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

Fushatë Phishing
Email
Konfirmimi i regjistrimit të marrë

Versioni: 1.0
Datë: 12/07/2024

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 informacionet të gjetura dhe të vendosura në dispozicion për përgatitjen e 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 malinje 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, përditësuar, ose ndryshuar çfarëdo pjesë 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

Është evidentuar qarkullimi i një fushate të gjerë Phishing drejt disa institucioneve në Republikën e Shqipërisë nga aktorë keqdashës, ku është shfrytëzuar dërgimi i email nga domain legjitim **fsdksh[.]gov[.]al** nga ku subjekti i email është: **Konfirmimi i regjistrit të marrë** me dërgues jo legjitim: **regjistrimi[.]konfirmimi[@]fsdksh[.]gov[.]al,** si në figurën numër 1.

Gjithashtu në email ka bashkëlidhur një skedar me emërtimin **konfirmim.tar, si në figurën numër 2.**

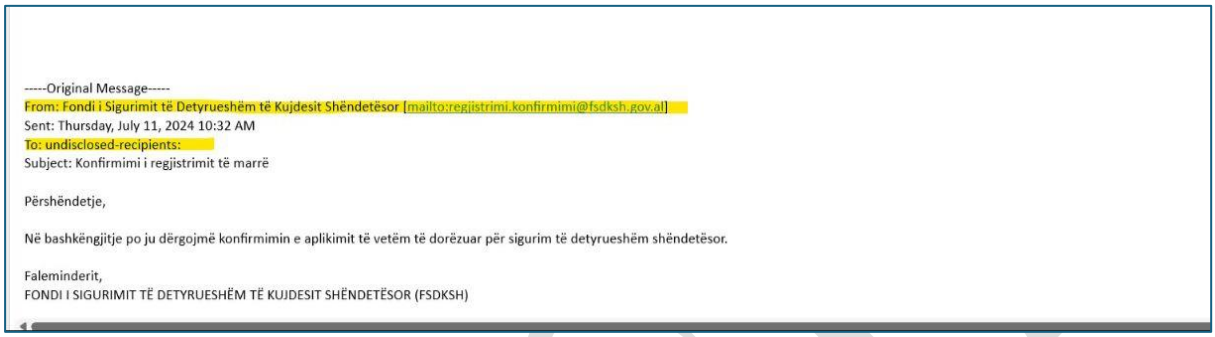

Figura 1: Përmbajtja e email Phishing

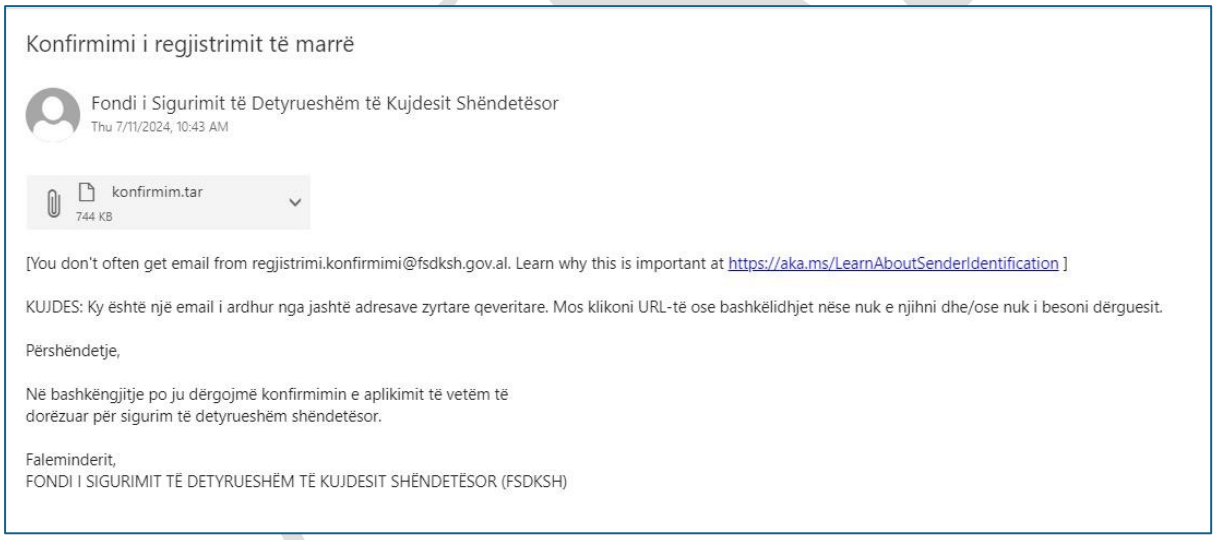

Figura 2: Skedari bashkëngjitur konfirmim.tar

Nga analiza fillestare e email header evidentohet se email është dërguar nga **posta[.]med[.]bg[.]ac[.]rs** me IP **147[.]91[.]120[.]120 ,** me AS 13092 dhe përket ***Akademska mreza Republike Srbije – AMRES***.

