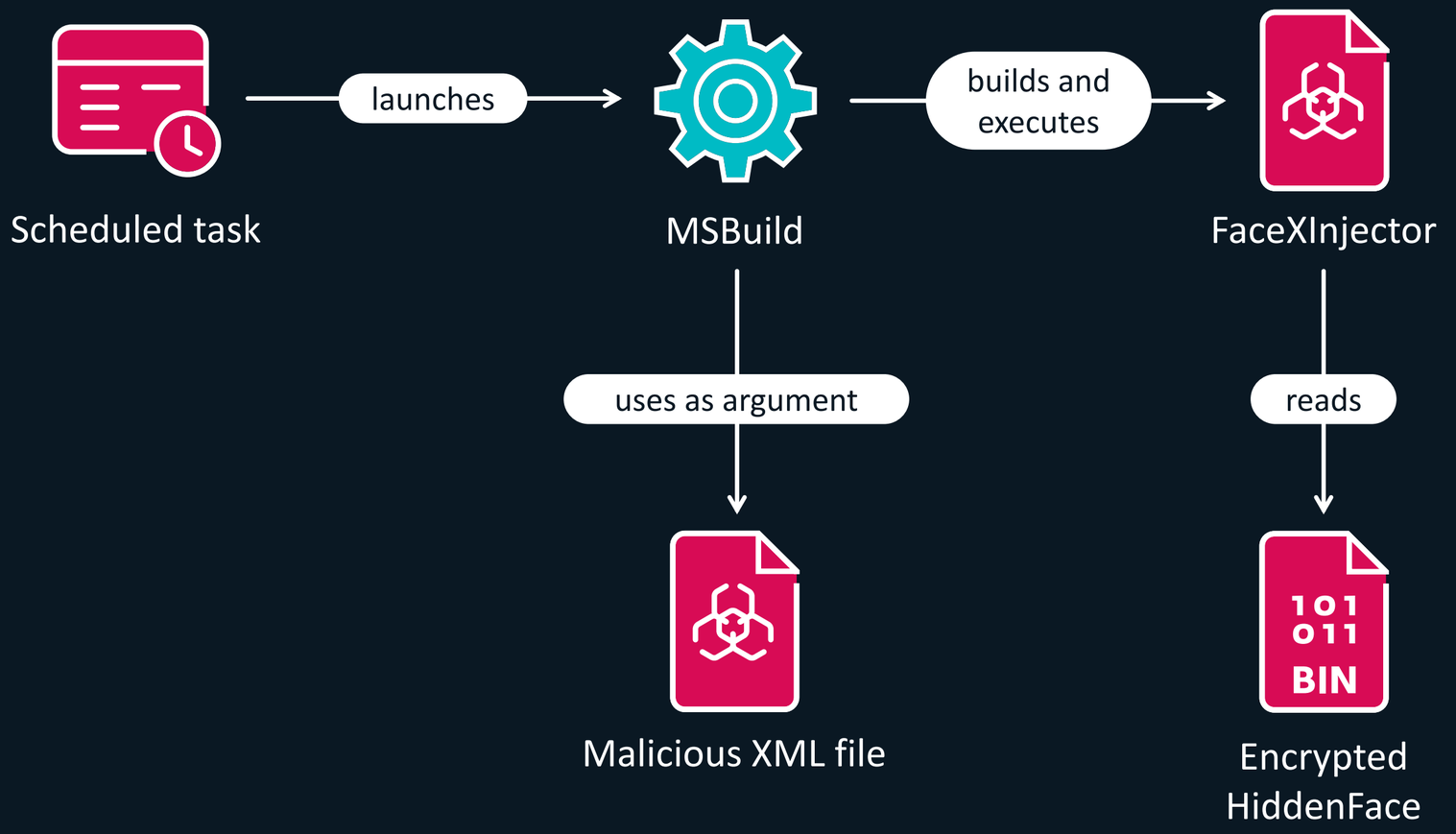
SHA-256(AES(payload) **)**

Execution chain - Installation



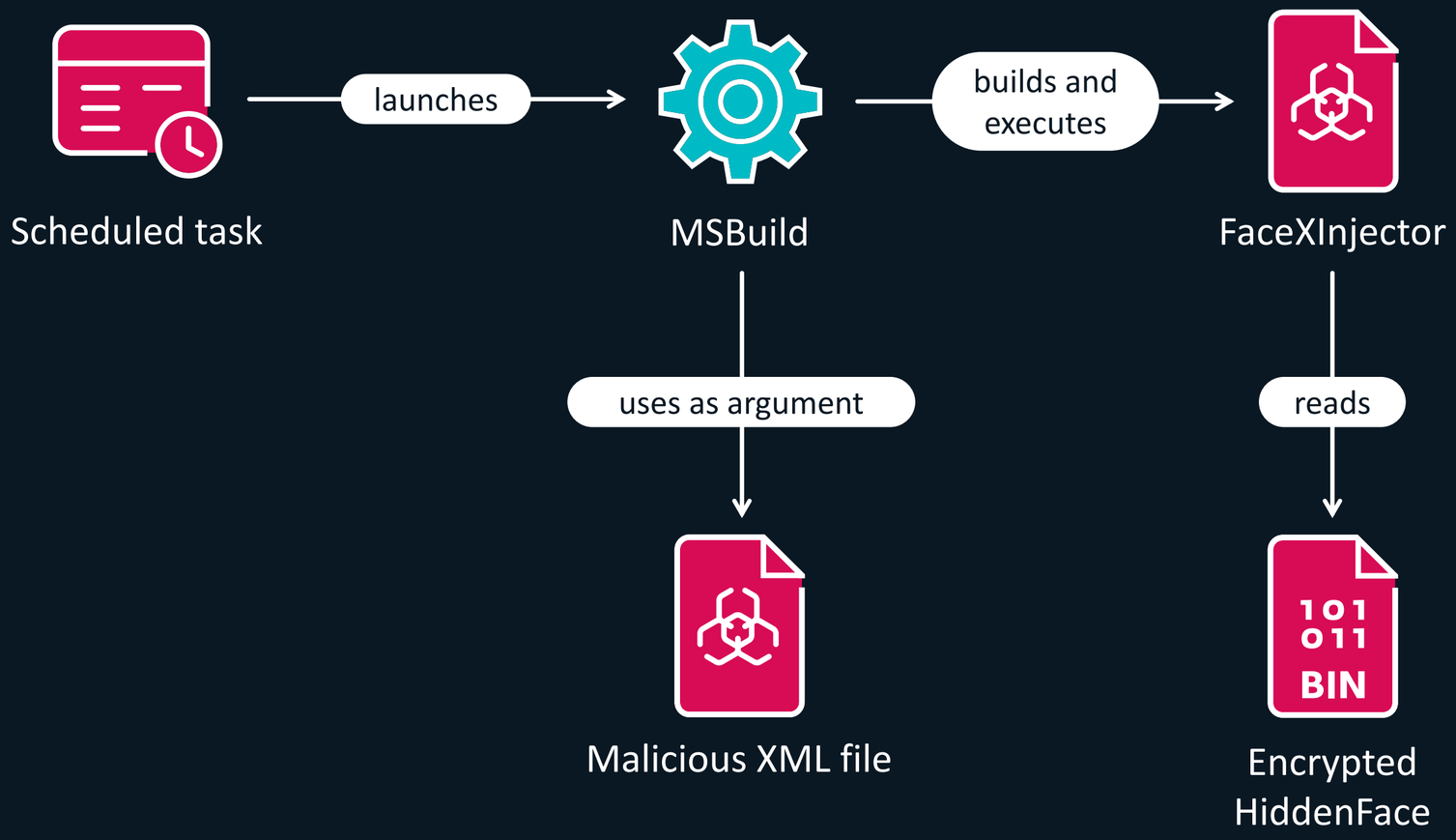
SHA-256(AES(payload)) = **SHA-256(** AES(payload) **) ?**

