Closing the Door: DeadBolt Ransomware Locks Out Vendors With Multitiered Extortion Scheme

trendmicro.com/en_us/research/22/f/closing-the-door-deadbolt-ransomware-locks-out-vendors-with-mult.html

June 6, 2022

In this report, we investigate the reasons that the DeadBolt ransomware family is more problematic for its victims than other ransomware families that previously targeted NAS devices.

By: Trend Micro June 06, 2022 Read time: (words)

By Stephen Hilt, Éireann Leverett, Fernando Mercês

The DeadBolt ransomware kicked off 2022 with a slew of attacks that targeted internet-facing Network-Attached Storage (NAS) devices. It was first seen targeting QNAP Systems, Inc. in January 2022. According to a report from attack surface solutions provider Censys.io, as of Jan. 26, 2022, out of 130,000 QNAP NAS devices that were potential targets, 4,988 services showed signs of a DeadBolt infection. A few weeks later, ASUSTOR, another NAS devices and video surveillance solutions vendor, also experienced DeadBolt ransomware attacks that targeted an unknown number of its devices. In March, DeadBolt attackers once again targeted QNAP devices; according to Censys.io, the number of infections reached 1,146 by March 19, 2022. Most recently, on May 19,2022, QNAP released a product security update stating that internet-connected QNAP devices were once again been targeted by DeadBolt, this time aiming at NAS devices using QTS 4.3.6 and QTS 4.4.1.

It’s interesting to note that the number of DeadBolt-infected devices is considerably high for a ransomware family that is exclusively targeting NAS devices. Earlier in 2022, we discussed the evolving landscape of attacks waged on the internet of things (IoT) and how cybercriminals have added NAS devices in their list of targeted devices. Our report detailed the ransomware families that cybercriminals used to target NAS devices, which include Qlocker, eCh0raix, and even bigger ransomware families such as REvil (aka Sodinokibi).

DeadBolt is peculiar not only for the scale of its attacks but also for several advanced tactics and techniques that its malicious actors have implemented, such as giving multiple payment options, one for the user and two for the vendor. However, based on our analysis, we did not find any evidence that it’s possible for the options provided to the vendor to work due to the way the files were encrypted. Essentially, this means that if vendors pay any of the ransom amounts provided to them, they will not be able to get a master key to unlock all the files on behalf of affected users.

Consider this example to understand this particular DeadBolt tactic: A crime group changes every lock in an entire apartment complex. The group then informs the apartment complex owner that they can give the apartment complex owner a master key that would allow the owner to successfully unlock all the apartment doors for his tenants if he pays them a certain amount. But in reality, the locks that the crime group installed are not master-keyed locks, making it impossible for the apartment complex owner to open the locks with one master key.

NAS devices typically contain sensitive files for both personal users and organizations. And the never-before-seen volume of NAS devices that this ransomware family has infected in a short period has led us to an investigation of DeadBolt. In this report, we investigate the reasons that the DeadBolt ransomware family is more problematic for its victims than other ransomware families that previously targeted NAS devices. We also used pertinent data to check if any user or vendor paid ransom, and how much the ransomware actors made from these attacks.

Research highlights

- The DeadBolt ransomware family targets QNAP and Asustor NAS devices.

- This ransomware uses a configuration file that will dynamically choose specific settings based on the vendor that it targets, making it scalable and easily adaptable to new campaigns and vendors.
- DeadBolt offers two different payment schemes: either a victim pays for a decryption key, or the vendor pays for a decryption master key that would theoretically work to decrypt data for all victims. However, as of this writing, we have yet to find evidence that decryption via a master key is possible.
- No more than 8% of DeadBolt victims paid the ransom amount.
- Based on our analysis, DeadBolt actors have notable web and operating system development skills.

Technical analysis

On the technical side, DeadBolt is reasonably interesting: It combines both encryption and decryption functionalities in a single executable that parses a JSON-based configuration file that includes ransom prices and contact details. It also creates a nicely formatted webpage so that victims can have easy access to the ransom message and instructions.