1	*	userid	posta.med.bg.ac.rs		7/11/2024 8:31:45 AM
2	4 minutes	posta.med.bg.ac.rs 147.91.120.120	pmg.med.bg.ac.rs	cipher ECDHE-RSA-AES256-GCM-SHA384 (256/256 bits)) (No client certificate requested)	7/11/2024 8:35:49 AM
3	7 seconds	pmg.med.bg.ac.rs 127.0.0.1	pmg.med.bg.ac.rs	ESMTP	7/11/2024 8:35:56 AM
4	2 seconds	pmg.med.bg.ac.rs 147.91.120.69	de225.cloudys.net	esmtps (TLS1.3) tls.TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (Exim 4.95) (envelope-from <regjistrimi.konfirmimi@fsdksh.gov.al>)	7/11/2024 8:35:58 AM

Figura 3: Detaje nga email header analysis

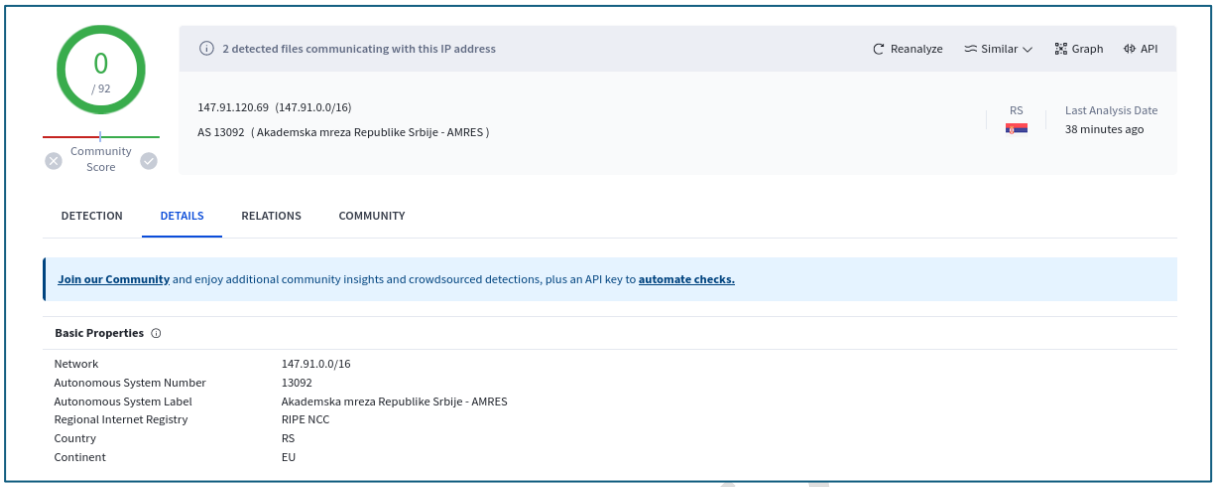

Figura 4: Informacione mbi IP nga ku është dërguar email

Nga analiza e thelluar, u evidentua se kjo fushatë është përhapur dhe në vendet e tjera të rajonit.

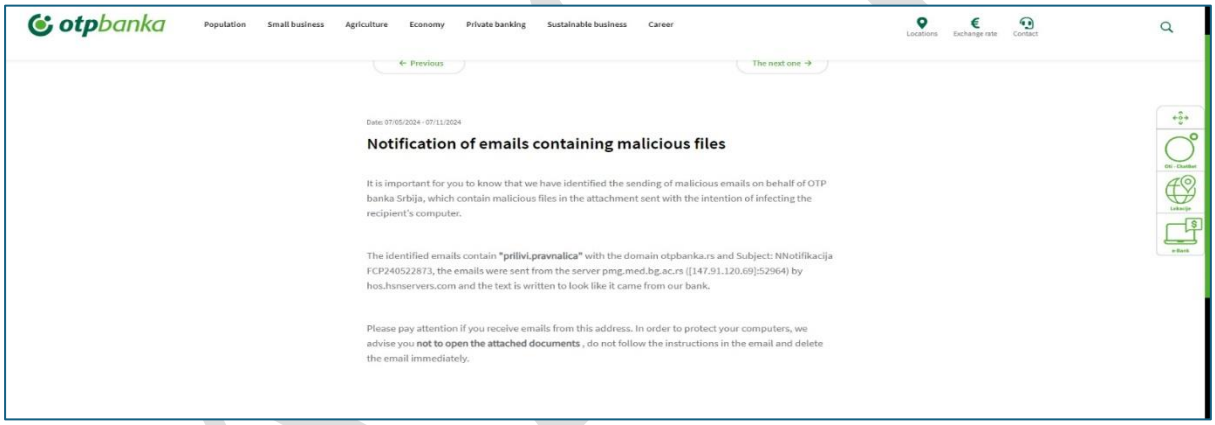

Figura 5: Raportim nga banka OTP , Serbi

Gjithashtu nga analiza e kryer e një email headeri, evidentohet se dërguesi është i njejtë, por marrësi me domain kroat.