Execution chain - Installation



SHA-256(AES(payload)) = **SHA-256(** AES(payload) **)**

Execution chain - Installation

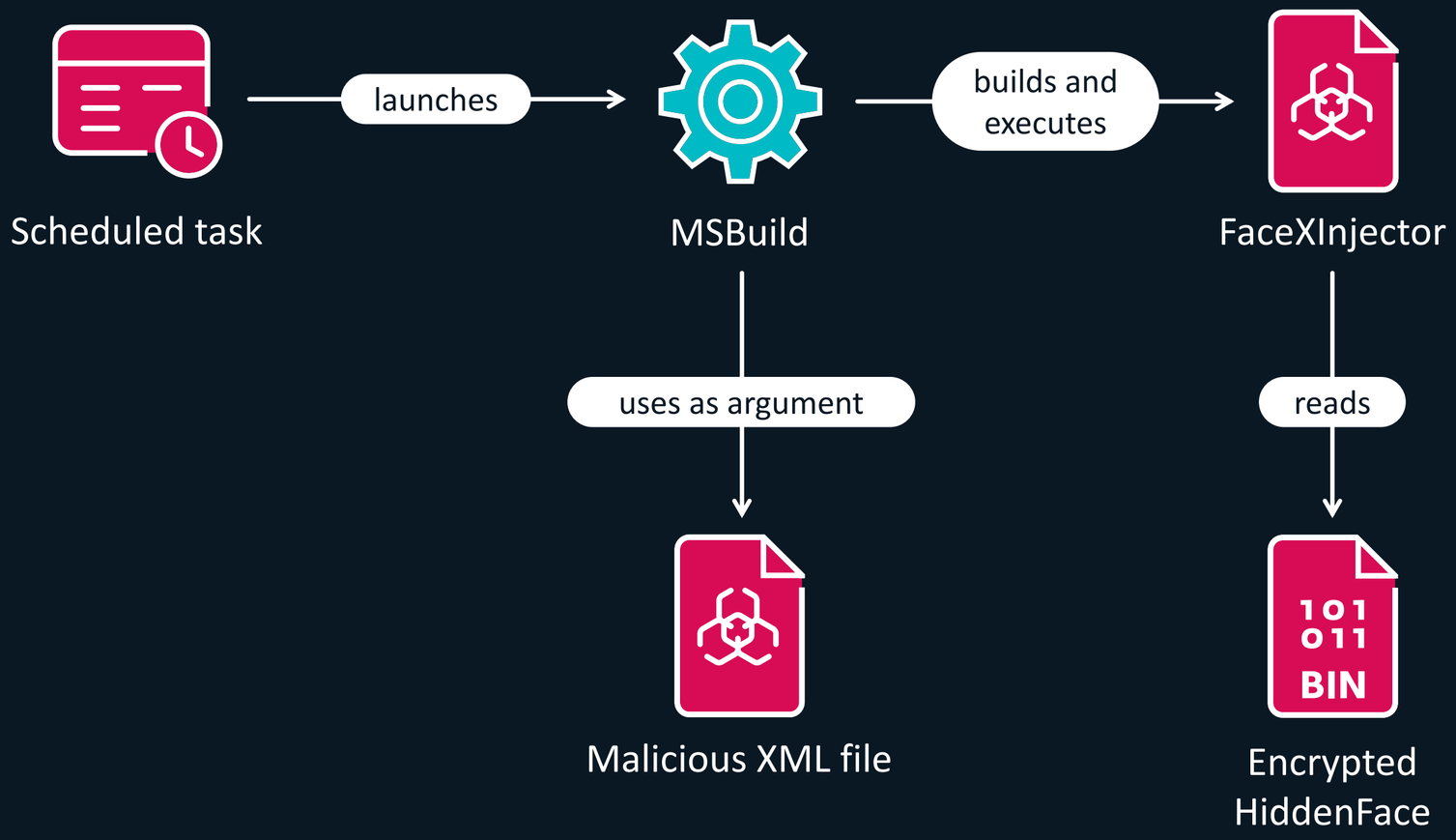


SHA-256(AES(payload))

AES material

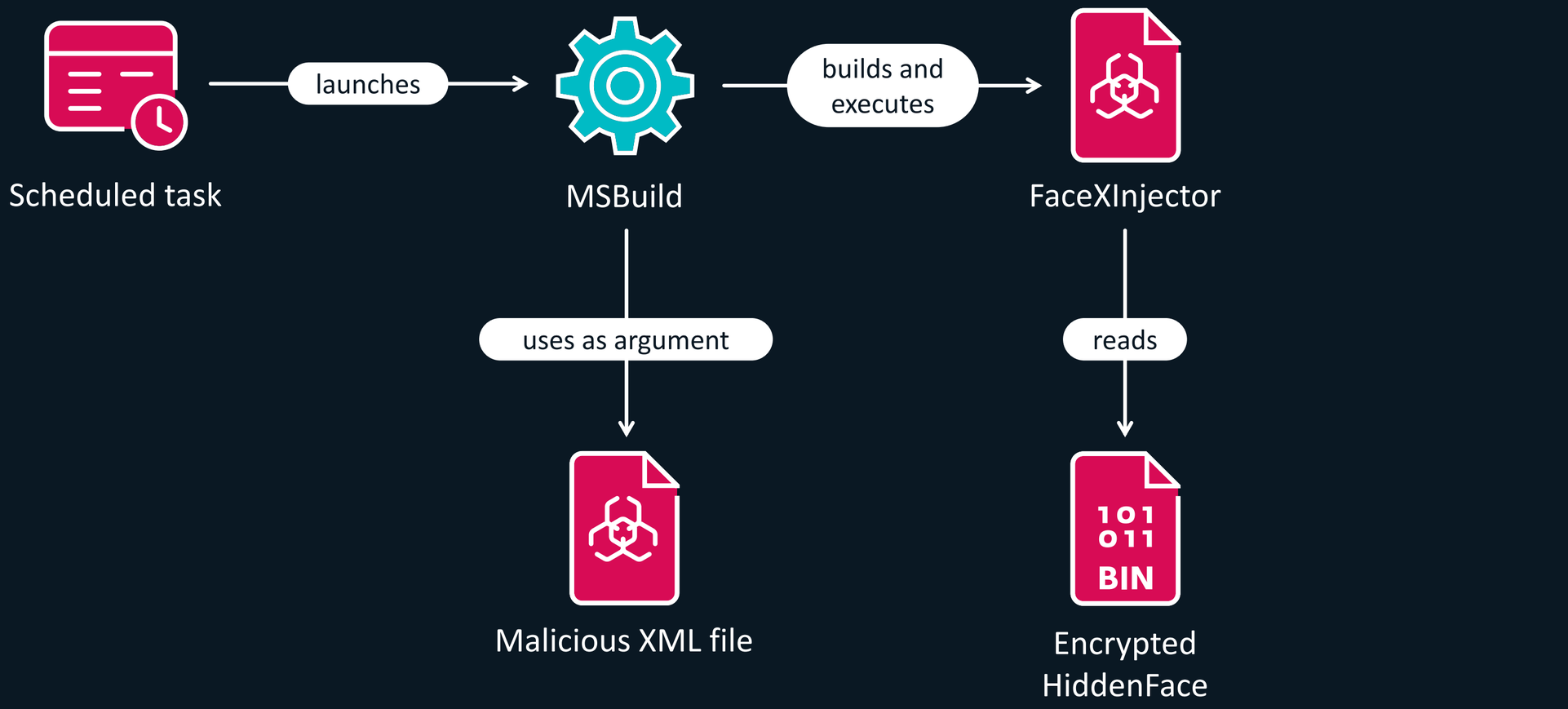
AES(payload)

