In the Shadows: Vawtrak Aims to Get Stealthier by adding New Data Cloaking | Proofpoint US

By October 01, 2015 Darien Huss and Matthew Mesa

Published: 2015-09-30 · Archived: 2026-04-05 13:08:28 UTC

In what is likely to be a short-lived cessation in Dridex campaigns while the criminal proponents behind that malware scramble to find a new delivery channel, it appears as though other malware purveyors may be positioning themselves to take additional market share of the lucrative crimeware arena. One recent development saw Vawtrak, previously a second-tier banking and information stealing trojan, emerge with new capabilities -- most notably new methods for data encoding and changes to C2 communication that appear to be an attempt to improve on the malware's detection evasion.

Part I: Attack Vectors and Infiltration

Before it can leverage its new capabilities, Vawtrak must be delivered to a target. While attachment-based phishing remains a leading delivery vector, Proofpoint also observed exploit kit-based attacks (aka "drive-by downloads") delivering this updated variant starting in late September.

Attachment-Based Phishing

Proofpoint observed several high volume email campaigns delivering the new Vawtrak variant.The emails claimed to have attachments that were faxes (Figure 1), subpoenas, price lists or financial reports in order to persuade the user to click on and open the attachment. The attachments contained macros from a service known as Xbagging or Bartalex[1], which in turn downloaded the Pony malware dropper from a remote internet site. Pony then downloaded and executed the Vawtrak payload.

