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

Fushatë Phishing nga Muddy Water
Analizë skedari - AteraAgent.exe

Versioni: 1.0
Datë: 10.07.2024

PËRMBAJTJA:

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 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 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.

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 Phishing nga aktorë keqdashës, ku është shfrytëzuar përdorimi i një aplikacioni legjitim dhe duke modifikuar kodin burimor, kryejnë veprime keqdashëse në kompjuterat dhe sistemet e infektuara. Analiza e skedarit **Atera**, lidhet me sulmet e fundit Phishing të lidhura me grupin Iranian, **MuddyWater**.

Skedari me vlerë **HASH sha256**:
55AF6A90AC8863F27B3FCAA416A0F1E4FF02FB42AA46A7274C6B76AA000AACC2 është një skedar në formatin **.msi** (***Microsoft Windows Installer***) i cili ekzekutohet nga vetë përdoruesi dhe instalohet në kompjuterin e tij. Për formate skedarësh të tillë ndryshojmë prapashtesën e skedarit nga **.msi** dhe i shtojmë prapashtesën **.7z** dhe tentojmë ta ekstraktojmë.

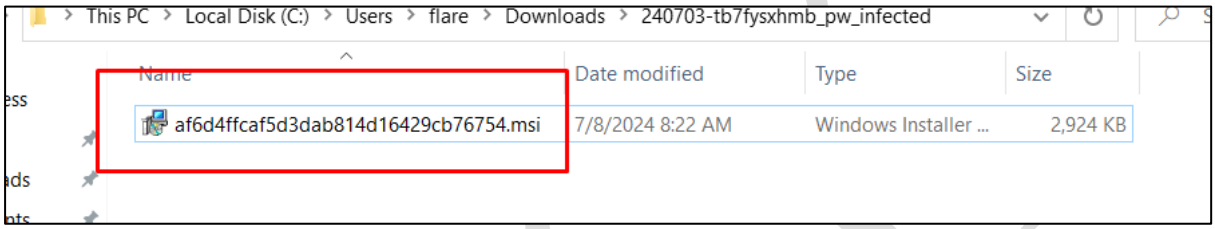


Figura 1. Skedari .msi.

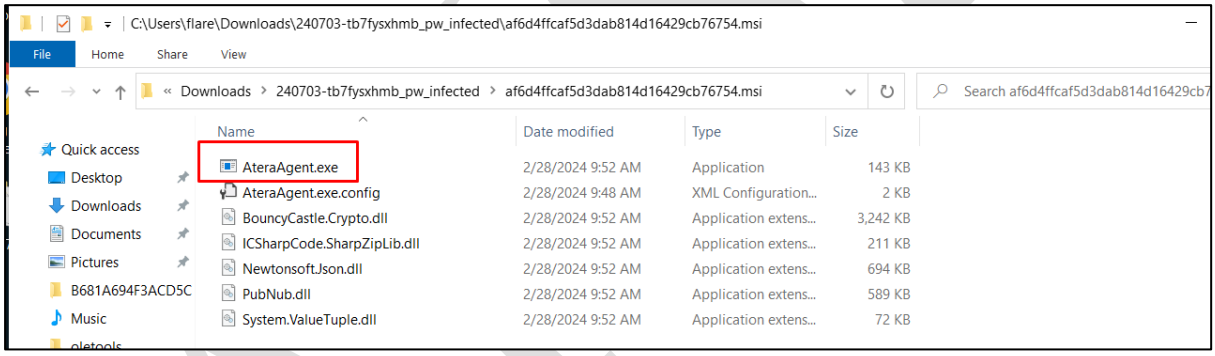


Figura 2. AteraAgent.exe.

