An Introduction to AlphaLocker

By Jim Walter

It is always a treat, as a malware researcher, to come across something new and unique, and to then follow the resulting rabbit hole as far as you can go. I believe most of us in the cybersecurity industry enjoy that particular part of the puzzle, especially when you are able to fully trace the origin of a novel artifact or binary. Starting with a single random file, and ending up with a broad picture of the economy behind that malware is highly satisfying and often eye-opening.

Which brings us to yet another family of ransomware – AlphaLocker.

Introducing AlphaLocker

This family of ransomware is directly purchased from the author via the Internet. The buyer can then choose to host/spread/distribute it in whatever way they see fit - as opposed to some of the more recent turn-key offerings like Ransom32, ORX-Locker, or Encryptor RAAS, which lack a full administrative panel and other customization features present in a fully packaged malware ‘kit’.

This is an interesting example to highlight for a couple of reasons. First and foremost, AlphaLocker is cheap compared to other types of ransomware. The first versions began to appear in March 2016, priced at only $65 USD, paid via Bitcoin.

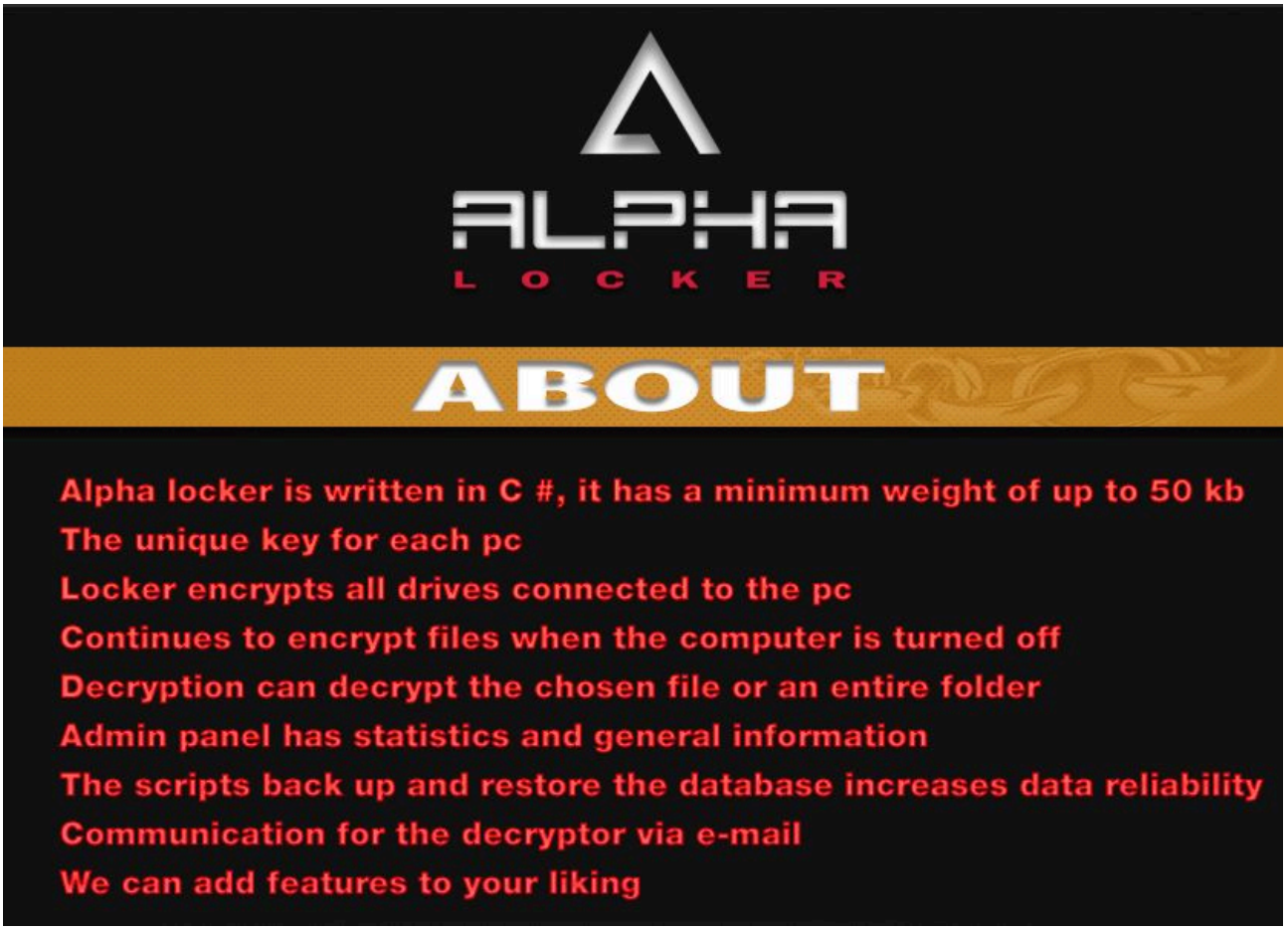


Figure1: AlphaLocker Advertising

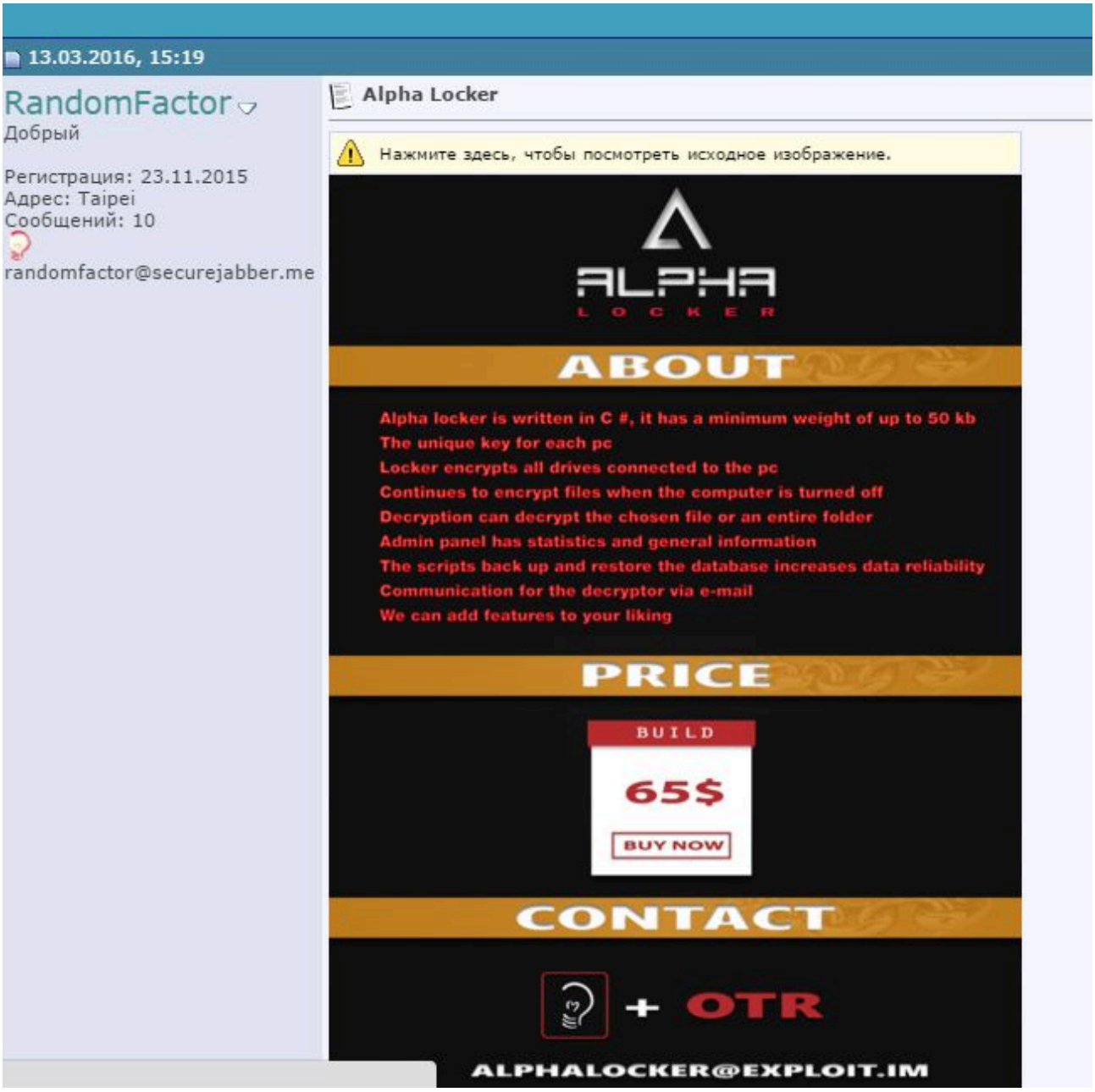


Figure 2: More AlphaLocker Advertising

Figure 3: Still More AlphaLocker Advertising

For that low price you get your own unique copy of the main executable (the actual ransomware), the master decryptor binary (based onHidden Tear), and your own administrative panel instance. Hosting, spreading, and other typical ransomware services are then left to the buyer.

The lower price point allows ‘less-skilled’ ne'er–do–wells to possess and control (and profit from) ransomware, with little to no coding and zero ramp-up time.

Sample Review