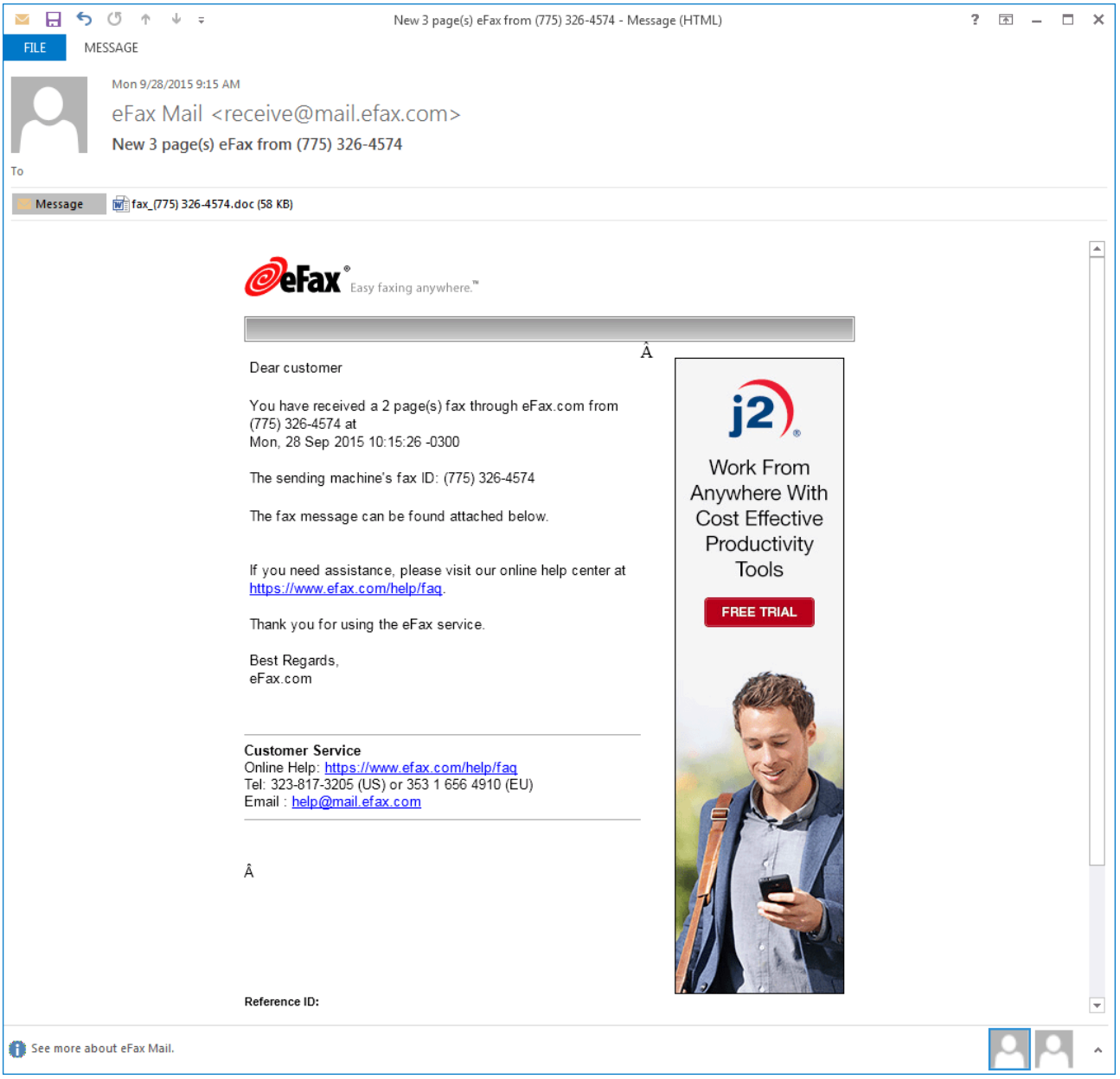

Figure 1. Email lure

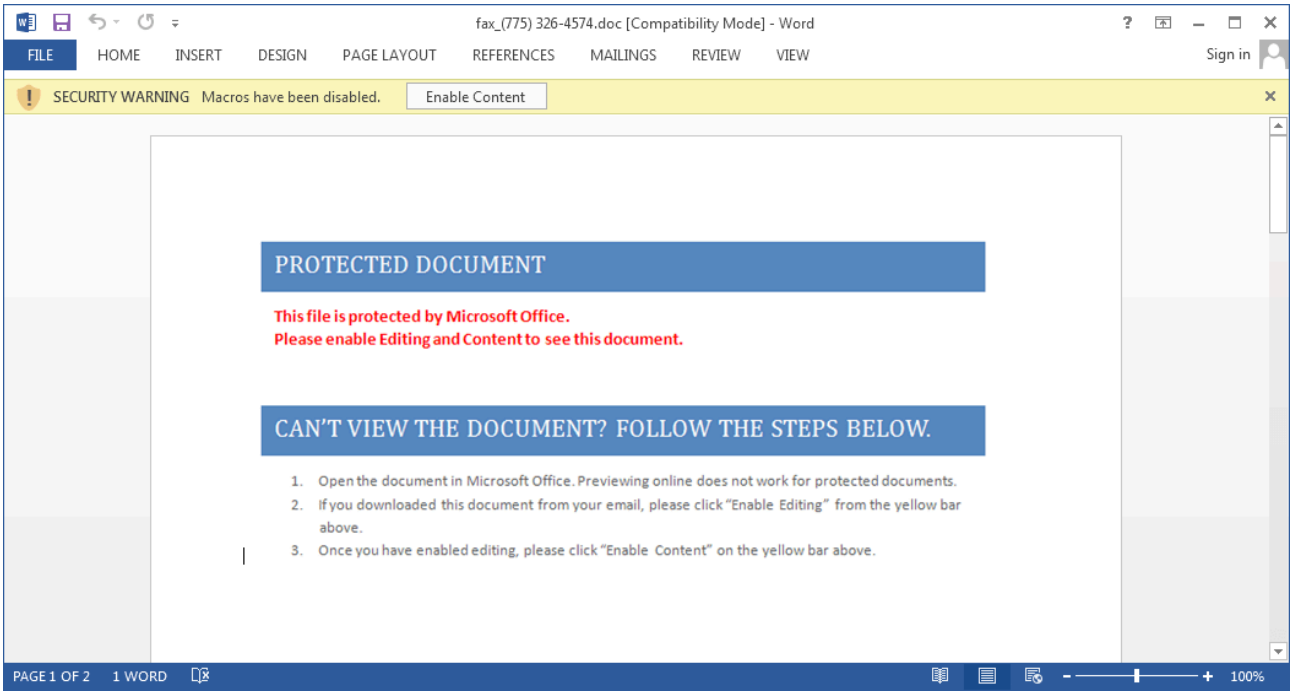

Figure 2. Malicious attachment

Proofpoint researchers have observed the following subjects and lures being used in the latest campaign to distribute the new variant of Vawtrak:

Date	**Email Subject**	**Lure**
Sept 17	Re: Re: defamation lawsuit	Subpoena
Sept 22	Re: Re: Re:	Financial report
Sept 23	Re: New offer	Price list
Sept 24	New 2 page(s) eFax from <phone number>	Fax
Sept 28	New 3 page(s) eFax from <phone number>	Fax
Sept 29	You have 1 new eVoice Voicemail (Callback: <phone number>)	Voice message

Sept 30	You have 1 new eVoice Voicemail (Callback: <phone number>)	Voice message

Exploit Kit-Based Delivery

Proofpoint researchers have also observed this Vawtrak variant distributed through the Angler exploit kit. In the example shown in Figure 3, we observed a malicious TDS which led to an instance of Angler EK downloading Bedep. Bedep then performed its usual routines (e.g., created a hidden desktop that engaged in ad-fraud via browsing and other botnet attacks), but also downloaded Vawtrak.