Execution chain - Installation



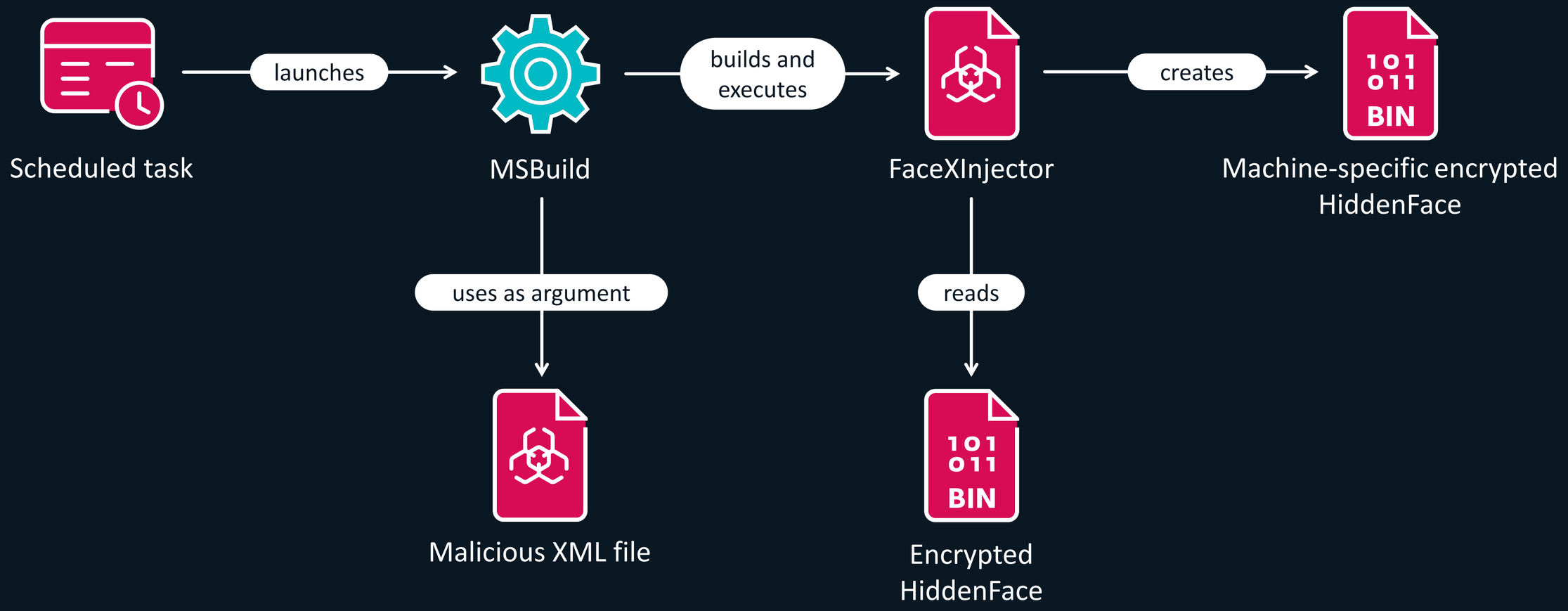
SHA-384(AES material **)**

Execution chain - Installation

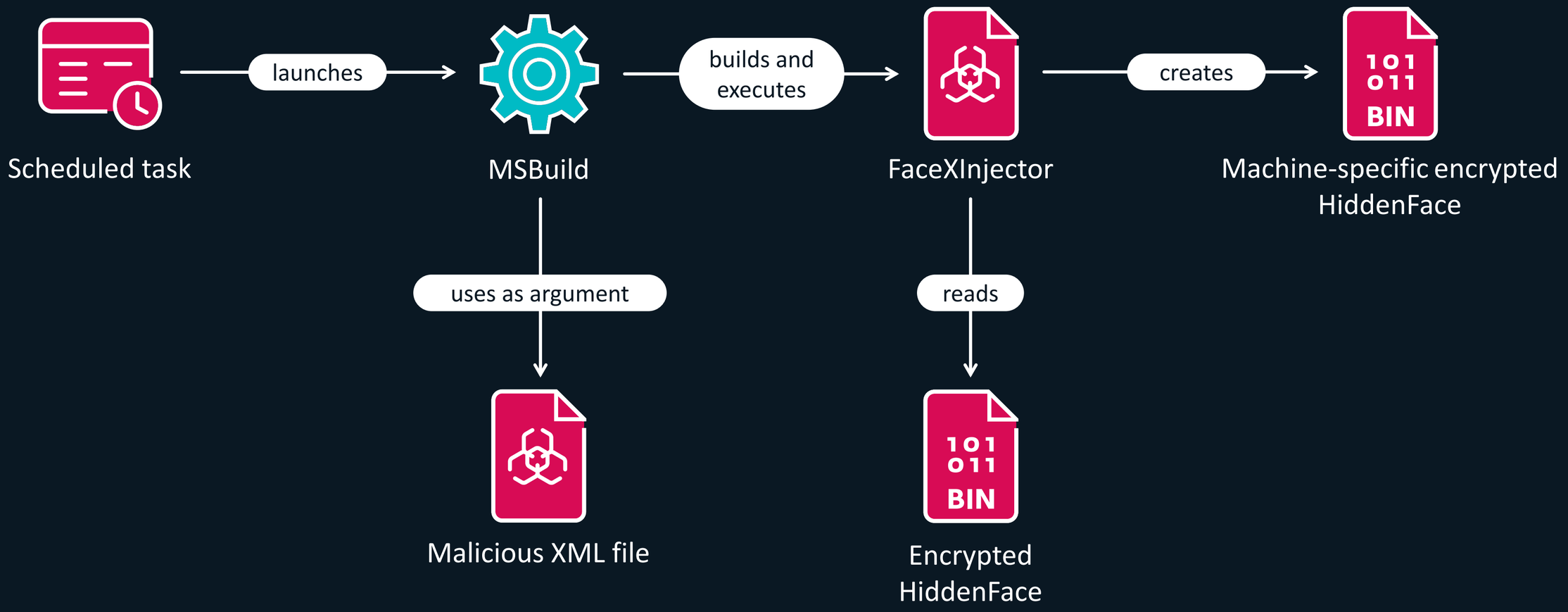


Decrypt(AES(payload) **)**