Also of note is the fact that the author(s) of AlphaLocker are continually generating updates to evade detection by traditional signature-based AV technologies. While that practice is absolutely the norm amongst malware authors, it never ceases to amaze just how easily the bad guys are able to keep up the evasion, staying one step ahead of signature-based detection technologies.

In reviewing a handful of AlphaLocker samples via a popular multi-engine sample-scanning site, most of the samples were only detected by between zero and nine out of 56 AV vendors. In one example, the AlphaLocker binary was compiled on 4/17/2016 and submitted to the sample-scanning site that same day, showing a 9/56 detection ratio. Upon a rescan 12 days later on 4/29/2016, the detection rate for that exact same sample was still only 22/56.

Worryingly enough, some of those detections are based on the respective vendors’ cloud-based detection mechanisms. In other words, if you take away connectivity to said service, that product would cease to detect as well.

We will cover detections with CylancePROTECT® further down in this blog, in order to illustrate how our artificial intelligence based approach stops malware dead pre-execution, without the need for signatures, prior knowledge of the sample, cloudy heuristics, frequent updates, and so on.

There is another critical point worth mentioning here. AlphaLocker is based on the Eda2 project, by Utku Sen. This was an 'open source' ransomware project that, until recently, was openly available via Utku's github. In January 2016, the source was pulled by Utku in response to the code being used in real attacks (and the data could not be recovered via a built-in backdoor). This is a CRITICAL point. Not only is the behavior blatantly and contextually malicious, but the actual source code is public and easy to find.

Again, there is no reason why any reputable AV product should fail to detect this ransomware. Unfortunately, most are still failing.