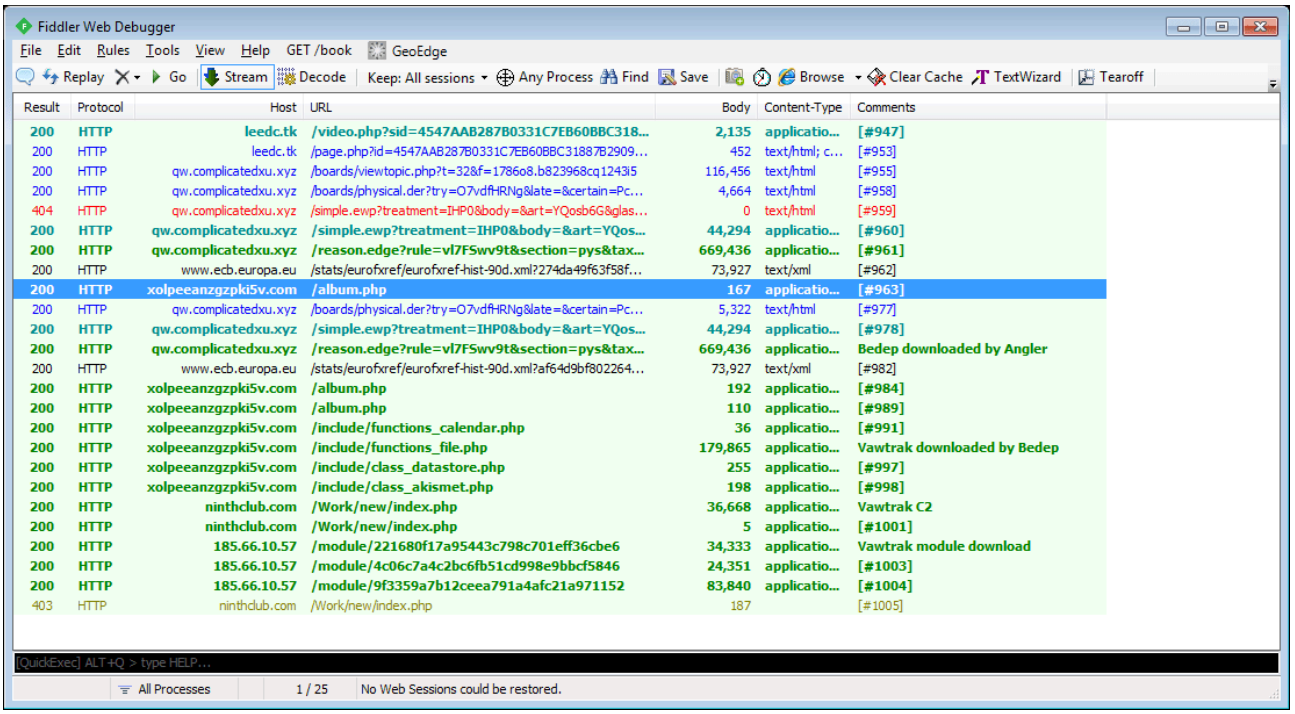

Figure 3. Angler EK distributing Bedep -> Vawtrak

Part II: Command & Control and Data Exfiltration -- Vawtrak gets an upgrade

Understanding communication to C2 and malware configuration files can play an important role in organizations' detection of malware and remediation thereof, enabling better assessment of the damage malware might have inflicted. The latest observed variant of Vawtrak has incorporated definitive changes designed to further thwart such efforts by defenders.

Modified Encoding and Encryption

As previous research has described[2,3,4], Vawtrak has historically used an encoding method resembling a Vernam cipher to hide configuration files, suspicious strings and mask data exfiltrated to C2. In its latest incarnation, Vawtrak still uses a linear congruential generator (LCG) fed by a pseudorandom number generator (PRNG) to produce the key used to encrypt the data; however, the utilized PRNG function is now changed. The code below is a simplified version of the new PRNG algorithm written in Python:

```
def prng(seed):
    return ((seed * 0x41C64E6D) + 0x3039 ) & 0xFFFFFFFF
```

String Encoding

String encoding utilizes LCG fed by the new PRNG algorithm, while the generated keys are then subtracted from each ciphertext byte to generate the plaintext string. The first DWORD in the encoded string is used as the seed, while that same value is XOR'ed against the second DWORD to calculate the size of the encoded string. The encoded string begins at the position after the second DWORD. Most suspicious strings in the unpacked Vawtrak DLL are encoded using this method.

HTTP Beacons

The HTTP traffic that is generated by Vawtrak to exfiltrate data to C2 is correspondingly changed, now drastically different in appearance as well as functionality. Figure 4 shows an example of the HTTP traffic generated by Vawtrak during the initial check-in with C2.

Figure 4. Vawtrak HTTP C2 check-in

PHPSESSID is used to transport an encoded RC4 key and additional data. The first 4-bytes of the decoded Cookie are used to RC4 encrypt the data contained in the POST's client body. This variant of Vawtrak utilizes a binary structure for most of the data transmitted to C2, as can be seen in the decrypted network traffic in Figure 5. This same encoding method is used during the exfiltration of credentials.

```
Extracted encoded PHPSESSID Cookie: 5C8EC19E61666B717F808B939EAAB7C5

Decoded PHPSESSID Cookie:
00000000: 5C D2 9C C1 03 00 00 00  07 00 02 00 00 00 00 00  \...............

RC4 key:
00000000: 5C D2 9C C1                                      \...

Decrypted HTTP client body:
00000000: 00 08 00 00 00 00 00 5B  00 00 00 01 0F 00 31 32  .......[......12
00000010: 37 2E 30 2E 30 2E 31 3A  38 38 38 38 00 02 08 00  7.0.0.1:8888....
00000020: 52 4F 42 55 53 54 50 43  03 09 00 52 4F 42 55 53  ROBUSTPC...ROBUS
00000030: 54 49 4E 43 04 10 00 02  01 00 02 06 01 01 01 00  TINC............
00000040: 01 B1 1D 00 00 00 00                              .......
```