Nga faza e ekstraktimit rezulton se kemi disa nën skedarë nga ku evidentohet **AteraAgent.exe**. Nga vetë emri kuptohet se kemi të bëjmë me një **RAT (Remote Access Trojan)**. Gjatë analizës evidentohet se skedari është i krijuar në **.NET** dhe me gjuhë programimi **C#.**

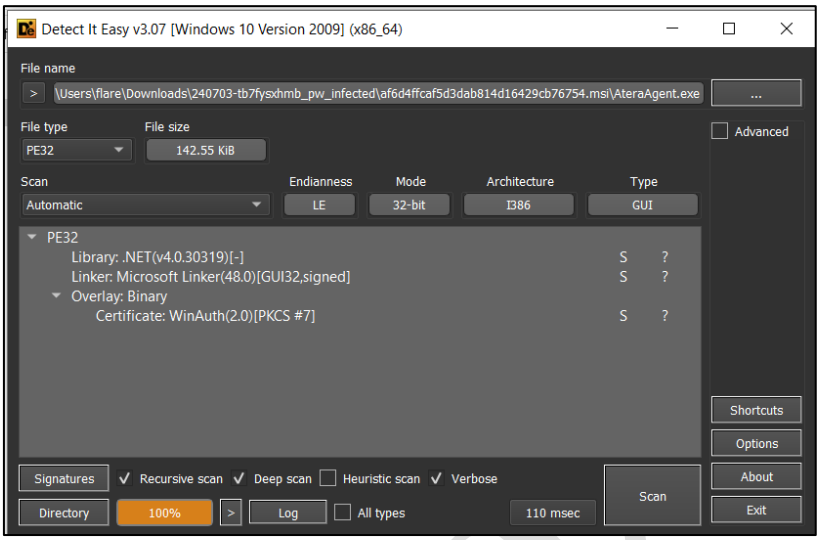


Figura 3. .NET framework.

Provojmë ta importojmë si projekt këtë skedar dhe evidentohet se funksionalitetet që ky **RAT** ofron janë nga më të ndryshmet.

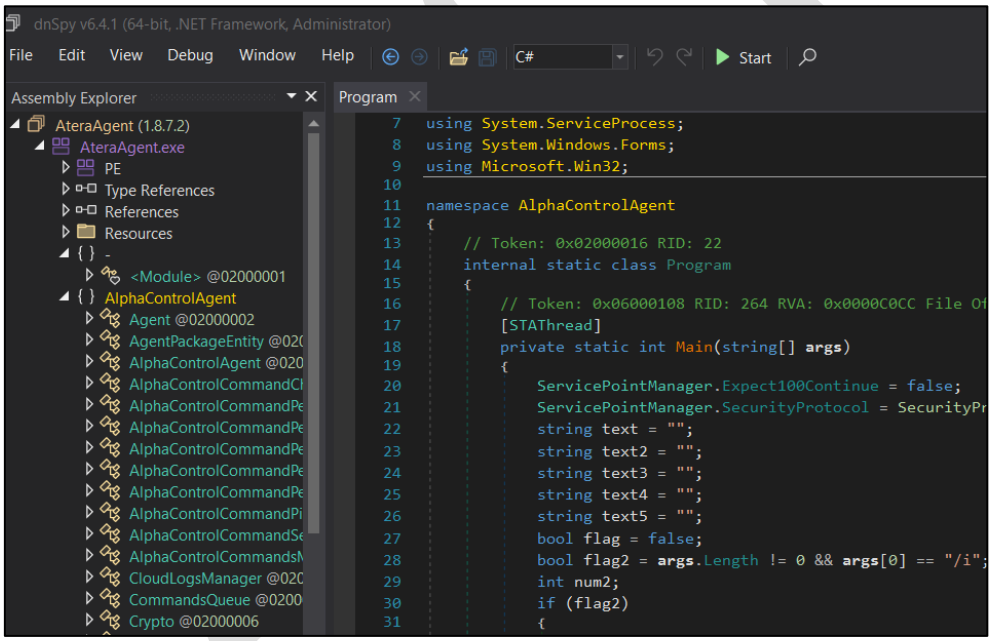


Figura 4. Funksionalitetet e AteraAgent.exe.