Delving Deeper

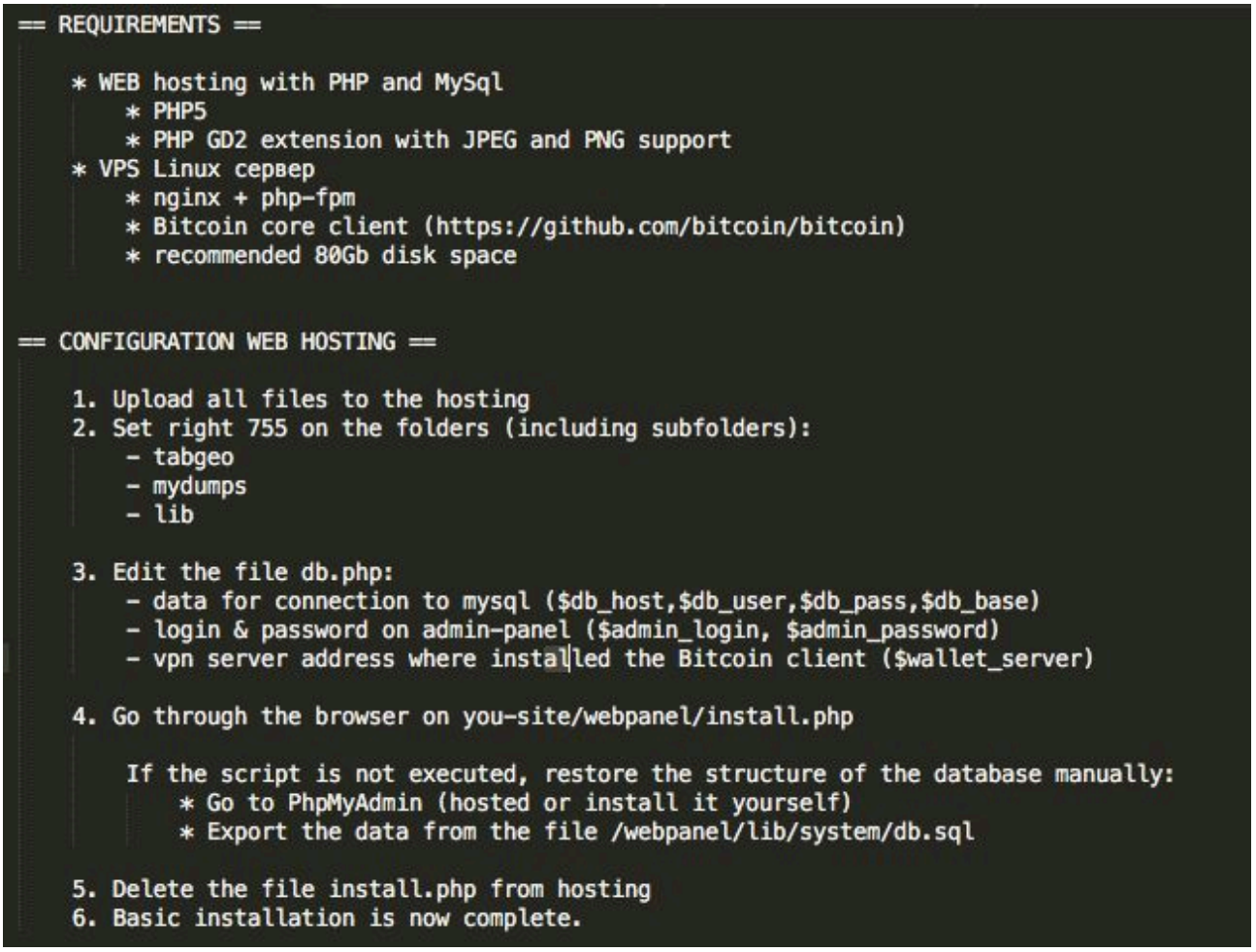
Before we explore this detection piece further, I'd like to walk you through the full AlphaLocker service, including the admin panels and specifics on the ransomware binaries themselves. During our analysis here at Cylance, we were able to get a rare and close glimpse into the AlphaLocker ecosystem. Sometimes we luck out and get to take careful advantage of silly oversights on the part of the 'bad guys'. In this case, we were able to find more than one active C2, where the initial config files were still present - in this case, install.php.

All of AlphaLocker's configuration and support files are unencrypted and in English, while the author(s) appear to be Russian (based on data contained in some of the panel files, as well as the particular forums in which the ransomware is advertised).

All of the included configuration and supporting files are shown below:

Figure 4: AlphaLocker C2 Panel Root 1

The included README file covers full installation, including setup of the panel itself (PHP modules, dependencies, etc.) as well as setup of the BTC-based payment system:



```
== REQUIREMENTS ==

    * WEB hosting with PHP and MySql
        * PHP5
        * PHP GD2 extension with JPEG and PNG support
    * VPS Linux сервер
        * nginx + php-fpm
        * Bitcoin core client (https://github.com/bitcoin/bitcoin)
        * recommended 80Gb disk space


== CONFIGURATION WEB HOSTING ==

    1. Upload all files to the hosting
    2. Set right 755 on the folders (including subfolders):
        - tabgeo
        - mydumps
        - lib

    3. Edit the file db.php:
        - data for connection to mysql ($db_host,$db_user,$db_pass,$db_base)
        - login & password on admin-panel ($admin_login, $admin_password)
        - vpn server address where installed the Bitcoin client ($wallet_server)

    4. Go through the browser on you-site/webpanel/install.php

        If the script is not executed, restore the structure of the database manually:
            * Go to PhpMyAdmin (hosted or install it yourself)
            * Export the data from the file /webpanel/lib/system/db.sql

    5. Delete the file install.php from hosting
    6. Basic installation is now complete.
```

Figure 5: AlphaLocker README 1

```
== КОНФИГУРАЦИЯ VPS СЕРВЕРА ==

    1. Install the required packages:
    sudo apt-get update
    sudo apt-get -y install qt4-qmake libqt4-dev libboost-dev libboost-system-dev libboost-filesystem-dev libboost-program-options-dev libboost-thread-dev
    sudo apt-get install git-core
    sudo apt-get install build-essential
    sudo apt-get install libssl-dev
    sudo apt-get install autoconf
    sudo apt-get -y install libtool autotools-dev
    sudo apt-get -y install libminiupnpc-dev
    sudo apt-get -y install libdb++-dev
    sudo apt-get -y install libprotobuf-dev
    sudo apt-get -y install libqrencode-dev
    sudo apt-get -y install libboost-all-dev
    sudo apt-get -y install pkg-config

    cd ~
    mkdir downloads
    cd ~/downloads
    wget http://download.oracle.com/berkeley-db/db-4.8.30.tar.gz
    tar zxvf db-4.8.30.tar.gz
    rm -f db-4.8.30.tar.gz
    cd db-4.8.30/build_unix
    ../dist/configure --prefix=/usr/local --enable-cxx
    make
    sudo make install
    cd ~/downloads
    rm -fr db-4.8.30/

    2. Installing bitcoind

    cd ~/downloads
    git clone https://github.com/bitcoin/bitcoin.git
    cd bitcoin
    ./autogen.sh
    ./configure
    make
    sudo make install
```

Figure 6: AlphaLocker README 2

```
5. Bitcoin client installation complete
    * To start the service, type in the terminal:
      bitcoind
    ***Important: create account 'mainwallet', it will gather all Bitcoin
              type in the terminal:
              bitcoin-cli getnewaddress mainwallet

    * For control via the terminal:
      bitcoin-cli <command>
      List and description of the commands:
      bitcoin-cli help

    * After starting the client starts synchronization (download) the entire history of bitcoin transactions
      This process may take considerable time and CPU consumption of server resources
      In March 2016 the base weight of 69Gb

    * Do not forget sometimes to make a backup of the wallet
        bitcoin-cli dumpwallet ~/downloads/dumpwallet.dat

6. Configuration nginx + php-fpm

    apt-get install nginx
    service nginx start
    apt-get install php5-fpm
    sudo nano /etc/nginx/nginx.conf

    Edit the following lines in the file:
    ...
    worker_processes  4;
    ...
        keepalive_timeout   2;
    ...
```