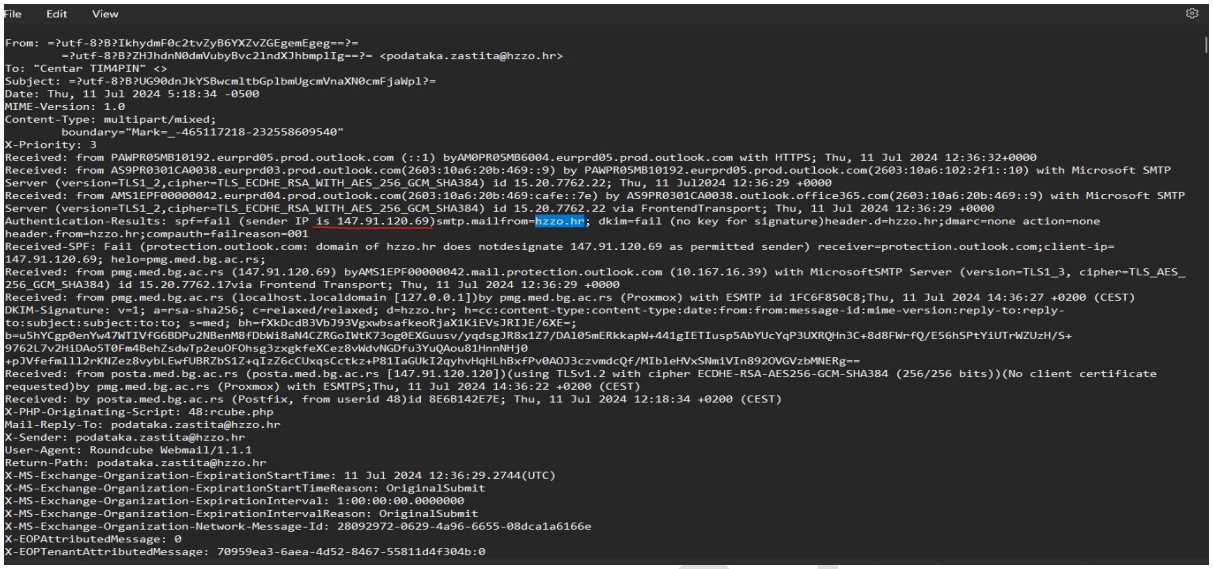

Figura 6: Analiza e një email header domain kroat

Analiza skedari konfirmim.tar

Skedari **konfirmim.cmd** nga vetë prapashtesa duket si një skedar që ekzekutohet nga ***command prompt*** por nëse e editojmë me **Notepad ++** evidentohet se skedari nis në pjesën e headerit me karakteret **MZ** që na jep një indicje që kemi të bëjmë me një skedar të ekzekutueshëm. Formati i skedarit është i krijuar nga frameworku i .**NET** me gjuhën e programimit **C#**. Skedari ka dhe një **signature** por e cila rezulton që nuk është e vlefshme.