Figure 5. Decrypted Vawtrak HTTP check-in

Depending on what is being exfiltrated, LZMAT is sometimes used to compress the exfiltrated data prior to encryption. Figure 6 shows the decrypted HTTP client body during an American Express attempted login, but the data is not yet in its plaintext format as it has been compressed with LZMAT. Figure 7 illustrates the observed plaintext data after it has been decompressed using LZMAT.

```
Decrypted HTTP client body:
00000000: 99 00 00 00 68 00 74 74  70 73 3A 2F 2F 6F 00 6E  ....h.ttps://o.n
00000010: 6C 69 6E 65 2E 61 6D 00  65 72 69 63 61 6E 65 78  line.am.ericanex
00000020: 00 70 72 65 73 73 2E 63  6F 00 6D 2F 6D 79 63 61  .press.co.m/myca
00000030: 2F 61 00 63 63 74 6D 67  6D 74 2F 30 75 73 20 A0  /a.cctmgmt/0us .
00000040: 01 6F 75 6E 74 00 73 75  6D 6D 61 72 79 2E 00 64  .ount.summary..d
00000050: 6F 3F 72 65 71 75 65 00  73 74 5F 74 79 70 65 3D  o?reque.st_type=
00000060: 00 61 75 74 68 72 65 67  5F A8 68 11 44 35 53 54  .authreg_.h.D5ST
00000070: 63 32 F7 06 20 47 57 46  F6 95 E6 46 06 50 86 D7  c2.. GWF...F.P..
00000080: 03 63 92 E6 16 16 60 D7  D3 56 E6 56 F7 85 2A 67  .c....`..V.V..*g
00000090: 74 64 C2 0A 0D 0A 00                              td.....
```

Figure 6. Decrypted but still compressed Vawtrak HTTP exfiltrated data

```
00000000  68 74 74 70 73 3a 2f 2f  6f 6e 6c 69 6e 65 2e 61  |https://online.a|
00000010  6d 65 72 69 63 61 6e 65  78 70 72 65 73 73 2e 63  |mericanexpress.c|
00000020  6f 6d 2f 6d 79 63 61 2f  61 63 63 74 6d 67 6d 74  |om/myca/acctmgmt|
00000030  2f 75 73 2f 6d 79 61 63  63 6f 75 6e 74 73 75 6d  |/us/myaccountsum|
00000040  6d 61 72 79 2e 64 6f 3f  72 65 71 75 65 73 74 5f  |mary.do?request_|
00000050  74 79 70 65 3d 61 75 74  68 72 65 67 5f 61 63 63  |type=authreg_acc|
00000060  74 41 63 63 6f 75 6e 74  53 75 6d 6d 61 72 79 26  |tAccountSummary&|
00000070  73 6f 72 74 65 64 5f 69  6e 64 65 78 3d 30 26 69  |sorted_index=0&i|
00000080  6e 61 76 3d 6d 65 6e 75  5f 6d 79 61 63 63 74 5f  |nav=menu_myacct_|
00000090  61 63 63 74 73 75 6d 0d  0a                       |acctsum..|
00000099
```

Figure 7. Decompressed data from Figure 3

Config encoding

This variant of Vawtrak typically receives a raw (no encoding) binary blob of data immediately after the initial check-in. This blob contains a binary structure that may contain separate segments, including but not limited to an encoded configuration, URLs for retrieving additional modules, and a URL for retrieving an updated version of itself.

To decode the configuration file, Vawtrak first uses the exact same decoding method that is used to decode suspicious strings. Next, the configuration file is decompressed using LZMAT. After decompression, the configuration is contained in a binary data structure which contains several further encoded configuration segments. Figure 8 depicts the purpose of the first few bytes of this structure.

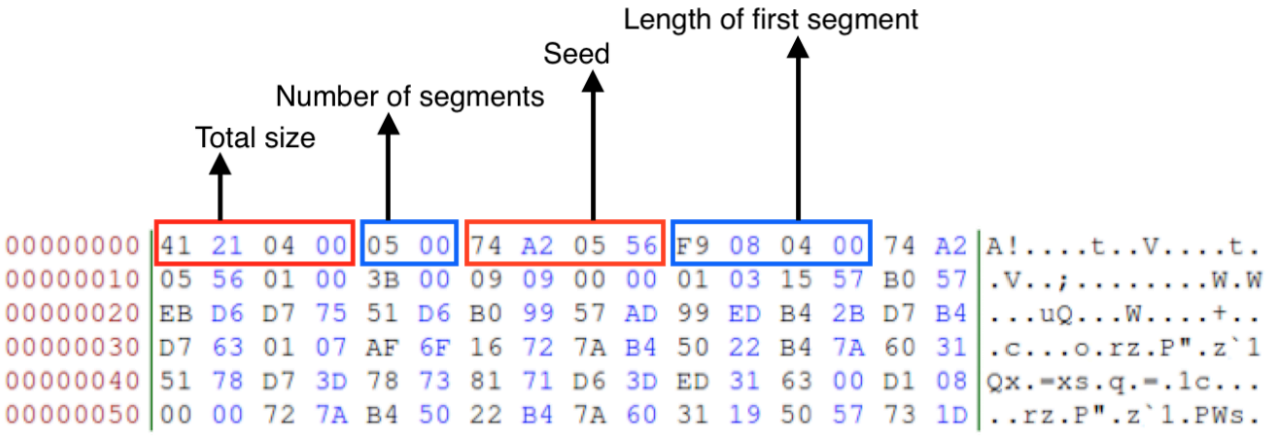

Figure 8. Decompressed encoded Vawtrak configuration

The next and last layer of encoding that Vawtrak uses to protect its configuration is a simple substitution cipher, where the S-box is created using the same PRNG algorithm. Each individual inject, target URL, etc., is contained in its own structure and decoded separately.