Figure 7: AlphaLocker README 3

The admin panel credentials AND the MySQL root credentials are stored, in plain text, within db.php in the server's root:

```
♦♦♦ yc♦a♦o♦♦♦ ♦a♦o♦♦♦♦e ♦a♦♦♦e ♦♦♦ ♦o♦♦♦♦♦e♦♦♦
♦ mysql cep♦cpy ♦ ♦a♦♦♦♦c ♦a install.php
*/

//♦o♦♦♦♦♦e♦♦e ♦ mysql
$db_host - "localhost";
$db_user = "root";
$db_pass = "[redacted]us!";
$db_base = "database1";
//♦o♦♦♦/♦apo♦♦ o♦ a♦♦♦♦♦♦♦
$admin_login = "[redacted]ist";
$admin_password = "fuckoff";
$admin_whois ="http://whois.domaintools.com/";
$wallet_server ="localhost"; /*address with php-connector to bitcoin demon*/
$bitcoin_price = 0.35;
/* ♦♦♦e ♦ac♦po♦♦♦ ♦♦♦ updatebd.php ♦ dumpersqldata.php
```

Figure 8: AlphaLocker db.php

AlphaLocker's admin login panel can be reached via login.php:

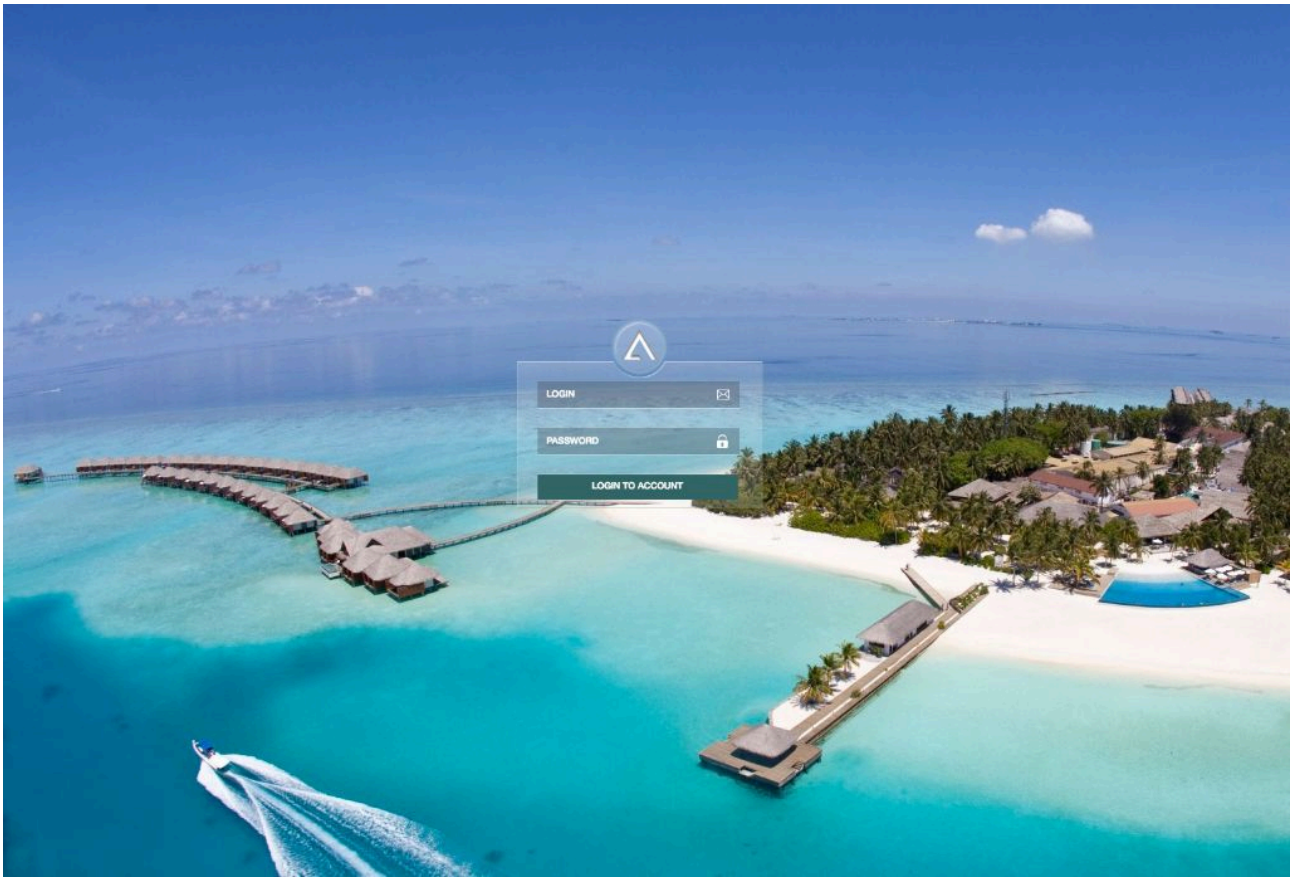


Figure 9: AlphaLocker Panel 1

Upon login, infected and encrypted hosts can be viewed and managed by the buyer. Stats and various maintenance settings are available as well.