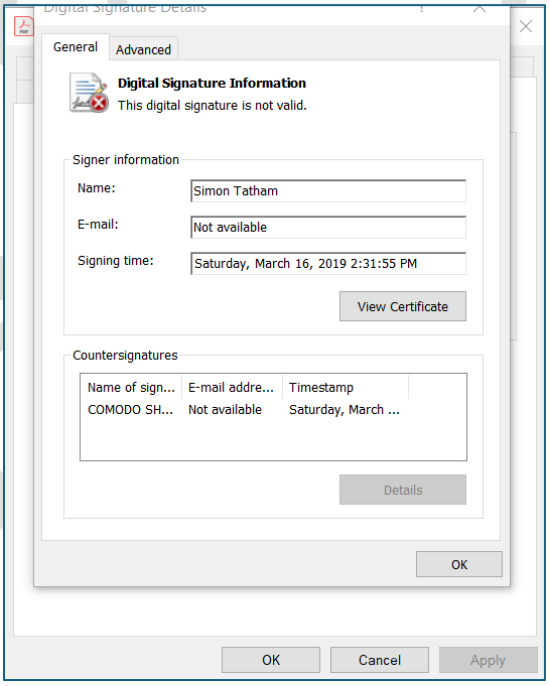

Figura 7: Signature e skedarit

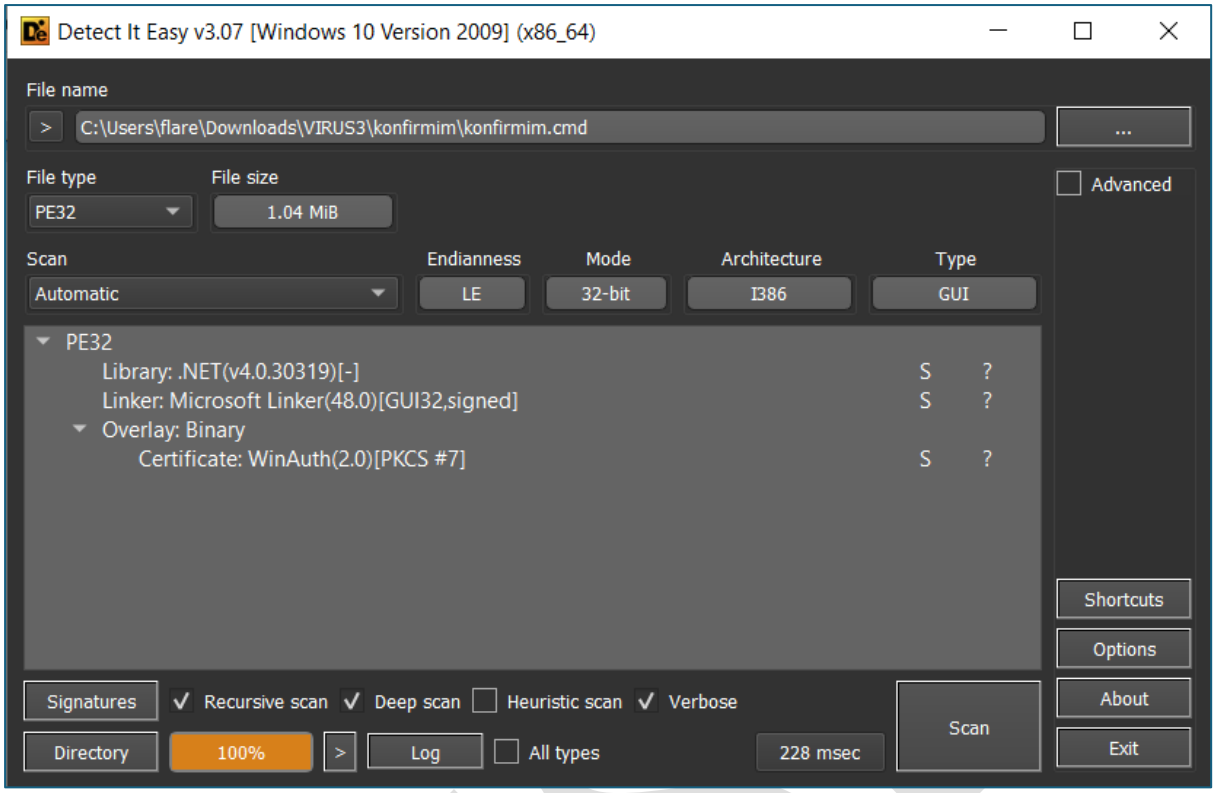

Figura 8: Frameworku i përdorur .NET

Duke qënë se frameworku i përdorur është **.NET**, mund të analizohet kodi duke e importuar si projekt. Pasi skedari importohet, evidentohet se automatikisht skedari merr emrin **PrIP** dhe ka disa **namespace**. Gjatë kërkimit evidentohen shumë rreshta kodi, të cilat në pamje të parë nuk na japin ndonjë informacion se kemi të bëjmë me një skedar keqdashës prandaj na duhet që të ndjekim funksionin **main()** hap pas hapi për të parë nëse kemi ndonjë skedar që ekzekutohet ose nëse kemi të bëjmë me **shellcode injection.** Projekti është i ndërtuar me **ASP.NET Windows Forms** dhe zakonisht kanë skedarë **resources** ku fshehin kodin keqdashës, përdorin algoritma tëpër kompleks dhe në **runtime** ekzekutojnë skedarët e fshehur. Evidentohet një nivel shumë i lartë fshehje, teknikë që aktorët keqdashës e përdorin për të anashkaluar antivirusin por edhe për ta bërë më të vështirë analizën.

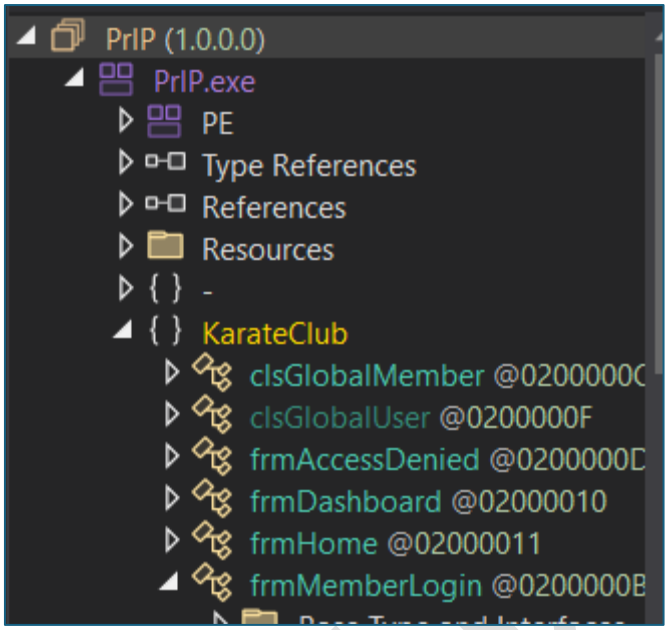

Figura 9: Skedari PrlP.exe

```
namespace KarateClub
{
    // Token: 0x0200000F RID: 15
    public class frmUserLogin : Form
    {
        // Token: 0x060000D5 RID: 213 RVA: 0x0000C6EC File Offset: 0x0000A8EC
        public frmUserLogin()
        {
            this.InitializeComponent();
        }

        // Token: 0x060000D6 RID: 214 RVA: 0x0000C734 File Offset: 0x0000A934
        private void btnClose_Click(object sender, EventArgs e)
        {
            for (;;)
            {
                IL_01:
                uint num = 1458116490U;
                for (;;)
                {
                    uint num2;
                    switch ((num2 = num ^ 1909541170U) % 3U)
                    {
                    case 1U:
                        frmUserLogin.\u200C\u200C\u202C\u206B\u200E\u206F\u200D\u206D\u202D\u200C\u202A\u200E\u202A\u206B\u206A\u206F
                          \u202B\u206C\u202C\u200D\u200B\u200F\u206C\u200C\u206C\u200F\u200C\u200C\u200F\u206C\u206B\u200E\u200E\u206E
                          \u202B\u202D\u200B\u200E\u202E\u200D\u202E();
                        num = (num2 * 1189533333U) ^ 1396536826U;
```