Execution chain - Installation

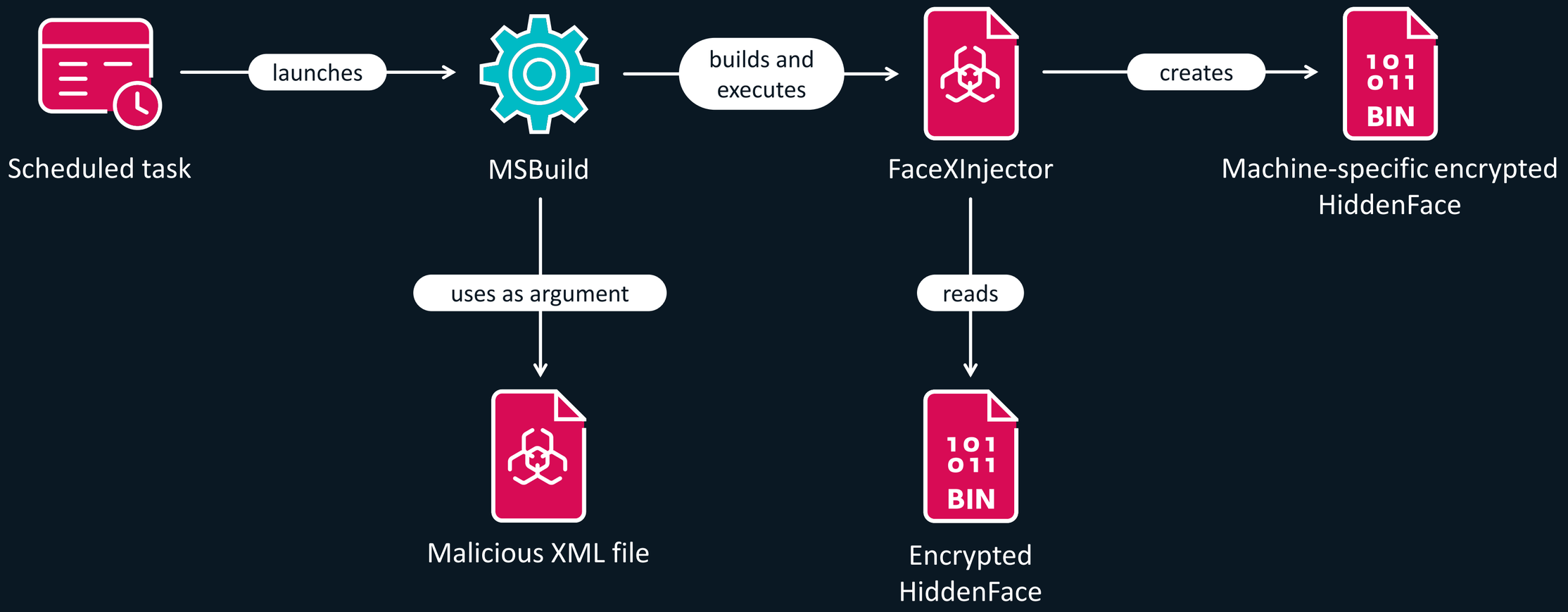


Execution chain - Installation



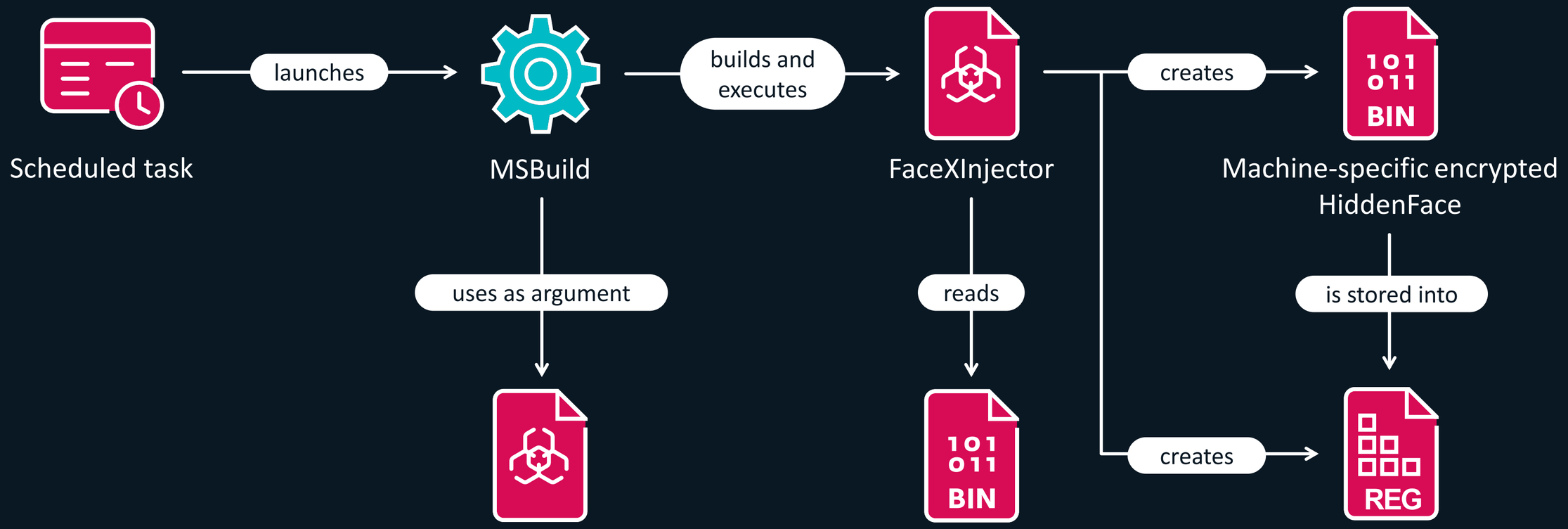
HKLM\Software\Microsoft\SQMClient\MachineId + hostname