Figure 10: AlphaLocker Panel 2

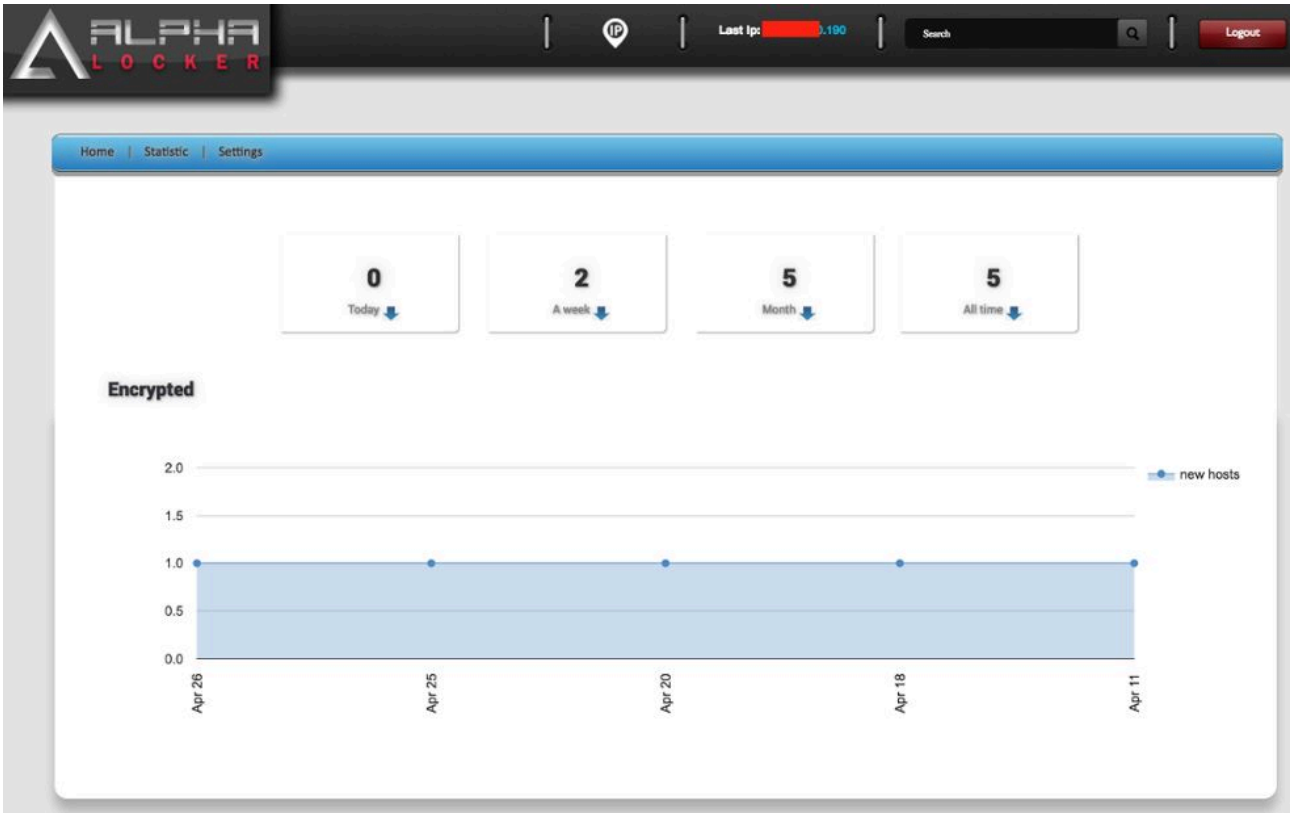


Figure 11: AlphaLocker Panel 3

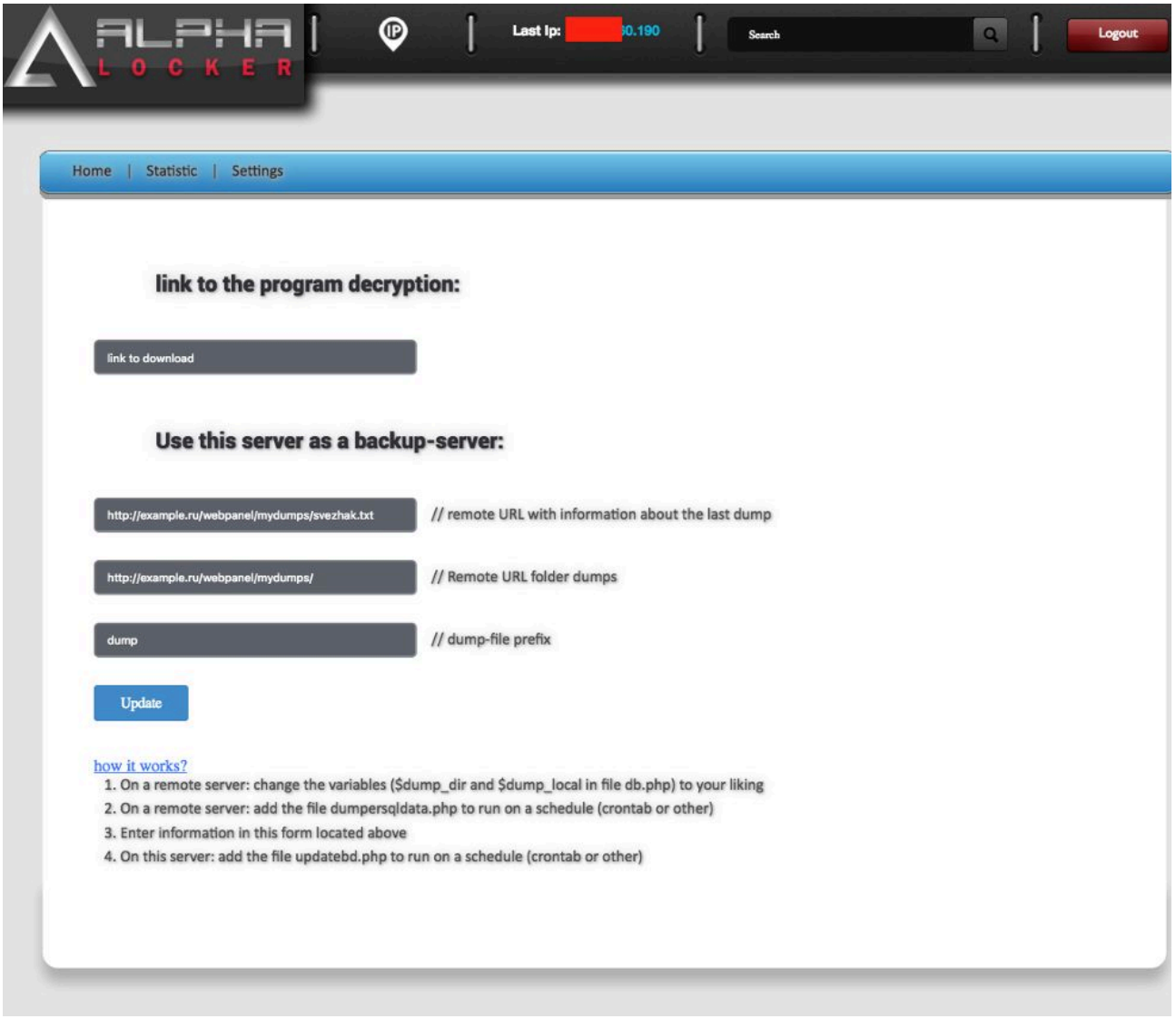


Figure 12: AlphaLocker Panel 4

The Malware

The actual ransomware executables for AlphaLocker are rather straightforward, given the EDA2 foundation. Files are individually encrypted with their own unique key (AES). AES keys are RSA-encrypted via a keypair stored in the local MySQL DB and posted to the C2. In general, for most EDA2-based malware, the flow is similar to the following:

1. **1. The executable sends a POST request to the C2, which contains the unique ID for the victim.**
2. **2. The C2 creates the public/private RSA (2048) keypair, and sends the public key to the main ransomware executable. The private key remains stored in the DB.**
3. **3. A random AES key is generated (per file).**
4. **4. The ransomware executable encrypts each file with the newly generated AES key.**

5.