Figura 10: Kod i fshehur

Pasi bëjmë deobfuskimin në namespace **KarateClub** thirret funksioni **InitializeComponent().** Në këtë funksion gjithmonë nis startimi i variablave, llojeve të parametrave që do të ketë forma dhe detaje të tjera. Ajo që është interesante në këtë fazë është pjesa e kodit ku deklarohet një *byte array* e cila i merr vlerat nga **componentResourceManager** me parametrin **rs4.**
Më pas deklarohet një *breakpoint* i dytë me një parametër **string** që duket sikur është një çelës për pjesën e dekriptimit. Prandaj për të parë **outputin** duhet që ta ndjekim me anë të **breakpoints.** Vendosim një **breakpoint** në rreshtin e kodit ku kemi klasën **Assembly.** Kjo klasë është pjesë e frameworkut që mund t'i vendosim si parametër një **bytearray** që është një skedar i ekzekutueshëm dhe mund të ekzekutojmë funksionet në runtime. Për ta parë nëse është skedar i ekzekutueshëm ose jo evidentohet nëse hapim një nga **bytearray** që kalon si parameter dhe evidentohet vlerat **hex 4D 5A.**

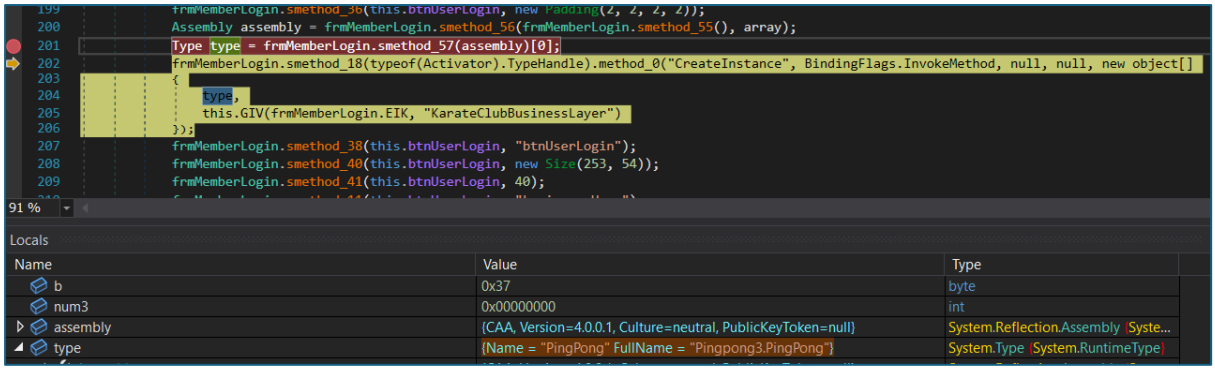

Figura 11: Skedari i ri i ekzekutueshëm

Nëse tentohet që të kalojmë në një rresht tjetër të kodit kjo gjë nuk është e mundur pasi skedari i ri nis ekzekutimin së bashku me proceset e tjera. Prandaj për të parë se çfarë ka ky skedar, këtë skedar e ruajmë në desktop dhe vijojmë me analizën. Skedari i ri përsëri përdor framework në ***.NET***.

Pasi importojmë skedarin **second-round** të krijuar nga ne, evidentohet se skedari në projekt merr automatikisht emrin **CA.dll** dhe ka një **namespace** me emrin **PingPong3** dhe dy namespace të tjera të enkoduara. Përsëri kodi ka fshehje, prandaj na duhet përsëri ta shohim për të kuptuar se cfare ndodh në këtë skedar. Ajo që është më interesante në këtë projekt është funksioni **Justy** e cila merr disa parametra **StringTypeInfo, String inputBlocksize** dhe **String EscapedIremotingFormatter**. Në kodin e funksionit evdentohet përdorimi i klasës **GzipStream**, pra nga **resources** merr si parametër **bg.tyn** e cila është një **bytearray** dhe për ta parë se çfarë po tentohet në këtë funksion përsëri vendosim një **breakpoint** dhe ajo se cfarë dallojmë është që kemi përsëri një skedar të ekzekutueshëm. Krijojmë një skedar të ri të ekzekutueshëm dhe tentojmë të thërrasim funksionin **Justy().**

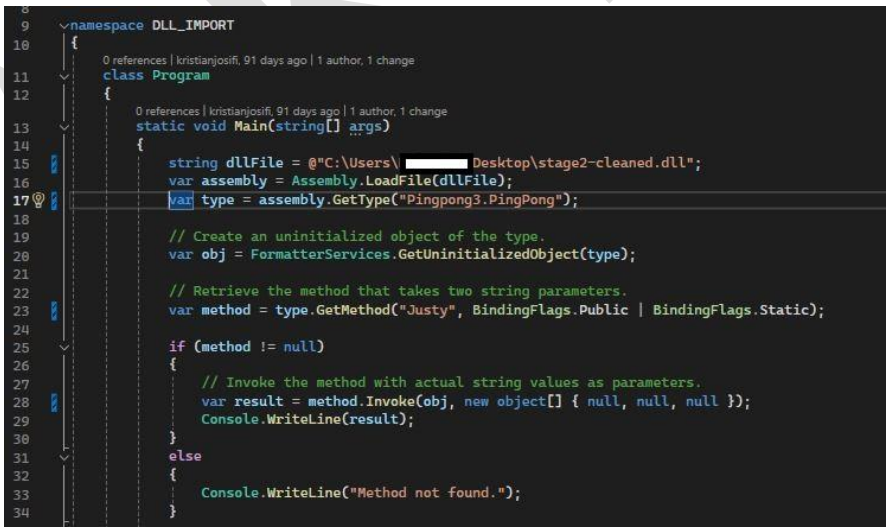

Figura 12: Thirrja e funksionit Justy()