Execution chain - Installation



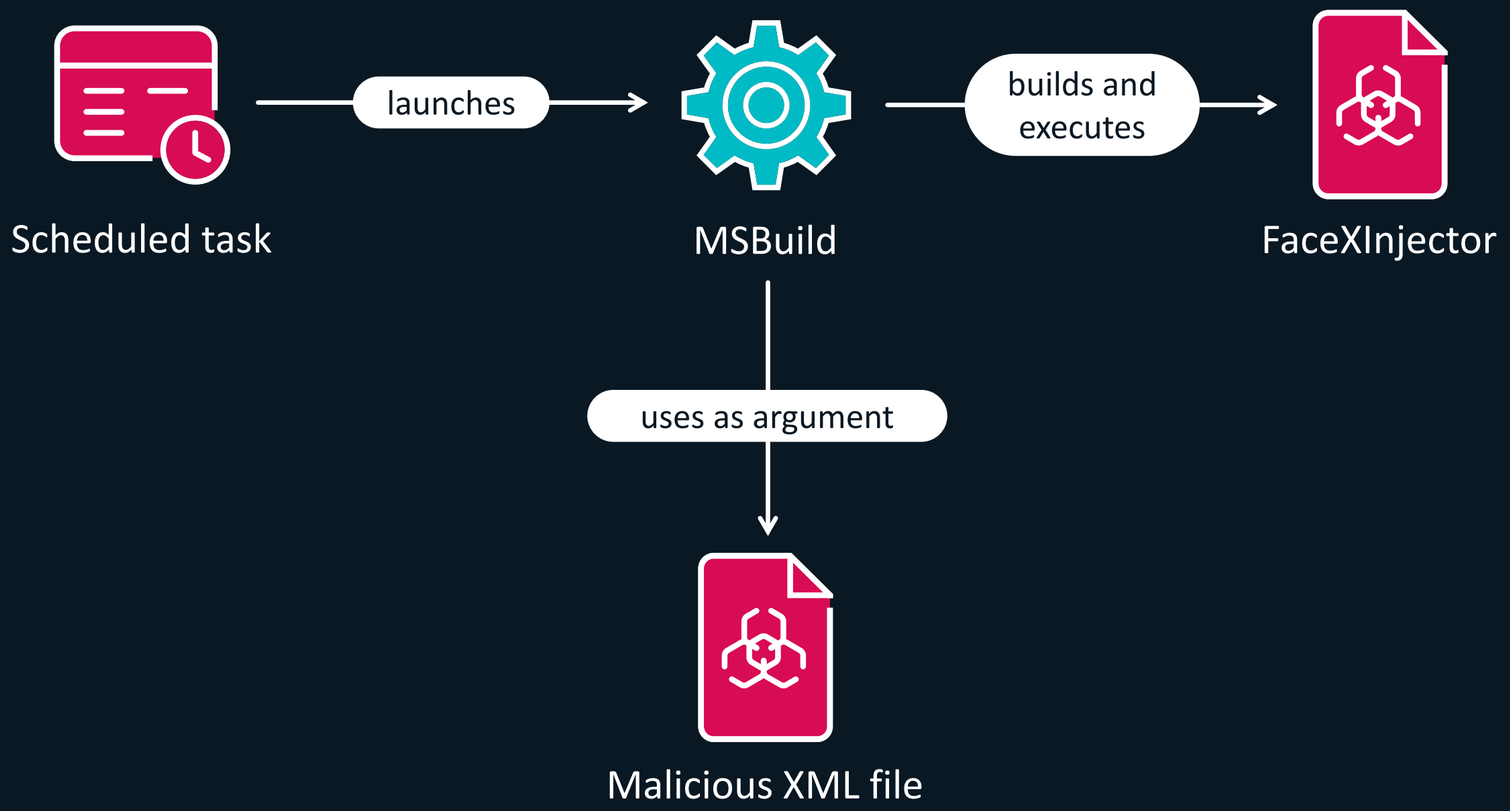
SHA-384(HKLM\Software\Microsoft\SQMClient\MachineId + hostname)