5. The AES key is encrypted with the RSA public key, and is then sent to the C2 via POST.

The samples we tested also modify the desktop background on the victim/host's computer. A common thread across the EDA2-based threats is the hosting of background images on imgur.com.

The affected encrypted file types can vary across samples. The individual buyer of the ransomware has the choice of file types to support. The default set of files, which will be encrypted across all available drives is as follows:

.asf, .pdf, .xls, .docx, .xlsx, .mp3, .waw, .jpg, .jpeg, .txt, .rtf, .doc, .rar, .zip, .psd, .tif, .wma, .gif, .bmp, .ppt, .pptx, .docm, .xlsm, .pps, .ppsx, .ppd, .eps, .png, .ace, .djvu, .tar, .cdr, .max, .wmv, .avi, .wav, .mp4, .pdd, .php, .aac, .ac3, .amf, .amr, .dwg, .dxf, .accdb, .mod, .tax2013, .tax2014, .oga, .ogg, .pbf, .ra, .raw, .saf, .val, .wave, .wow, .wpk, .3g2, .3gp, .3gp2, .3mm, .amx, .avs, .bik, .dir, .divx, .dvx, .evo, .flv, .qtq, .tch, .rts, .rum, .rv, .scn, .srt, .stx, .svi, .swf, .trp, .vdo, .wm, .wmd, .wmmp, .wmx, .wvx, .xvid, .3d, .3d4, .3df8, .pbs, .adi, .ais, .amu, .arr, .bmc, .bmf, .cag, .cam, .dng, .ink, .jif, .jiff, .jpc, .jpf, .jpw, .mag, .mic, .mip, .msp, .nav, .ncd, .odc, .odi, .opf, .qif, .xwd, .abw, .act, .adt, .aim, .ans, .asc, .ase, .bdp, .bdr, .bib, .boc, .crd, .diz, .dot, .dotm, .dotx, .dvi, .dxe, .mlx, .err, .euc, .faq, .fdr, .fds, .gthr, .idx, .kwd, .lp2, .ltr, .man, .mbox, .msg, .nfo, .now, .odm, .oft, .pwi, .rng, .rtx, .run, .ssa, .text, .unx, .wbk, .wsh, .7z, .arc, .ari, .arj, .car, .cbr, .cbz, .gz, .gzig, .jgz, .pak, .pcv, .puz, .rev, .sdn, .sen, .sfs, .sfx, .sh, .shar, .shr, .sqx, .tbz2, .tg, .tlz, .vsi, .wad, .war, .xpi, .z02, .z04, .zap, .zipx, .zoo, .ipa, .isu, .jar, .js, .udf, .adr, .ap, .aro, .asa, .ascx, .ashx, .asmx, .asp, .indd, .asr, .qbb, .bml, .cer, .cms, .crt, .dap, .htm, .moz, .svr, .url, .wdgt, .abk, .bic, .big, .blp, .bsp, .cgf, .chk, .col, .cty, .dem, .elf, .ff, .gam, .grf, .h3m, .h4r, .iwd, .ldb, .lgp, .lvl, .map, .md3, .mdl, .nds, .pbp, .ppf, .pwf, .pxp, .sad, .sav, .scm, .scx, .sdt, .spr, .sud, .uax, .umx, .unr, .uop, .usa, .usx, .ut2, .ut3, .utc, .utx, .uvx, .uxx, .vmf, .vtf, .w3g, .w3x, .wtd, .wtf, .ccd, .cd, .cso, .disk, .dmg, .dvd, .fcd, .flp, .img, .isz, .mdf, .mds, .nrg, .nri, .vcd, .vhd, .snp, .bkf, .ade, .adpb, .dic, .cch, .ctt, .dal, .ddc, .ddcx, .dex, .dif, .dii, .itdb, .itl, .kmz, .lcd, .lcf, .mbx, .mdn, .odf, .odp, .ods, .pab, .pkb, .pkh, .pot, .potx, .pptm, .psa, .qdf, .qel, .rgn, .rrt, .rsw, .rte, .sdb, .sdc, .sds, .sql, .stt, .tcx, .thmx, .txd, .txf, .upoi, .vmt, .wks, .wmdb, .xl, .xlc, .xlr, .xlsb, .xltx, .ltm, .xlwx, .mcd, .cap, .cc, .cod, .cp, .cpp, .cs, .csi, .dcp, .dcu, .dev, .dob, .dox, .dpk, .dpl, .dpr, .dsk, .dsp, .eql, .ex, .f90, .fla, .for, .fpp, .jav, .java, .lbi, .owl, .pl, .plc, .pli, .pm, .res, .rsrc, .so, .swd, .tpu, .tpx, .tu, .tur, .vc, .yab, .aip, .amxx, .ape, .api, .mxp, .oxt, .qpx, .qtr, .xla, .xlam, .xll, .xlv, .xpt, .cfg, .cwf, .dbb, .slt, .bp2, .bp3, .bpl, .clr, .dbx, .jc, .potm, .ppsm, .prc, .prt, .shw, .std, .ver, .wpl, .xlm, .yps, .1cd, .bck, .html, .bak, .odt, .pst, .log, .mpg, .mpeg, .odb, .wps, .xlk, .mdb, .dxg, .wpd, .wb2, .dbf, .ai, .3fr, .arw, .srf, .sr2, .bay, .crw, .cr2, .dcr, .kdc, .erf, .mef, .mrw, .nef, .nrw, .orf, .raf, .rwl, .rw2, .r3d, .ptx, .pef, .srw, .x3f, .der, .pem, .pfx, .p12, .p7b, .p7c, .jfif, .exif

Example: http://i.imgur.com/(xxxxx).jpg (where xxxxx is a random string of letters).