Funksioni **main()** është funksioni kryesor që nis në klasën Program**.cs**. Aty nis implementimi kryesor dhe nisja e klasës **Agent** e cila duket së përdor **Design Pattern të Singleton.** Në funksionin **main** nis pjesa e konfigurimeve nga ku lexon si parametra argumentat që i kalon agjentit gjatë ekzekutimit, kjo bëhet me qëllim që të bëhet lidhja me serverin remote *command and control* (C2). Kjo realizohet nëpërmjet një emaili të kompromentuar, id parametër të agjentit,etj.

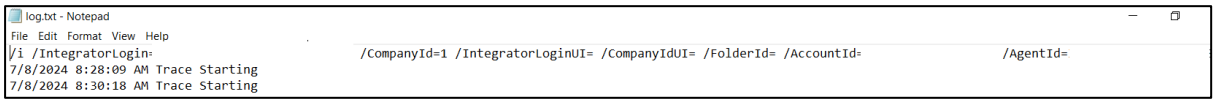


Figura 5. parametrat që kalohen si argumenta.

Në klasën **Agent.cs** evidentohet funksioni **SendQueryCommandsRequestAndHandleReceivedCommands** që kalon si parametër në ndërtuesin e klasës **new ThreadStart.** Nëse kontrollojmë nënfunksionet e këtij funksioni evidentohet thirrja e funksionit **SendQueryCommandsRequest** që ka të implementuar logjikën e marrjes dhe dërgimit të komandave Remote.

```
this.recurring_packages_timer = new System.Threading.Timer(new TimerCallback(this.ExecuteRecurringPackages), null,
this.recurring_packages_retrieval_timer = new System.Threading.Timer(new TimerCallback(this.RetrieveRecurringPackag
this._commandsQueue.Load();
SystemFreezeListener.SystemDefrosted += this.OnSystemDefrosted;
Thread thread2 = new Thread(new ThreadStart(this.SendQueryCommandsRequestAndHandleReceivedCommands));
thread2.Start();
this.InitializePubnub();
this.StartPubnubSubscribe();
this.SendAgentStartingCommand();
Thread thread3 = new Thread(new ThreadStart(this.CallSendQueryCommandRequestBackupLoop));
thread3.Start();
Thread thread4 = new Thread(new ThreadStart(this.RunEnqueuedCommandsLoop));
thread4.Start();
});
thread.Start();
```

Figura 6. Funksioni SendQuerycommandRequests.

Gjatë analizës evidentohet funksionaliteti i marrjes së informacionit mbi kompjuterin e kompromet uar nëpërmjet funksionit **GetMachineName().** Nëse shikojmë implementimin e funksionit realizohet nëpërmjet ***System.Environment.MachineName.***

```
private static string GetMachineName()
{
    string text = "";
    try
    {
        text = System.Environment.MachineName;
    }
    catch
    {
    }
    bool flag = !string.IsNullOrEmpty(text);
    string text2;
    if (flag)
    {
        text2 = text;
    }
    else
    {
        try
        {
            text = SystemInformation.ComputerName;
        }
        catch
        {
        }
```

Figura 7. Funksioni GetMachineName().

Marrja e informacionit mbi sistemin e operimit realizohet nëpërmjet funksionit **GetOS()** ku përdoret klasa **ManagementClass** dhe i kalon si parametër stringun **“Win32_OperatingSystem”.**

```
private static string GetOS()
{
    string text;
    try
    {
        ManagementClass managementClass = new ManagementClass("Win32_OperatingSystem");
        using (ManagementObjectCollection.ManagementObjectEnumerator enumerator = managementClass.GetInstances().GetEnumerator(
        {
            if (enumerator.MoveNext())
            {
                ManagementObject managementObject = (ManagementObject)enumerator.Current;
                return managementObject["Name"].ToString();
            }
        }
        text = null;
    }
    catch
    {
        text = null;
    }
    return text;
}
```

Figura 8. Funksioni GetOS().