Execution chain - Installation

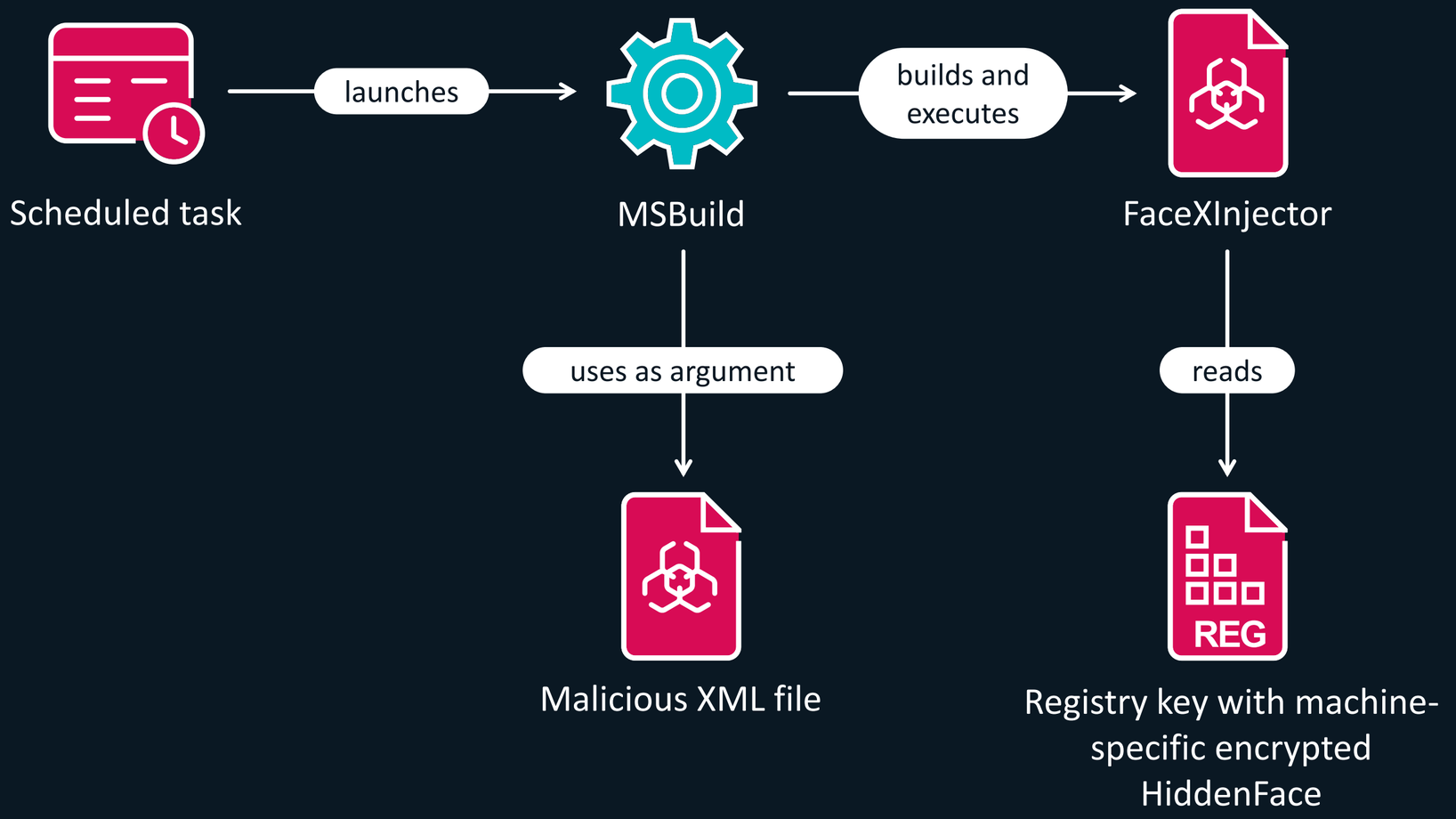


HKCU|HKLM\Software\License\{<16 hex characters>}

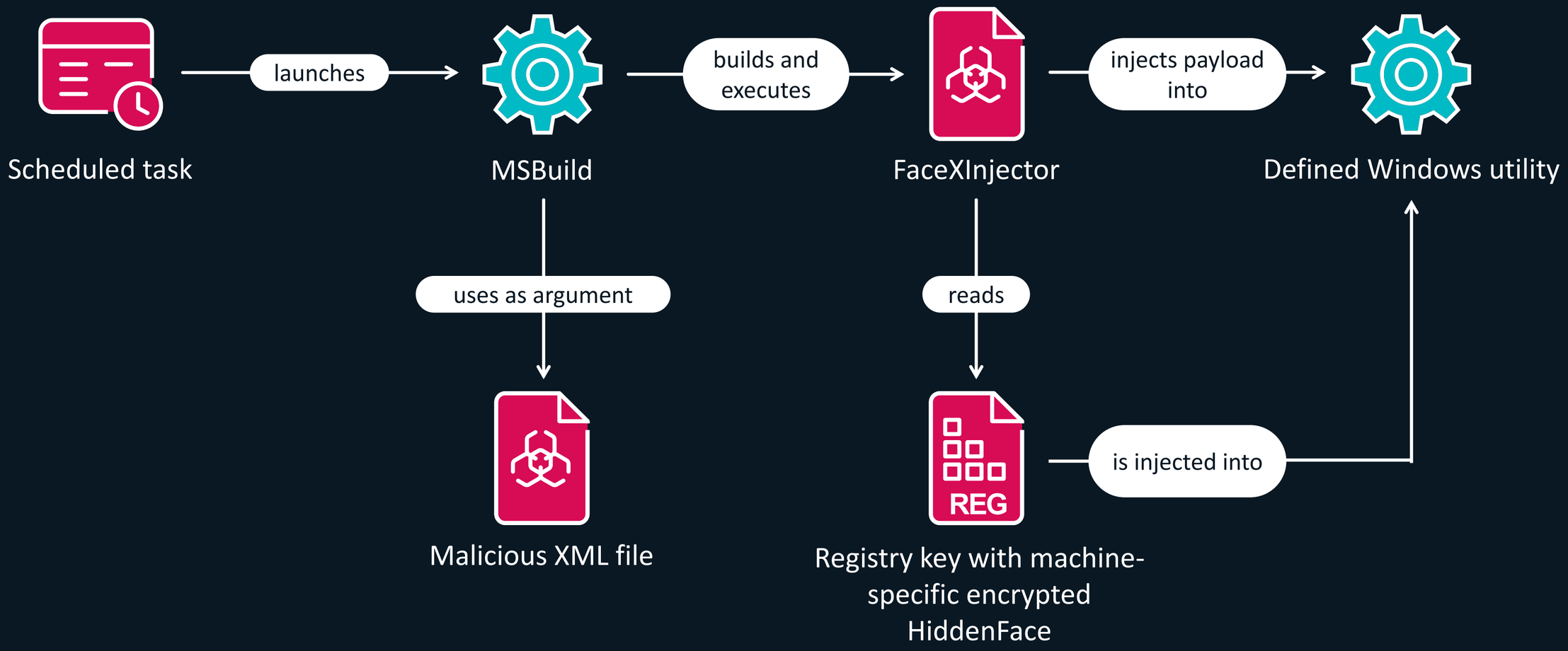
Execution chain - Injection



Execution chain - Injection



Execution chain - Injection



Example: perfmon.exe, wermgr.exe, or powercfg.exe

Startup

Startup

- **Dynamically resolves Windows APIs**
- **Performs few defensive actions**
  - Removes API resolution code
    → Memory dump is malformed
  - Restricts DLL loading to Microsoft-signed ones
  - Sleeps randomly in between 30 and 60 seconds
    → Likely to avoid behavioral analysis by sandbox or security solutions
  - Periodically checks running processes against a list of blacklisted applications
    - Debuggers, process monitors, network analysis tools …

Startup

- Creates mutex
  → Only one instance at a time
- Loads external modules
- Initializes internal framework
- Starts network communications

Modular System

Modular system

- **Core feature of HiddenFace**
- **Module:**
  Built-in functions or shellcode labeled by ID numbers
- **HiddenFace contains several built-in modules**
- **External modules are loaded from a file**
- **Additional modules can be sent by an operator**
  - Internal framework provided to a module received from a C&C server

External Modules

External modules

- **Stored in a file – AES-256-CBC-encrypted**
- **User-specific filename**
- **User-specific AES key and IV**
- **Algorithmically determined**
  - **Hostname** and **username** is used

Note: Most of the assets that are usually hardcoded in malware (e.g., encryption keys, filenames), are generated by HiddenFace.

External modules – Module Entry

Name	Description
Type	Module type (immediate, specific minute, etc.)
ID	ID to identify the module
Tag	(Optional) Additional label for the module
Time	Describes a specific time or a period; used for scheduled execution
Shellcode / Parameters	Contains either the module’s shellcode or parameters for a built-in module

External modules – Execution

- **Each module is executed based on its type**

Type	Description
Immediate	Immediately and only once
Specific minute	Specified minute every hour
Specific time	Specified time every day
Periodic	Every X minutes
Process monitor periodic	X minutes after the last check for running processes

Internal Framework

Internal framework

- **Provided to every module received from the C&C server**
- **Features:**
  - Access and modify external modules
  - Utilize internal memory storage
  - List running modules
  - Changes to the framework itself
- **Allows to create a tailored environment with needed capabilities**

Internal framework

- **Lookup function is used to obtain and execute desired function**

Function ID	Description
CCA8EB22C9E23C5D0577FC1F03060A5E	Add framework function
3D75B9B060499764C13527149E89D8DC	Remove framework function
CF05E89B7EAF28FE0DBF3B771B6C07B7	Write to memory storage
9BB2D76EDA1355D875D1D53DEEAA85B9	Read from memory storage
AC636E53FA3EC973F0E9535C8358C3E9	Remove data from memory storage
AC2BC61134888753316C1AC63DE465FE	Read external modules file
50515EF4F20DAA90B575DFFEAB4A97C0	Add module to external modules file
B5F39B21F0CC65CB1E3C75C6BFB7AB25	Write data to external modules file If no data is provided → file is deleted
1AA52A58C2C7B8E0079FF255D7294E70	Return list of running modules

Active Communication

Active communication

- Actively connects to a C&C server
- Works in sessions
- Hard-coded list of C&C URLs (templates)
- Uses domain generation algorithm (DGA)
- Uses custom protocol over TCP (on port 443)

Active communication – DGA

http://$n[].tw8sl.com:443/#180

Symbol	Description
$n	Variable to replace with a generated string (e.g., sofvgckcmyixg)
[]	Use hostname in the algorithm → Creates unique domain
#<num>	Increase domain's lifespan to <num> days