DeadBolt samples are 64-bit Linux Executable and Linkable Format (ELF) files that have been compiled using the Go programming language. The malware is meant to be run manually by an attacker, or at least in a post-compromised environment. If one tries to execute a DeadBolt sample in a new, uncompromised environment, it just tells the user how to use it and then exits:

```
$ ./444
encrypt usage: ./444 -e <config> <dir>
decrypt usage: ./444 -d <key> <dir>
```

The two supported operation modes are encrypt (-e) and decrypt (-d). For encrypting, DeadBolt expects a JSON configuration file that we have yet to find in the wild. However, by reversing the file, we can infer a valid configuration file expected to be passed as an argument to the DeadBolt main executable:

```
{
"key": "5da2297bad6924526e48e00dbfc3c27a",
"cgi_path": "./cgi.sh",
"client_id": "fb2e2de57fb405512f539a1c302e2b4f",
"vendor_name": "Testing Vendor",
"vendor_email": "contact@testingvendor",
"vendor_amount": "0.5",
"payment_amount": "0.1",
"vendor_address": "3FZbgi29cpjq2GjdwV8eyHuJJnkLtktZc5",
"master_key_hash": "2dab7013f332b465b23e912d90d84c166aefbf60689242166e399d7add1c0189",
"payment_address": "1F1tAaz5x1HUXrCNLbtMDqcw6o5GNn4xqX",
"vendor_amount_full": "1.0"
}
```

The parameters are explained as follows:

Parameter name	Description
cgi_path	This is the path where a Bash Common Gateway Interface (CGI) script will be written. This script is later used to replace a legitimate script used in the device administration web interface.
client_id	This ID will be added to the encrypted files.
Key	A 128-bit Advanced Encryption Standard (AES) key used for encrypting individual files
master_key_hash	The SHA-256 hash master key

payment_amount	The ransom amount that the victim would need to pay to get a decryption key
payment_address	A Bitcoin wallet ID that the victim will use to pay the ransom amount
vendor_amount	The ransom amount that the actors will try to charge the vendor for disclosing vulnerability details
vendor_amount_full	The ransom amount that a vendor would need to pay to get the decryption master key and vulnerability details
vendor_address	A Bitcoin wallet ID that the vendor will use to pay the ransom amount
vendor_name	Should contain the vendor name of the victim’s device, such as QNAP
vendor_email	Contains the vendor’s email address

Besides a valid JSON configuration file, the DeadBolt executable expects to receive a directory to start encrypting or decrypting files. This is one of the first times during our analysis that we discovered how DeadBolt differs from other NAS ransomware families before it: It has an amount that the vendor, such as ASUSTOR or QNAP, could theoretically pay to get all of the victims' information back. Additionally, this is one of the first times that we have seen two ransoms in one attack — one for the victims so that they can regain access to their files and data and one for the NAS vendor. This two-pronged ransom demand tactic could also be highly effective in the case of a service provider in a supply chain compromise. In fact, the REvil group implemented a similar approach in its attack on Kaseya, in which an intrusion set that Trend Micro dubbed “Water Mare” was deployed. The approach involves an attacker taking over a software company and then pushing out a backdoored software update that installs embedded malware. The victims can choose to pay the ransom amount themselves, but they are also more likely to put pressure on the vendor to pay the ransom on their behalf.

DeadBolt actors demand individual victims to pay 0.03 bitcoin (US$1,159.56 as of this publishing) to get their data back, which is quite a lot of money to demand for encrypted NAS devices . Meanwhile, the vendors are given two ransom payout options: one is for just the information about the exploit, with the ransom demand starting at 5 bitcoins (US$ 193,259.50 as of this publishing), while the other is for the exploit information and the master decryption key, with a ransom demand of 50 bitcoins (US$1,932,595.00 as of this publishing).

We ran a test to see if DeadBolt can encrypt test files in a $HOME/test folder:

```
$ mkdir test
cp /bin/ls test/document.docx
cp /bin/top test/spreadsheet.xls
```

Entropy, a numeric indication of the degree of randomness, suggests that the higher the number, the more random it is. Higher numbers, or numbers with an entropy value greater than 7.0, also often indicate that a file is encrypted, compressed, or packed if the file is an executable. Finding the entropy of a file is a simple test to ensure that the ransomware is properly encrypting files. Here is an example that shows the entropy of some test files:

```
$ entropy test/*
5.85 test/document.docx
5.83 test/spreadsheet.xls
```

After providing the JSON configuration file and running DeadBolt on the test files, the files were encrypted, a .deadbolt extension was appended to them, and a ransom note was created:

```
$ ./444 -e deadbolt.json test/
done
$ ls test/
document.docx.deadbolt  '!!!_IMPORTANT_README_WHERE_ARE_MY_FILES_!!!.txt'   spreadsheet.xls.deadbolt
```