Gjatë analizës së kodit evidentohet funksioni **SetEnvironmentInRegistry**, gjatë instalimit të agjentit krijohet një subfolder me path **“SOFTWARE\ATERANETWORKS\AlphaAgent”**. Kjo bëhet me qëllim që të realizohet vazhdueshmëria e skedarit keqdashës.

```
1 reference
private static void SetEnvironmentInRegistry(DateTime expiryDateTime, string environmentNameValue)
{
    RegistryKey registryKey = null;
    try
    {
        registryKey = Registry.LocalMachine.CreateSubKey("SOFTWARE\\ATERA Networks\\AlphaAgent");
        bool flag = registryKey == null;
        if (flag)
        {
            throw new Exception("Key AlphaAgent not found in Registry");
        }
        registryKey.SetValue("EnvironmentExpiry", expiryDateTime.ToString(new DateTimeFormatInfo()), RegistryValueKind.String);
        registryKey.SetValue("EnvironmentName", environmentNameValue, RegistryValueKind.String);
    }
    catch (Exception ex)
    {
        Agent._logger.ErrorException("Failed to update environment in Registry", ex);
    }
    finally
    {
        bool flag2 = registryKey != null;
        if (flag2)
```

Figura 9. Funksioni SetEnvironmentInRegistry

Kemi dhe një kontroll i cili tenton të ekzekutojë një process **silent** ku realizohet nëpërmjet klasës **ProcessStartInfo** dhe në pjesën ndërtuese merr parametrat përkatës.

```
if (flag2)
{
    ProcessStartInfo processStartInfo = new ProcessStartInfo
    {
        FileName = text4,
        Arguments = "/SILENT",
        UseShellExecute = false,
        CreateNoWindow = true,
        WorkingDirectory = AppDomain.CurrentDomain.BaseDirectory,
        ErrorDialog = false
    };
    Process process = Process.Start(processStartInfo);
    bool flag3 = process == null;
    if (flag3)
    {
        throw new Exception("Process.Start() returned null");
    }
```

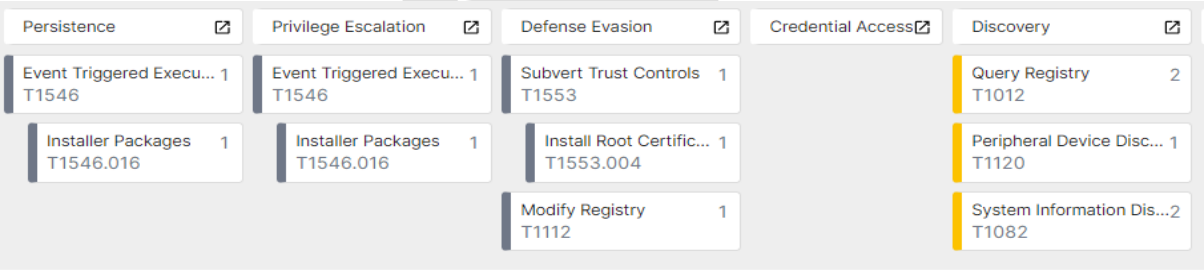
Figura 10. Krijimi i një procesi silent

Indikatorët e kompromitimit

HASH

55af6a90ac8863f27b3fcaa416a0f1e4ff02fb42aa46a7274c6b76aa000aacc2
8fbd374d4659efdc5b5a57ff4168236aeaab6dae4af6b92d99ac28e05f04e5c1
c152b0c74d704054e2962e2a6198195dc96b4de92f97d8243519e7dcbfca4bd3

Teknikat e MITRE ATT&CK



Rekomandime

AKSK rekomandon:

- Bllokimin e menjëhershëm të Indikatorëve të Kompromitimit, 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 menaxhimin 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".