Pasi e ndjekim me **breakpoint** dhe kapim vlerën e re kalojmë në raundin e 3-të dhe kontrollojmë formatin e skedarit dhe evidentohet se kemi të bëjmë me një **dll** me emrin **Gamma.dll** dhe ka si namespace **ReactionDiffusionLib.**

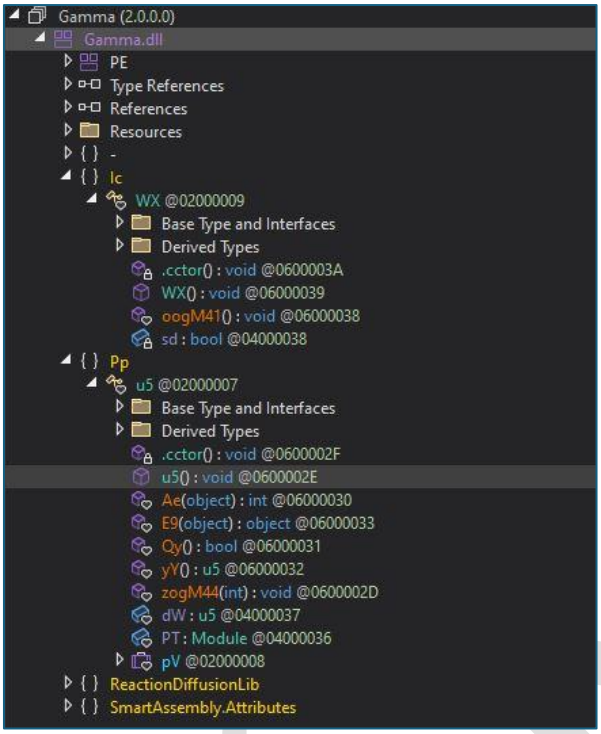

Figura 13: Gamma.dll

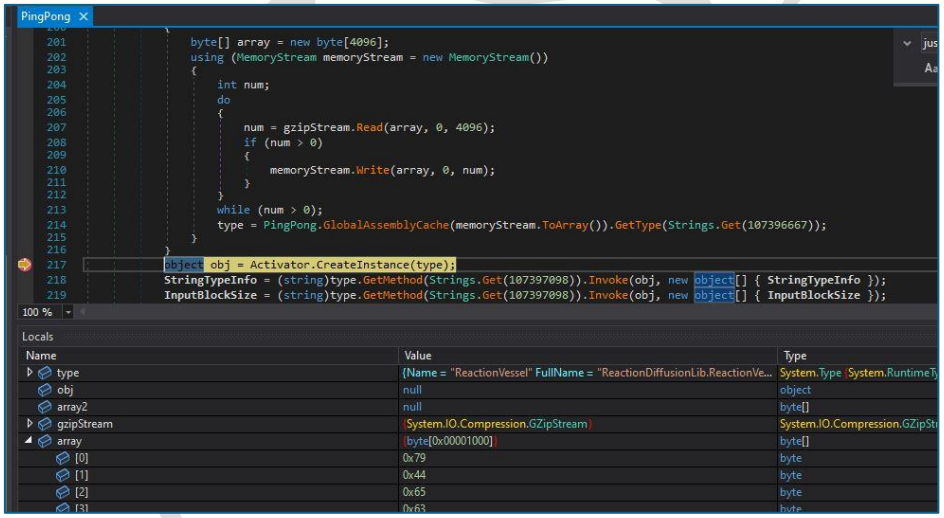

Figura 14: ReactionDiffusionLib

Në funksionin **Justy()** të **PingPong** tentohet të ekzekutohet funksioni **CasualitySource** e cila ndodhet në **ReactionDiffusionLib**. Pasi e importojmë këtë **dll** përsëri evidentohet se kemi të bëjmë me skedar në ***.NET.Bytearray*** e ruajmë në Desktop dhe e importojmë. Në namespace **ReactionDiffusionLib** evidentohet gjithashtu një funksion me emrin **CopyMemory** ku është importuar nga **dll kernel32.dll** legjitim i Windows. Ky funksion përdoret për të kopjuar një bllok memorie nga një vendndodhje në një tjetër. Zakonisht përdoret nëpër pikat e shfrytëzimit **Buffer Overflow.** Kjo na jep një ide e cila na jep informacion se kemi të bëjmë me **shellcode injection** e cila vendos në një pjesë të memories një hapësirë dhe aty vendos kodin keqdashës.

```
8
9   namespace ReactionDiffusionLib
10  {
11      // Token: 0x02000005 RID: 5
12      [DllImport("kernel32.dll")]
        0 references
13      public unsafe static extern int CopyMemory(void* pDest, void* pSrc, uint length);
14
15      // Token: 0x0600000A RID: 10 RVA: 0x000020D4 File Offset: 0x000002D4
        0 references
16      public static byte[] SearchResult(byte[] BinaryCompatibility, string Opcode)
17          {
18              byte[] bytes = Encoding.BigEndianUnicode.GetBytes(Opcode);
19              int num = (int)(BinaryCompatibility[BinaryCompatibility.Length - 1] ^ 112);
20              byte[] array2;
21              for (;;)
22              {
23                  int num2 = num;
24                  byte[] array = new byte[BinaryCompatibility.Length + 1];
25                  int num3 = 0;
26                  int num5;
27                  int num4 = (num5 = 0);
28                  int num6;
29                  if (num4 == 0)
30                  {
31                      if (4 != 0)
32                      {
33                          num6 = num4;
34                      }
```