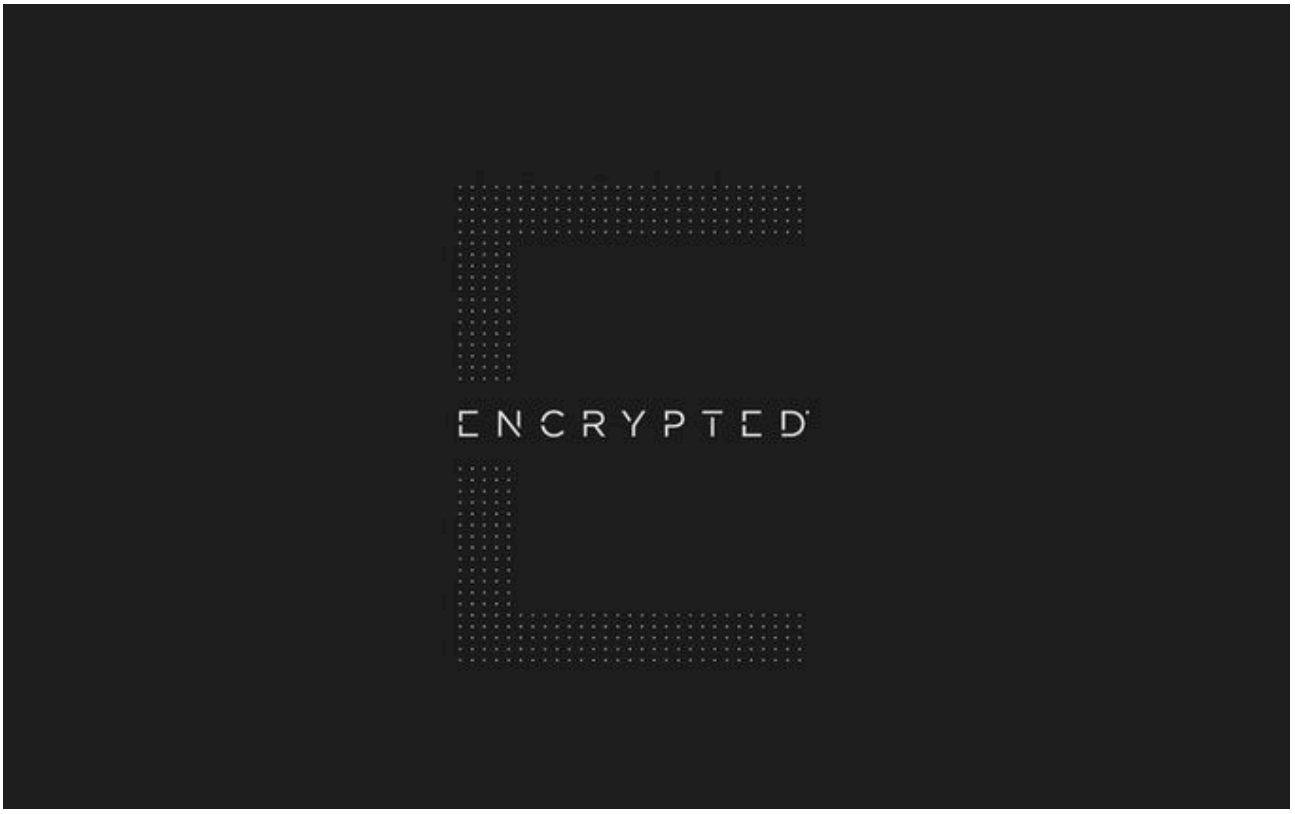


Figure 13: Background Image Direct 1

Figure 14: Background Image Direct 2

AlphaLocker is not the only family of ransomware pulling their wallpaper from imgur.com in this fashion. Another EDA2-based kit known as SkidLocker does the same:

Figure 15: SkidLocker Background 1

Skidlocker strings show the hosting of the image as follows:

http : //23.227.199.175/createkeys.php
http : //23.227.199.175/getamount.php
http : //23.227.199.175/savekey.php
http : //23.227.199.175/update.php
http : //23.227.199.175/finished.php
http : //23.227.199.175/exception.php
http : //i.imgur.com/By3yCwd.jpg
http : //23.227.199.83/Decrypter.exe

Upon infection and subsequent encryption of files, the victim is provided with a simple text file with instructions on how to pay the ransom. Key.php is typically hosted on the same C2 server, and the victim's unique ID is passed to key.php as a parameter:

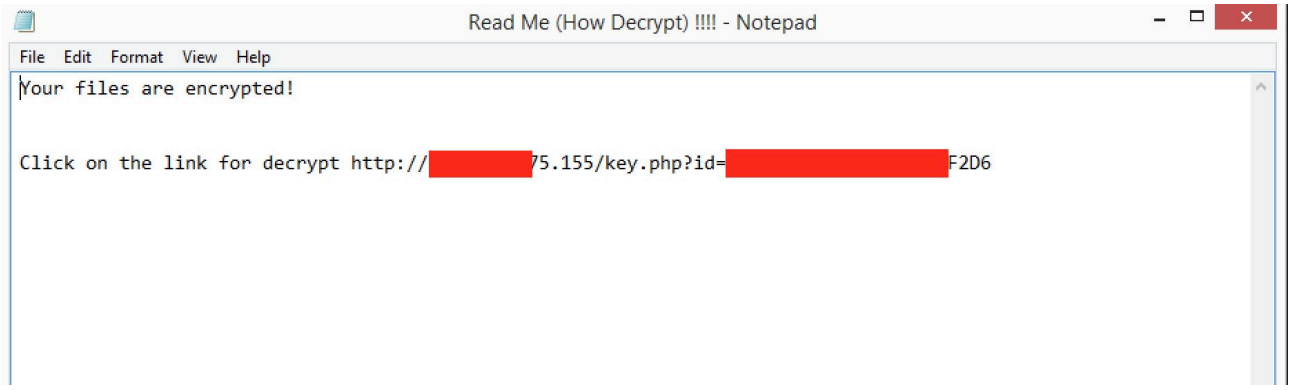


Figure 16: AlphaLocker Ransom Note (README Notepad File)

When the victim browses to the assigned URL, they will see the following BTC payment interface:

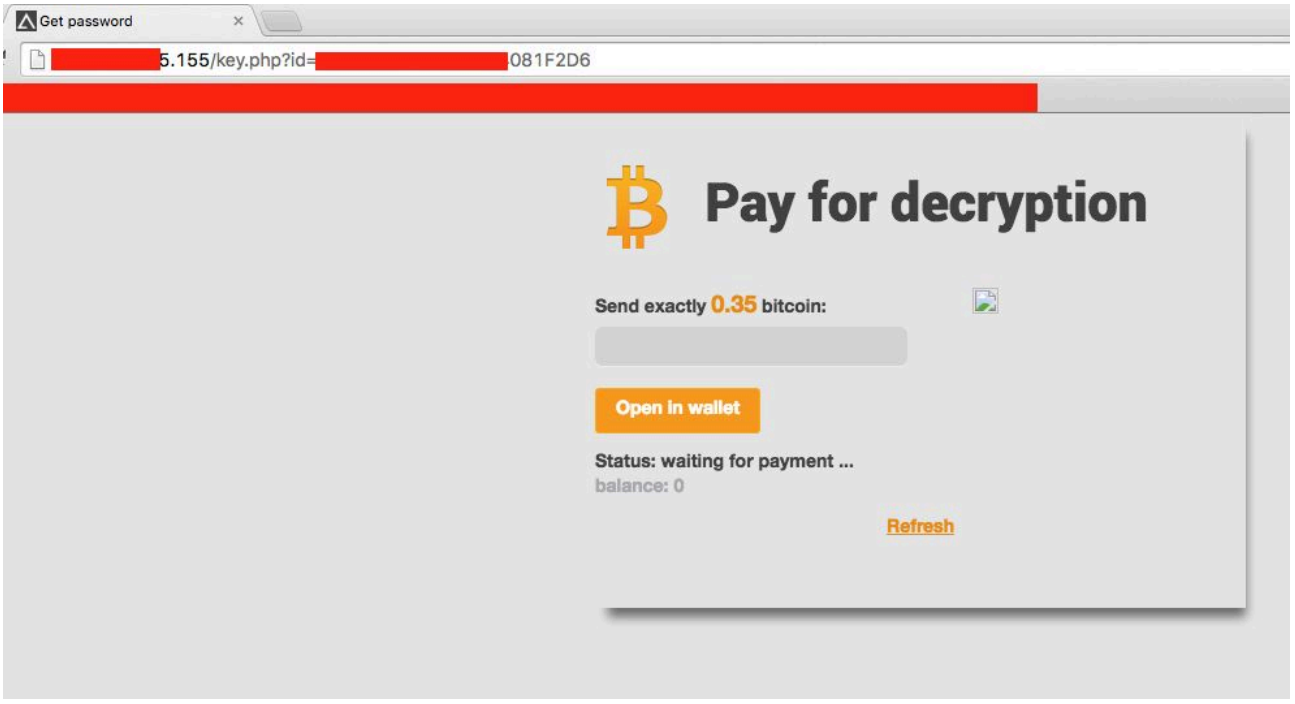


Figure 17: AlphaLocker BitCoin Payment page

So, as far as EDA2-based ransomware goes, AlphaLocker sticks close to default.

CylancePROTECT vs. AlphaLocker

Moving onto detection, we gathered multiple samples of AlphaLocker from a variety of sources and pitted them against CylancePROTECT. CylancePROTECT detected and prevented 100% of the AlphaLocker samples tested:

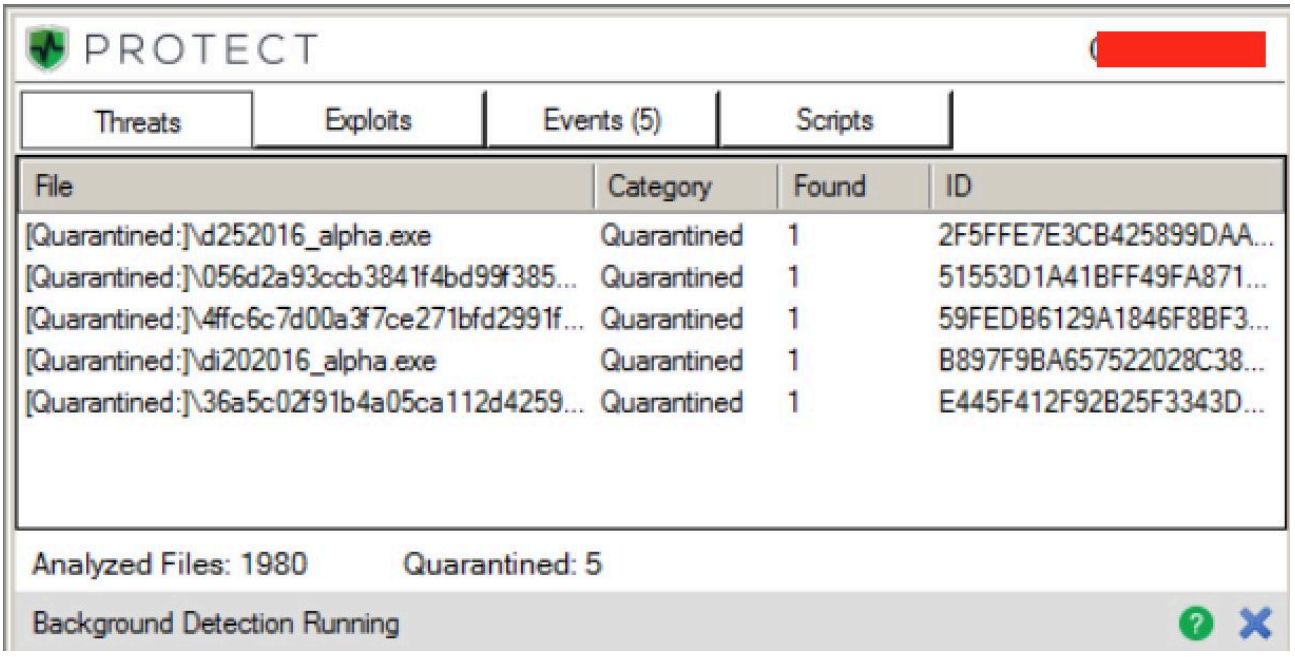


Figure 18: CylancePROTECT Minimized View, Showing Quarantined AlphLocker Files

Figure 19: CylancePROTECT Console View

Hashes (SHA256)

2f5ffe7e3cb425899daa815145112297b4cb1e712835e997ef64518efa212754
59fedb6129a1846f8bf3ba7717d87dd17f9f6ebf5c2089bb17cb766f67219c56
b897f9ba657522028c38ba260da17c58c8f75e4e7faca75e681f4c4cb60b90c9
ea33d7c7948a02f40f7c2531379bf0046e1d45b5d2b9bf4d9de88b77476f1600
51553d1a41bff49fa871269f232bba5f5567f34071ebd133b677bffedc26c90f
e445f412f92b25f3343d5f7adc3c94bdc950601521d5b91e7ce77c21a18259c9

Convinced that the next generation of endpoint security is right for your organization? Contact a Cylance expert to get started!