Storing Configuration

In addition to decoding the configuration immediately upon receiving it, Vawtrak also stores the encoded configuration in the registry after adding an additional layer of encoding. First, the seed (first DWORD) is XOR'ed against the VolumeSerialNumber of the drive which contains the result from the Windows API function GetTempPath. Next, the entire encoded configuration is encoded further using the addition LCG algorithm. This value is then stored in the registry using an encoded key. We have observed this variant using the following registry keys denoting the configuration file:

-"#0"
-"#1"

However, these keys are first encoded in a similar fashion as older variants, but are first XOR'ed against the VolumeSerialNumber. Figure 9 contains a screenshot of stored Vawtrak information, while the highlighted key contains the encoded configuration.

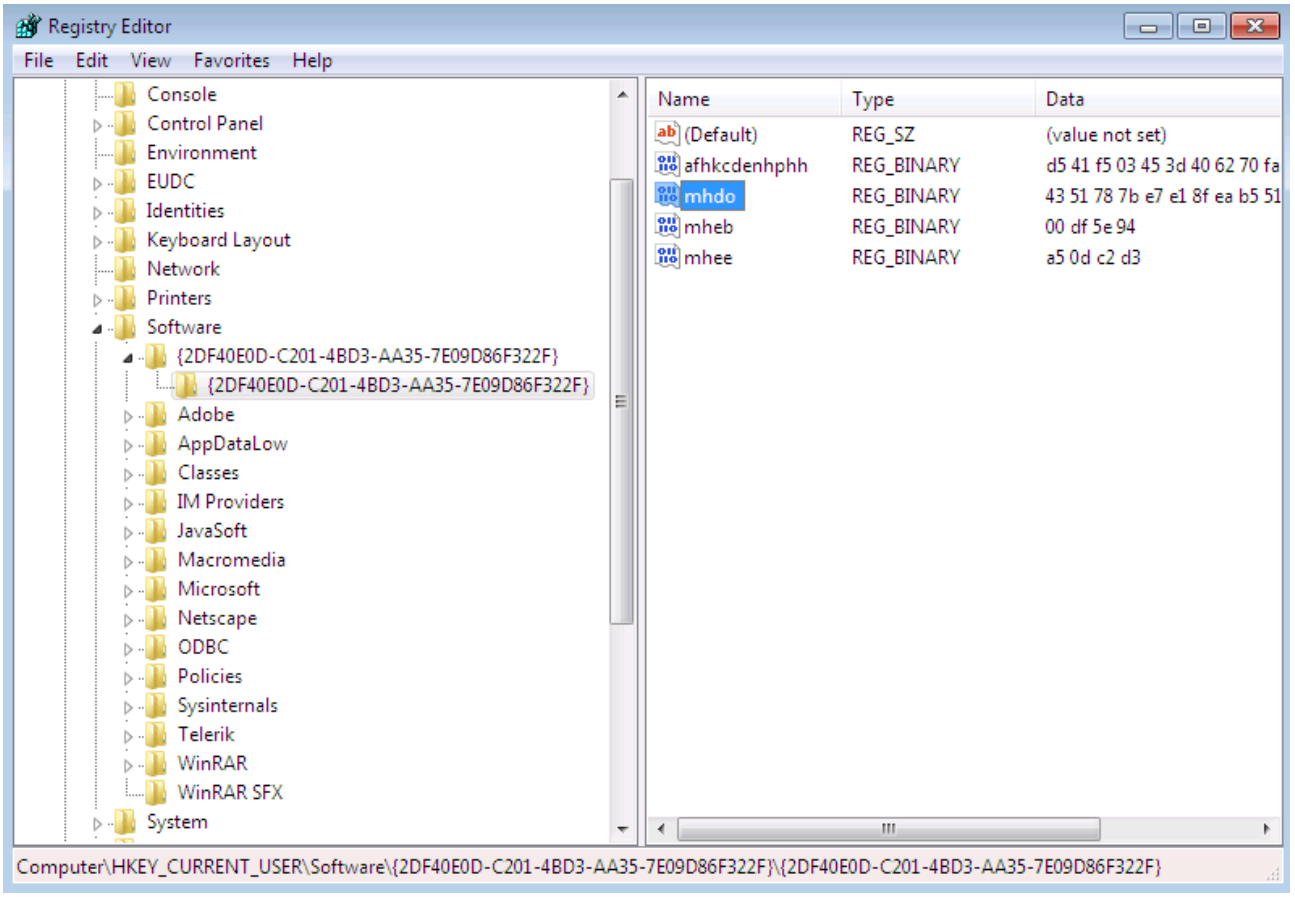

Figure 9. Vawtrak information stored in the Registry

Retrieving Modules

As mentioned in the previous section, when Vawtrak receives its configuration from C2 it may also receive a list of URIs that are targets to additional modules. Vawtrak spawns a new thread for each module it should retrieve. The module is first received in an encoded state, which is decoded using the same subtraction LCG algorithm described in the previous sections. The decoded module contains a RSA signature at the beginning, which is used to verify the integrity of the compressed module. In each of decompressed "modules" we have analyzed, they have all contained a x86 and x64 version of the module. Each module may then be decompressed separately depending on the architecture of the infected machine. So far we have only observed the following modules:

[hxxp://185.66.10[.]57/module/9f3359a7b12ceea791a4afc21a971152 -> injecter_32.dll / injecter_64.dll]
[hxxp://185.66.10[.]57/module/4c06c7a4c2bc6fb51cd998e9bbcf5846 -> dg_32.dll / dg_64.dll]
[hxxp://185.66.10[.]57/module/221680f17a95443c798c701eff36cbe6 -> pony_32.dll / pony_64.dll]

Retrieving Updates

As previously mentioned, in addition to retrieving modules Vawtrak may also receive a URL target pointing to an "update." The update is contained in a binary data structure similar to the modules' structure; however, the seed is contained in the second DWORD instead of the first. A RSA signature is then contained in the next 0x80 bytes, while the encoded update is contained in the remaining bytes following the signature. The update may be decoded using the same LCG subtraction algorithm. The URLs containing updated DLLs can be found in Appendix A.

Web Injects and Stolen Data

Vawtrak still functions similarly to previous versions regarding stealing data and web injects. In the sample's configuration that we analyzed, several financial institutions and online services were targeted such as Amazon.com. For several of the organizations, a custom web inject was tailored to steal additional information beyond just login credentials. Victims attempting to login to Amazon were presented with the following credit card form (Figure 10) via Vawtrak's web inject mechanism.

Figure 10. Web injected fake Amazon.com credit card form

Should a victim fill out this form, the credit card data along with their Amazon login credentials will be sent to the botnet operators via the method described in the *HTTP Beacons* section. Figure 11 shows the decrypted output that is delivered to the malware's C2.

```
Decrypted HTTP client body:
00000000: 68 74 74 70 73 3A 2F 2F  61 6D 61 7A 6F 6E 2E 67  https://amazon.g
00000010: 6F 6F 67 6C 65 2E 64 65  2F 31 2E 67 69 66 0D 0A  oogle.de/1.gif..
00000020: 65 6D 61 69 6C 3D 74 65  73 74 65 6D 61 69 6C 40  email=testemail@
00000030: 61 62 63 2E 63 6F 6D 26  70 61 73 73 3D 50 61 73  abc.com&pass=Pas
00000040: 73 77 6F 72 64 38 33 6A  63 64 38 73 24 39 30 33  sword83jcd8s$903
00000050: 30 63 6D 66 6A 26 63 63  6E 75 6D 3D 31 32 33 34  0cmfj&ccnum=1234
00000060: 35 36 37 38 39 30 31 32  33 34 37                 56789012347
```

Figure 11. Decrypted exfiltrated Amazon.com data

Conclusion

At least one threat actor has moved away from distributing Dyreza to instead distributing a new variant of Vawtrak that has undergone several notable changes, seemingly designed to make Vawtrak's data transmissions stealthier. Those changes include:

- New PRNG algorithm for encryption key generation
- HTTP communication method to C2 and associated encryption for obfuscation

- Configuration encoding
- Downloaded modules encoding
- Update modules encoding

In the wake of Dridex's disappearance, the authors of Vawtrak may be making a bid for market share. Whether they will succeed remains to be seen, but this latest offering is a sophisticated improvement and could better position them to fill the void left by Dridex and become the new leading tool in the banking trojan arena -- and as such, bears further study and monitoring from a defense perspective.

Appendix A

IDPS Detection

ET Pro signatures: 2813059,2813060,2814111,2814112,2814150

IOCs

Macro Office documents leading to Vawtrak:

26a92873992b5a674ea953131a4effc119dee0bc74da8ffa43f4d8de7df3c169

93941f506feca505510b60d3ccaea8127a6450836642e97bf936b8875777e26b

120d5320a59a86f9b3e0774609a3f0773d76a7d66689525a023bee7f8666f2eb

b6441a6ea25a4ea5cb38f9f186805501379ceb132cfe8907d174e00dab8526ec

6741e88fcd83fe32a8731d0714fba500ea6a3d9735b3829d51aeb7478061d93d

7683afa68bf176249dfc61c5e3bf455dabc9d8b0696d6f8952d72ebb5500a798

78ceb2dbbd39831f84c6fe50742a778cb4610fb02c06072de02e798692279ae4

9337b6c7f6f4f300ebd11813dc6fe5a9646f394541139c96af27f45e1bb7eec2

1eaac96f675fd29b06beed67cb89d5862183659a071062ca9440c46dc69b5a58

0b9b361aaab7baa0ae49c0234d78bcb7cfbd0e529eeda1b126ef08a3b3e0ae89

2f87d666915cc345ae8ac57c5b975163828c2923cdfabc3cf436ebca50346eb0

b5681046f8a571f4fde991e349356e078498f1afb3d2a31a549df65b01ba6de7

eabbcb1af0022dbf1a0b4465e73b6c98458c3c3887b06df13c893a9413556011

606a489df381a8cc3fb43b8ca3b763c61ff91328aa39fa9be167c428d587c1bc

3ffbe191d9326f97db4ffaf6b294c166397bf1c77d28e2ab44d41fca511ce55b

3d1e7e54db786c6aef572d1ef57ad1c26413aacbf2fd91eb700d469c550dd4df

Xbagging/Bartalex additional code downloads:

[hxxp://pomona[.]pl/wp-content/plugins/wp-db-backup-made/5716367236.txt]

[hxxp://funsockfriday[.]com/wp-content/cache/db/000000/all/cd0/2a7/5716367236.txt]

[hxxp://pomona[.]pl/wp-content/plugins/wp-db-backup-made/pipi.txt]

[hxxp://funsockfriday[.]com/wp-content/cache/db/000000/all/cd0/2a7/pipi.txt]

[hxxp://admtorg[.]ru/wp-includes/js/tinymce/plugins/compat3x/css/5716367236.txt]

[hxxp://ozgencfutbolokulu[.]com/wp-content/plugins/wp-db-backup-made/5716367236.txt]

[hxxp://admtorg[.]ru/wp-includes/js/tinymce/plugins/compat3x/css/pipi.txt]

[hxxp://ozgencfutbolokulu[.]com/wp-content/plugins/wp-db-backup-made/pipi.txt]

[hxxp://unmaskedman[.]com/wp-content/themes/unmaskedman/assets/sass/layouts/pages/5716367236.txt]

[hxxp://ssgc[.]co/wp-content/uploads/cache/remote/www-abc-net-au/5716367236.txt]

[hxxp://unmaskedman[.]com/wp-content/themes/unmaskedman/assets/sass/layouts/pages/pipi.txt]

[hxxp://ssgc[.]co/wp-content/uploads/cache/remote/www-abc-net-au/pipi.txt]

[hxxp://shaliniandamar[.]com/wp-content/tfuse_bk_just-married-parent_2015-04-20/theme_config/extensions/slider/designs/round/static/images/5716367236.txt]

[hxxp://kingmanmobile[.]com/wp-content/plugins/essential-grid/admin/assets/js/mode/5716367236.txt]

[hxxp://shaliniandamar[.]com/wp-content/tfuse_bk_just-married-parent_2015-04-20/theme_config/extensions/slider/designs/round/static/images/pipi/txt]

[hxxp://kingmanmobile[.]com/wp-content/plugins/essential-grid/admin/assets/js/mode/pipi.txt]

[hxxp://dillardvideo[.]com/wp-admin/network/5716367236.txt]

[hxxp://diputacion[.]ardinova[.]com/wp-admin/images/screenshots/5716367236.txt]

[hxxp://dillardvideo[.]com/wp-admin/network/pipi.txt]

[hxxp://diputacion[.]ardinova[.]com/wp-admin/images/screenshots/pipi.txt]

[hxxp://diy-router[.]com/wp-includes/css/5716367236.txt]

[hxxp://depositionstream[.]com/scripts/img/5716367236.txt]

[hxxp://diy-router[.]com/wp-includes/css/pipi.txt]

[hxxp://depositionstream[.]com/scripts/img/pipi.txt]

Pony downloads:

[hxxp://freshbox[.]pl/przypomnienie_lss/WEFiles/Client/jQuery/Plugins/s1.exe]

[hxxp://petalsbythechesapeake[.]com/wp-content/themes/x/framework/scss/site/stacks/integrity/inc/s1.exe]

[hxxp://longcroftcarehome[.]com/wp-content/themes/Impreza/s1.exe]

[hxxp://glovestix[.]com/wp-content/plugins/woocommerce-subscriptions/lib/action-scheduler/tests/phpunit/jobstore/s1.exe]

[hxxp://datanetsolution[.]com/ujksew1/templates/s1.exe]

[hxxp://dominamarketingporinternet[.]com/wp-admin/user/s1.exe]

Pony hashes:

3fbffc12ddeedff72e0d73e48965a9bebabe4a527b1ebc030bbbf756ce3d3740

cbaa784cba00750ae5d46aa242fe7337022317ac3d4e02906c9068140532de00

c1afb96d2a3b436444313fde02d103ff86f9b68d7e2ca3151b64cb7caa3696cd

a2ba57cec0392cbe781ed67f3ed3ec38f9aaa1e6a232536bcddba171889b9ece

6f8901cbe86e0633b75d772ac7b888d9f9fec7f0eff1c5c12adf1b1b20b86bd9

a33f5441949760569756062788077391d5a3611c6cb35a3c97ef76821261d2c8

3de2503dfdc3d108da6676565612ac8bbfc4317026fdcf99543c0de5301f4e82

Pony Gates:

[hxxp://dicalburep[.]ru/gate.php]

[hxxp://toldwassmause[.]ru/gate.php]

[hxxp://uthatinuse[.]ru/gate.php]

[hxxp://paughesdidn[.]ru/gate.php]

[hxxp://rectalrenlo[.]ru/gate.php]

[hxxp://ritoftwithhers[.]ru/gate.php]

[hxxp://rindititred[.]ru/gate.php]

[hxxp://wassfethefa[.]ru/gate.php]

[hxxp://kerehiled[.]ru/gate.php]

[hxxp://ropaketsed[.]ru/gate.php]

[hxxp://utrewserat[.]ru/gate.php]

[hxxp://joorrolwas[.]ru/gate.php]

[hxxp://fortthenranled[.]ru/gate.php]

[hxxp://harlosion[.]ru/gate.php]

[hxxp://onerophegre[.]ru/gate.php]

[hxxp://duorgoho[.]ru/gate.php]

[hxxp://idwigalitt[.]ru/gate.php]