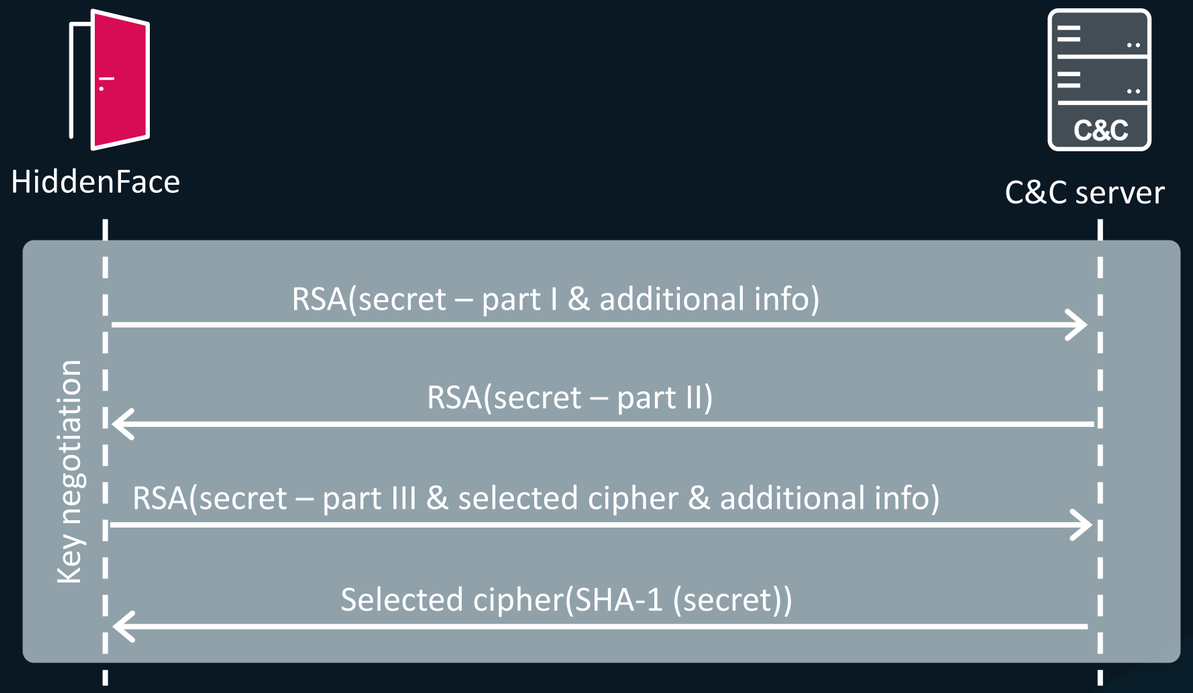
TrendMicro's example:

http://$d.hopto.org:443

Note: Some of the domains are under direct MirrorFace control.

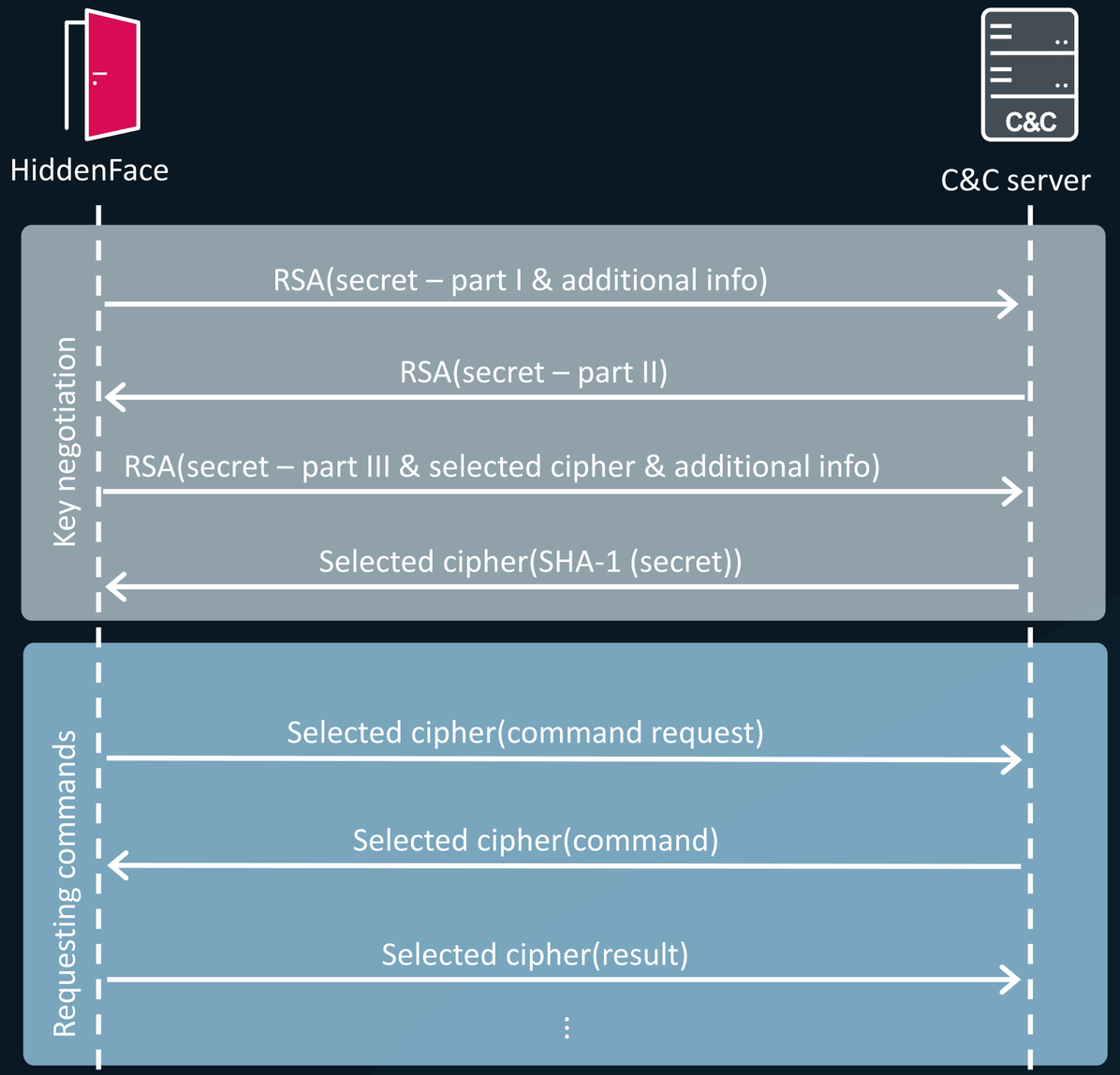
Active communication – Establishing a session

- **All messages exchanged are encrypted**
- **First messages are RSA-2048 encrypted**
  - To send collected information
  - To exchange key materials for a symmetric encryption cipher
- **Symmetric encryption cipher is used until the end of the session**
- **Cipher randomly selected by HiddenFace**
  - DES, 3DES, two-key 3DES
  - AES-CBC (128/192/256)
  - RC2, RC4



Active communication – Commands handling

- Commands executed by modules
- Server sends module ID and necessary data
- Module ID not found
  → Additional temporary module
  → Access to internal framework



Active communication – Commands

Function ID	Description
3B27D4EEFBC6137C23BD612DC7C4A817	Create a process
9AA5BB92E9D1CD212EFB0A5E9149B7E5	Write to a file
3C7660B04EE979FDC29CD7BBFDD05F23	Exfiltrate a file
12E2FC6C22B38788D8C1CC2768BD2C76	Read content from the file named **%SystemRoot%\System32\msra.tlb**
2D3D5C19A771A3606019C8ED1CD47FB5	Timestomp directory content

*Note: **msra.tlb** contains credentials collected by **MSRAStealer** – MirrorFace's publicly undescribed stealer.*

MSRAStealer

- **Passive credentials stealer**
- **Upon deployment registered as password filter and authentication package**
- **Password filter**
  - Legitimate use: Enforce password policy
  - MSRAStealer: collects credentials on a password change
- **Authentication package**
  - Legitimate use: Analyze logon data
  - MSRAStealer: collects credentials on user's logon
- **Collected credentials are dumped into msra.tlb – AES-256-CBC encrypted**
- **HiddenFace used to exfiltrate the credentials**