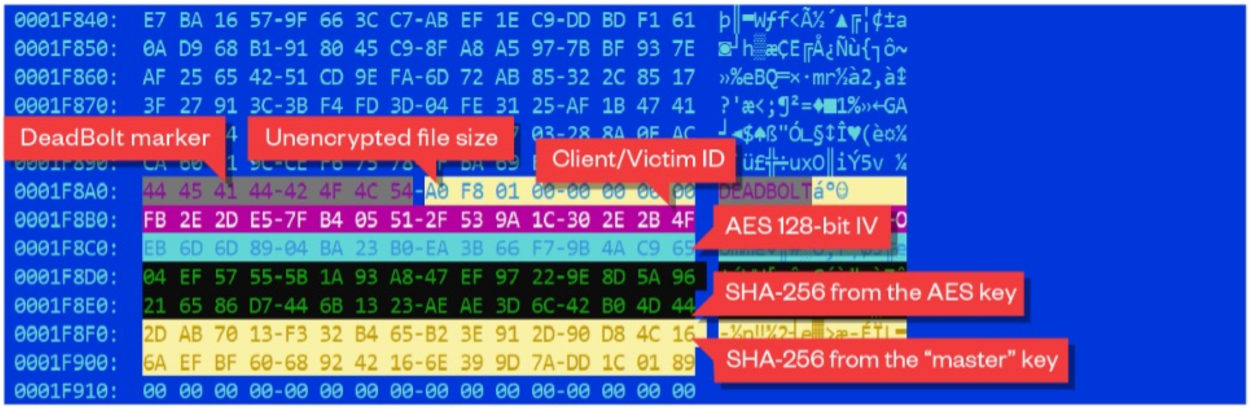
```
$ entropy test/*deadbolt
8.00 test/document.docx.deadbolt
8.00 test/spreadsheet.xls.deadbolt
```

After we ran DeadBolt on our test files, the entropy values increased from 5.8 to 8.0.

Encryption

DeadBolt uses AES-128-CBC to encrypt files with a provided key from the configuration file. After encrypting the file's content, it appends the following data to the encrypted file in binary format:

- A “DeadBolt” string
- The original file size
- A 16-byte client (victim) ID
- The AES initialization vector (IV) that is different for each file
- The SHA-256 of the AES 128-bit key
- The SHA-256 of the “master” key
- 16 null-bytes