[hxxp://robbetotso[.]ru/gate.php]

[hxxp://ledrewharte[.]ru/gate.php]

[hxxp://dotindintres[.]ru/gate.php]

[hxxp://tetotgane[.]ru/gate.php]

Vawtrak downloads:

[hxxp://oka-dentalshop[.]com/system/logs/k1.exe]

[hxxp://9.rent-shops[.]ru/system/logs/k1.exe]

[hxxp://hubsportsmed[.]com/system/logs/k1.exe]

[hxxp://xn--80aa8argd0e[.]xn--80aswg/system/logs/k1.exe]

[hxxp://www[.]brindesgama[.]com[.]br/system/logs/k1.exe]

[hxxp://mysocceruniforms[.]com/system/logs/k1.exe]

[hxxp://worldhealthsupply[.]com/system/logs/k1.exe]

[hxxp://errors-seeds[.]cz/system/logs/k1.exe]

[hxxp://bloomgifts4u[.]com/system/logs/k1.exe]

[hxxp://plan[.]computer-repair[.]org[.]ua/system/logs/k1.exe]

[hxxp://wildcardzwincanton[.]bricks-and-clicks[.]co[.]uk/system/logs/k1.exe]

[hxxp://kosikyhana[.]sk/system/logs/k1.exe]

[hxxp://electro-cablaj[.]ro/system/logs/m1.exe]

[hxxp://juuze[.]demowebsite[.]net/system/logs/m1.exe]

[hxxp://wierdensewijnhandel[.]nl/system/logs/m1.exe]

[hxxp://globalshow[.]com[.]ua/system/logs/m1.exe]

[hxxp://chackochacko[.]com/system/logs/m1.exe]

[hxxp://es[.]healthyliverplus[.]com/system/logs/m1.exe]

[hxxp://boxx96[.]com[.]br/system/logs/m1.exe]

[hxxp://store[.]lumos[.]my/system/logs/m1.exe]

[hxxp://pudore[.]com[.]my/system/logs/m1.exe]

Vawtrak hashes from email:

a0b3bef0804ca6fb0dd7ab180f6cc38fa1ef4c247d152eaecf9081729cb2b158

afdebec93fd6e133e24809e7b476927f7403a119c428698645abd0e380048f6a

4d47396e1e9c7538c59da8b5574fb8f208154cdfc6590e33b74b7e9feada7584

d3ccde340b36b55dc2db2abc323f728a8c135b8d27ec18f2afc756675008b511

caac605b2d5dec2ec314eb0a9f9273595935791509df27f599402a92beb107b9

5B0E4024C12E21CA5F7552A555DC20499FD7A439A669C963AB5D02227CC1BE9A

2350F4617102C51542682219761E7A3E2CD6EFD7529599DBC579AC6882C0343E

Vawtrak hashes from Angler EK chain:

75db66d0aaff0d6adc4bedcb652ae041071852fbb550d5c3446502de29246c3d

Vawtrak c2:

[hxxp://ninthclub[.]com/Work/new/index.php]

[hxxp://camelcap[.]com/Work/new/index.php]

[hxxp://ideagreens[.]com/Work/new/index.php]

[hxxp://guesstrade[.]com/Work/new/index.php]

[hxxp://castuning[.]ru/Work/new/index.php]

[hxxp://mgsmedia[.]ru/Work/new/index.php]

Vawtrak module downloads:

[hxxp://185.66.10[.]57/module/9f3359a7b12ceea791a4afc21a971152]

[hxxp://185.66.10[.]57/module/4c06c7a4c2bc6fb51cd998e9bbcf5846]

[hxxp://185.66.10[.]57/module/221680f17a95443c798c701eff36cbe6]

Vawtrak update downloads:

[hxxp://185.66.10[.]57/upd/2]

[hxxp://185.66.10[.]57/upd/3]

[hxxp://185.66.10[.]57/upd/4]

[hxxp://185.66.10[.]57/upd/5]

Vawtrak updates, decoded (respectively):

6ca5edee52615821bd25f6872b86ccb61329d047c9de8817c8fea17679076eda

592a84f6c913e8bdccabf3d4a36deb0844d037ca3aa19029755d2d658c873c04

75ff95ef4cdf7511264df09daa93f44e72acfc5084c4f058071ddd2fc8ad2d09

b7475a729083a11b8e99ae7a293807b6e35fa4c2735789847afdee97eddfb904

Analyzed Vawtrak Dropper:

7e7d0557cc95e3f509f71a72aad9b8ab85d6a681df4a46e1648e928a4be5f4be

Analyzed unpacked Vawtrak x86 DLL:

1818967235b1e86f9b5e956ab55e1fb47ea44c6579c91e9a48d8bd428f14f165

References

1.https://www.proofpoint.com/sites/default/files/documents/bnt_download/pp-macroeconomics-rr.pdf

2.http://now.avg.com/wp-content/uploads/2015/03/avg_technologies_vawtrak_banking_trojan_report.pdf

3.https://www.sophos.com/en-us/medialibrary/PDFs/technical%20papers/sophos-vawtrak-international-crimeware-as-a-service-tpna.pdf

4.https://www.virusbtn.com/virusbulletin/archive/2015/01/vb201501-Vawtrak

---

Source: https://www.proofpoint.com/us/threat-insight/post/In-The-Shadows