Passive Communication

Passive communication

- Hard-coded list of ports to listen on (e.g., 47000)
- Windows firewall reconfigured to allow communication
- Communication AES-128-CBC encrypted
- AES key and IV generated on:
  <year><hour (utc)><day><month>
- SHA-256 hash = AES key
- SHA-1 hash = AES IV

Passive communication – Commands

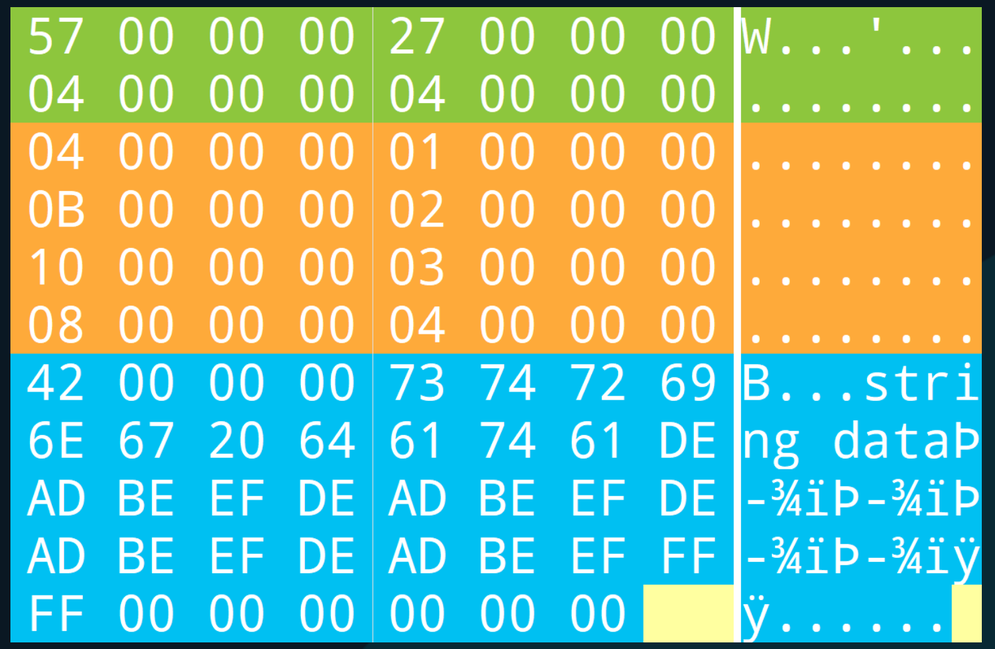
Command ID	Description
`0x0BE9`	Keep-Alive
`0x2359`	Create a process
`0x235A`	Exfiltrate a file
`0x235B`	Write to a file
`0x235C`	Set working directory
`0x235D`	Execute shellcode

Note: Execute shellcode – Shellcode is turned into a module first. Not added to the list of available modules and not provided with the access to the internal framework.

Data Structuring System

Data structuring system

- HiddenFace uses system to structure data
- For communication, but also internally
- Every structured data blob consists of:
  - Header
  - Metadata
  - Actual data



Data structuring system

Header

Offset	Size (bytes)	Description
0	4	Total size in bytes
4	4	Data section size in bytes
8	4	Number of metadata entries
12	4	Maximum possible number of metadata entries

Metadata

Offset	Size (bytes)	Description
0	4	Data size in bytes
4	4	Data type

Data structuring system – Data

- Consists of arbitrary content
- Heavily depends on the data’s purpose
- Every data item is categorized and defined in metadata
- HiddenFace distinguishes more than 80 data types

Example 1 – “Exfiltrate a file” command

Data type	Description
0x0BD1	Randomly generated data
0x03E8	Type of message Always set to 0xBE3, representing “Command request”
0x03EA	Receiving thread ID
0x0FA1	Module ID Always set to 3C7660B04EE979FDC29CD7BBFDD05F23, representing “Exfiltrate a file”
0x1389	(Optional) Request tag
0x138C	Item of unknown purpose
0x1772	Name of the file to exfiltrate
0x0BC2	(Optional) Base directory if the filename is relative
0x1774	(Optional) Known file size
0x1775	(Optional) Known last write time
0x1776	(Optional) Chunk information (file offsets)
0x1779	(Optional) Known SHA-1 hash of the file

Example 2 – Data passed internally to run a module

Data type	Description
`0x0FA1`	Module ID
`0x0FA2`	(Optional) Module's shellcode
`0x1389`	(Optional) Tag
`0x1390`	(Optional) Event name; to limit module's execution to one instance only
`0x138C`	Item of unknown purpose
`0x1398`	Internal framework's lookup function

Conclusion

Conclusion

- HiddenFace (NOOPDOOR) – Backdoor developed and exclusively used by MirrorFace
- The most complex malware in MirrorFace’s arsenal
- Developed with heavy focus on modularity
  → Can be tailored to current needs
- Utilizes other interesting techniques and mechanisms
  - DGA, data structuring approach, various anti-detection/-analysis techniques
- Protective execution chain shows HiddenFace is especially valuable to MirrorFace
- HiddenFace is a reasonably big project

Digital Security
Progress. Protected.

Thank you.

Note: IOCs after this slide.


dominik.breitenbacher@eset.com


@dbreitenbacher


dbreitenbacher

IOCs

IOCs – Files

SHA-1

41ACA6FCF8DF6599764DA638B2BAFDFD5E3EAD8B
512F3C8953AC079B57D1E13F3B8E97F99A054CE9
85E831EAC0AD5A308394BEB1CB7CE702C754FDB6
D96B05E516E9BB3E0AD8702D162440139E33D972

Scheduled Tasks

c:\windows\system32\tasks\microsoft\windows\user profile service\hiveupload
c:\windows\system32\tasks\microsoft\windows\wininet\cachetask
c:\windows\system32\tasks\microsoft\windows\shell\createobject
c:\windows\system32\tasks\microsoft\windows\workplace join\automatic-device-check
c:\windows\system32\tasks\microsoft\windows\media center\pbdadiscoveryw3

IOCs - Files

FaceXInjector XMLs

C:\Windows\system32\diskmgmt.config
C:\Windows\system32\MusNotification.xml
C:\Windows\system32\NetMgmtIF.xml
C:\Windows\system32\BrowserSettingSync.xml
C:\Windows\system32\BluetoothDesktopHandlers.xml

Encrypted HiddenFace

C:\Windows\system32\ActivationManager.tlb
C:\Windows\system32\ksetup.dat
C:\Windows\system32\LaunchWinApp.dat
C:\Windows\system32\win32k.tlb
C:\Windows\system32\Windows.Devices.Custom.dat

IOCs - Network

MirrorFace-controlled servers

5.180.44[.]139
202.182.118[.]157
207.148.97[.]235

C&C domains

vtfraznzdcns.myvnc[.]com
okzhfafcyumv.foeake[.]org
gjeyxinbutely.torefrog[.]com
hopekxpjyqloj.torefrog[.]com
kcxtdemxszlb.torefrog[.]com
lrsjvqxvzqua.torefrog[.]com
ogxzarazhzgu.torefrog[.]com
orufdqjuirceapb.torefrog[.]com
smfyuxgkeqiwgqw.torefrog[.]com