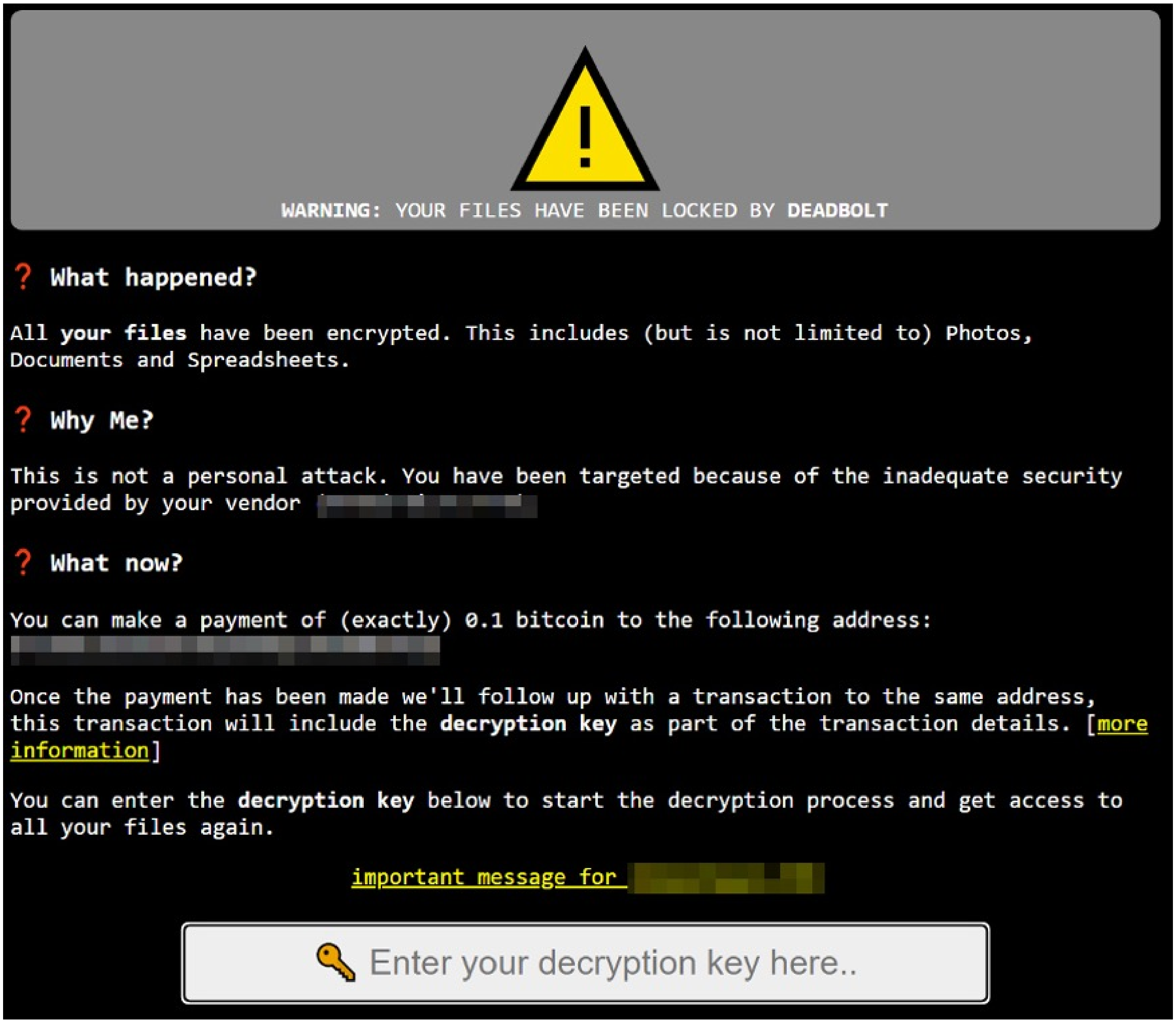
```
0001F840:  E7 BA 16 57-9F 66 3C C7-AB EF 1E C9-DD BD F1 61  þ║▬Wƒf<Ã½´▲╒¦¢±a
0001F850:  0A D9 68 B1-91 80 45 C9-8F A8 A5 97-7B BF 93 7E  ◙┘h▒æÇE╔Å¿Ñù{┐ô~
0001F860:  AF 25 65 42-51 CD 9E FA-6D 72 AB 85-32 2C 85 17  »%eBQ═×·mr½à2,à↨
0001F870:  3F 27 91 3C-3B F4 FD 3D-04 FE 31 25-AF 1B 47 41  ?'æ<;¶²=♦■1%»←GA
[illegible]                            03-28 8A 0F AC  ┘◄$♠ß"Ó└§↨Î♥(è☼%
[illegible]                                            ü£╪÷uxO║iÝ5v ½
0001F8A0:  44 45 41 44-42 4F 4C 54-A0 F8 01 00-00 00 00 00  DEADBOLTá°☺
0001F8B0:  FB 2E 2D E5-7F B4 05 51-2F 53 9A 1C-30 2E 2B 4F
0001F8C0:  EB 6D 6D 89-04 BA 23 B0-EA 3B 66 F7-9B 4A C9 65
0001F8D0:  04 EF 57 55-5B 1A 93 A8-47 EF 97 22-9E 8D 5A 96
0001F8E0:  21 65 86 D7-44 6B 13 23-AE AE 3D 6C-42 B0 4D 44
0001F8F0:  2D AB 70 13-F3 32 B4 65-B2 3E 91 2D-90 D8 4C 16
0001F900:  6A EF BF 60-68 92 42 16-6E 39 9D 7A-DD 1C 01 89
0001F910:  00 00 00 00-00 00 00 00-00 00 00 00-00 00 00 00
```

©2022 TREND MICRO

Figure 1. An example of a DeadBolt-encrypted file in binary

Ransom Note

A file named “*!!!_IMPORTANT_README_WHERE_ARE_MY_FILES_!!!.txt”* is created on the infected device's target root directory. A ransom note is also shown when victims try to access the web administration page of their NAS devices. This is because DeadBolt replaces the legitimate CGI script to show this ransomware page. It is important to point out here that the prices, vendor names, and contact information were all manually crafted in our JSON configuration file, and such values do not reflect the actual values that DeadBolt victims will see in their systems:



WARNING: YOUR FILES HAVE BEEN LOCKED BY **DEADBOLT**

? What happened?