Figura 15: Funksioni CopyMemory

Gjatë analizimit manual u arritën këto skedarë por përsëri nuk kemi skedarin keqdashës kjo për arsye të fshjehes dhe algoritmave tepër kompleks .

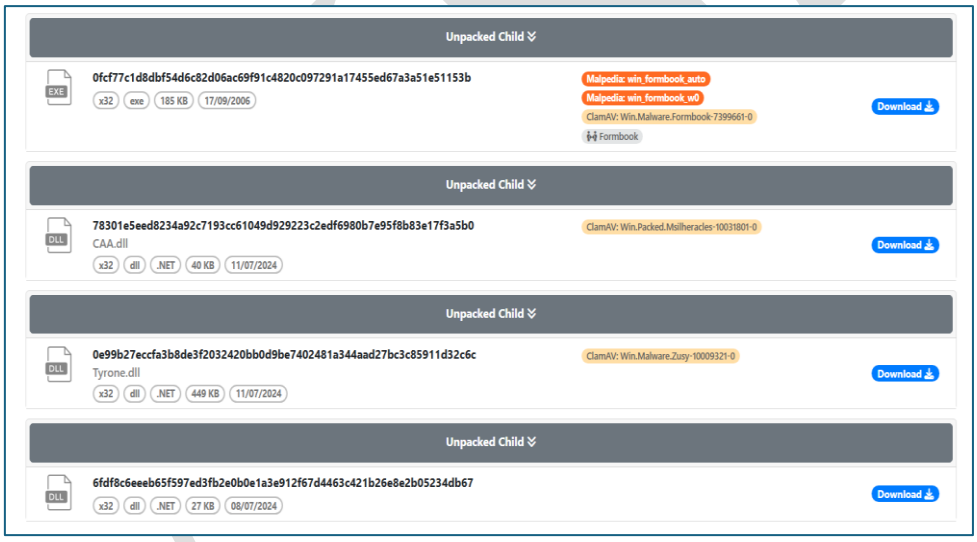

Figura 16: Amalizimi automatik

Ajo që evidentohet është se kemi dhe dy skedarë **Tyrone.dll** dhe një skedar të ri që është i paidentifikuar me emër, por automatikisht është identifikuar si skedari keqdashës **Formbook.** Ky skedar është një **spyware** i klasifikuar si vjedhës i të dhënave. **Formbook** është i klasifikuar si **MaaS** pra ***Malware as a Service***. Teknika që përdoret më shpesh njihet si **Process hollowing.** Skedari **Tyron.dll** është i shkruajtur në **.NET** dhe shërben si ndihmës për skedarin keqdashës të **Formbook**, pasi nëpërmjet saj skedari arrin qëndrueshmërinë në kompjuterin e infektuar, ekzekutimin e procesit në formë të fshehtë . Një shembull i tillë është ne funksionin e fshehur më poshtë ku tentohet të nisi një **proces hidden.**

```
public static void BX0Q1DpcnZ(string string_0)
{
    Process process = pQ6V9TTs6bVoyOtX7O.ELccDVdqtM();
    ProcessStartInfo processStartInfo = pQ6V9TTs6bVoyOtX7O.POPc7thsxE();
    pQ6V9TTs6bVoyOtX7O.vqCcX7RJG5(processStartInfo, <Module>.smethod_5<string>(343917627U));
    pQ6V9TTs6bVoyOtX7O.LfMctlZDCy(processStartInfo, string_0);
    pQ6V9TTs6bVoyOtX7O.faccocnwjD(processStartInfo, ProcessWindowStyle.Hidden);
    pQ6V9TTs6bVoyOtX7O.s3kcQb53En(process, processStartInfo);
    Process process2 = process;
    pQ6V9TTs6bVoyOtX7O.UJJcc5GVIA(process2);
}
```

Figura 17: Proces i fshehur

Nëse kontrollojmë **strings** të skedarit **Formbook** evidentohet vargje karakteresh të cilat jan pa kuptim dhe janë të fshehura. Prandaj tentojmë ***deobfuskimin*** e tyre. Ajo që evidentohet janë **strings** si **Login Data**, user, pass madje dhe funksionet si **InternetOpenA,InternetconnectA** etj, për të bërë të mundur lidhjen me serverin **Command And Control (C2).**

```
encrypted_key
Local State
Pass
User
Internet Explorer\IntelliForms\Storage2
\Opera Software\Opera Stable\Login Data
Pass
Name
__Vault
rc.ini
Iexplor
Outlook Recovery
rv.ini
ri.ini
Password
2016
image/png
image/jpeg
im.jpeg
is.jpeg
Unknown
Host
User-Agent:
urlmon.dll
User-Agent:
www.
login
auth
pass
user
Sniff from:\t
Server:\t
```

Figura 18: Strings të deobfuskuara nga Formbook

Indikatorët e kompromet imit

- **Domain**
  posta[.]med[.]bg[.]ac[.]rs
  pmg[.]med[.]bg[.]ac[.]rs
  sukhiclothing[.]com (Command And Control) C2 Server
  almouranipainting[.]com
  cataloguia[.]shop
  zaparielectric[.]com
  whcqsc[.]com
  ioco[.]in
  aduredmond[.]com
  vavada611a[.]fun
  humtivers[.]com
  jewellerytml[.]com
  mcapitalparticipacoes[.]com
  inhlcq[.]shop
  solanamall[.]xyz
  moviepropgroup[.]com
  thegenesis[.]ltd
  cyberxdefend[.]com
  skinbykoco[.]com
  entermintlead[.]com
  honestaireviews[.]com
  wyclhj7gqfustzp[.]buzz
  w937xb[.]com
  bakuusa[.]online
  sabong-web[.]com
  52cg2[.]club
  jasonnutter[.]golf
  odbet555[.]app
  vipmotoryatkiralama[.]com
  auravibeslighting[.]com
  pulsesautos[.]com
  imdcaam[.]com
  vivaness[.]club
  bovverbadges[.]com
  giaydonghai[.]online
  aditi-jobs[.]com
  numericalsemantics[.]com
  shoprazorlaser[.]com
  lovedacademy[.]com
  gets-lnds[.]io
  teyo293[.]xyz
  banditsolana[.]com
  delivery-jobs-76134[.]bond
  ppp5716[.]buzz
  zjmeterial[.]com
  de-ponqk[.]top
  bntyr76rhg[.]top
  servicepmgtl[.]world

nailtimelocust[.]top
paperappa[.]com
80sos[.]com
daysofbetting[.]com
slaytheday[.]fun
travauxdefou[.]com
bx2zyg[.]com
thecoxnews[.]com
qriskaq[.]com
top-dao[.]com
krstockly1[.]shop
roiwholesale[.]com
pajero777ads[.]click
twistedrubytx[.]com
thesovreignkingdomofmaui[.]info
cataclysmicgamingapparel[.]com
verxop[.]xyz
xn--kwra1023b[.]com
winterclairee[.]com

- **URL**
  http[:]//www[.]imdcaam[.]com/dn03/?ARrxNN=URhw1ZxcIs5da1k+vMTqZFryLoAICCIR37JNPpiCybm1EsRHUECMqVHccUGvhI4Ma8f5og==&LXTpg=0v1DUfwX4XoDi6ip

- **IP**
  147[.]91[.]120[.]69

- **HASH**

909903ADDA36DCFC103A5260990C7CA1F47B4644672F2D94446C7E6E86F25A35	konfirmimi.exe
384610B76013F3B0D420033AC5AFEE88EA2516F08B1C652A261BB42ECE4C7BCC	konfirmimi.tar
0E99B27ECCFA3B8DE3F2032420BB0D9BE7402481A344AAD27BC3C85911D32C6C	tyrone.dll
0FCF77C1D8DBF54D6C82D06AC69F91C4820C097291A17455ED67A3A51E51153B	formbook malaware
6FDF8C6EEEB65F597ED3FB2E0B0E1A3E912F67D4463C421B26E8E2B05234DB67	gamma.dll
78301E5EED8234A92C7193CC61049D929223C2EDF6980B7E95F8B83E17F3A5B0	caa.dll

Teknikat e MITRE ATT&CK

Tactic	Technique	ID	Description
Initial Access	Spearphishing Attachment	T1566.001	Formbook is typically delivered via spearphishing emails with malicious attachments.
Execution	User Execution	T1204.001	Execution occurs when a user opens the malicious attachment, often disguised as a legitimate file (e.g., a document or image).

Tactic	Technique	ID	Description
Persistence	Registry Run Keys / Startup Folder	T1547.001	Formbook adds registry keys or files to the startup folder to ensure persistence across system reboots.
Privilege Escalation	Process Injection	T1055	Injects code into other processes to gain higher privileges or evade detection.
Privilege Escalation	Process Hollowing	T1055.012	Formbook uses process hollowing to inject malicious code into the address space of a legitimate process to evade detection and escalate privileges.
Defense Evasion	Obfuscated Files or Information	T1027	Uses various obfuscation techniques to evade detection by security software.
Credential Access	Credential Dumping	T1003	Extracts credentials from browsers and other applications to steal sensitive information.
Discovery	System Information Discovery	T1082	Gathers information about the system, such as OS version, hardware details, and running processes.
Collection	Input Capture	T1056.001	Captures user inputs, including keystrokes, to steal sensitive information like passwords and other credentials.
Exfiltration	Exfiltration Over C2 Channel	T1041	Exfiltrates collected data over the Command and Control (C2) channel.
Command and Control	Web Service	T1102.001	Uses legitimate web services for command and control communications to blend in with normal traffic and avoid detection.

Rekomandime

AKSK rekomandon:

- Bllokimin e menjëhershëm të Indikatorëve të Kompromentimit, 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".