All **your files** have been encrypted. This includes (but is not limited to) Photos, Documents and Spreadsheets.

? Why Me?

This is not a personal attack. You have been targeted because of the inadequate security provided by your vendor [redacted]

? What now?

You can make a payment of (exactly) 0.1 bitcoin to the following address:
[redacted]

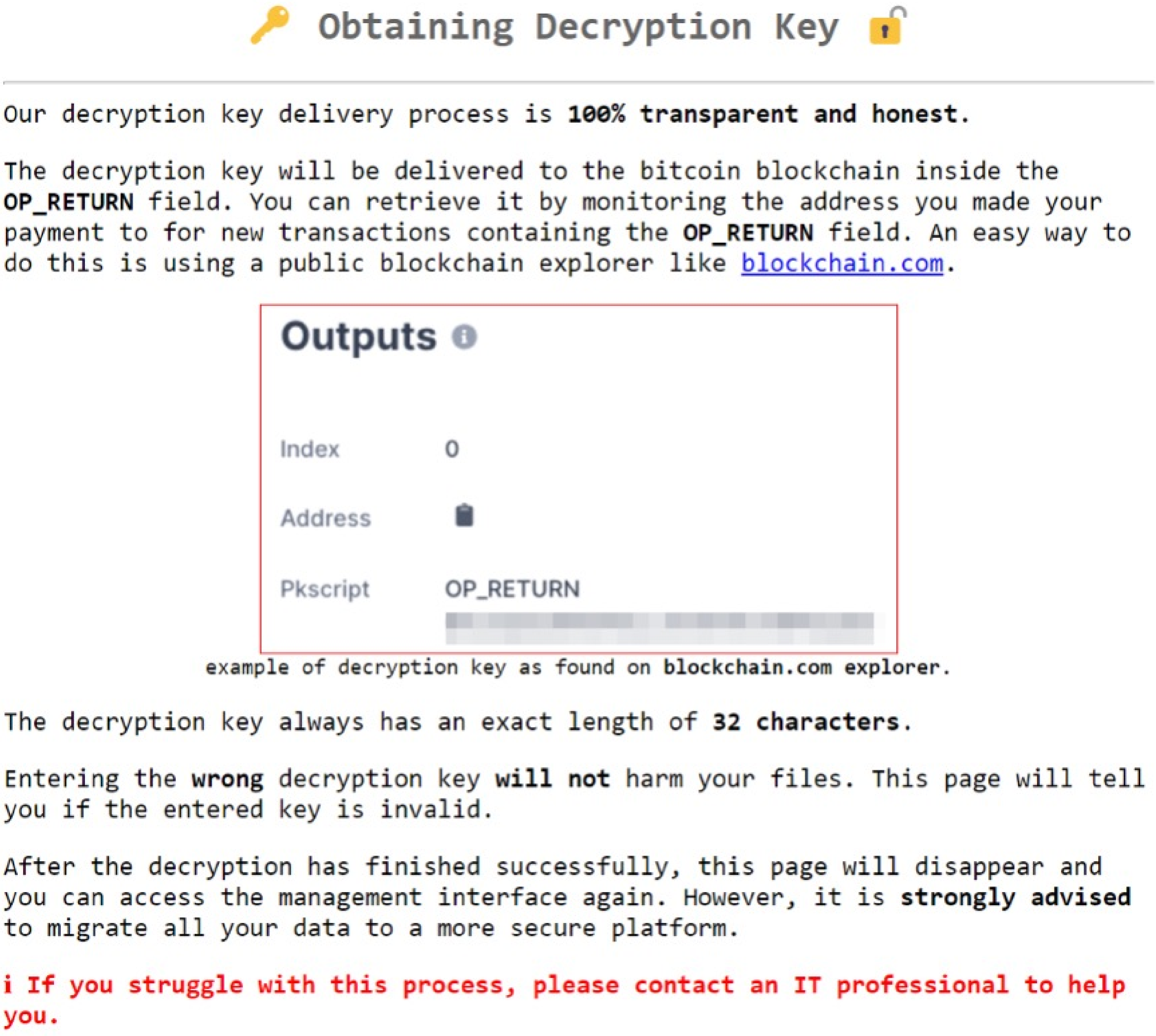
Once the payment has been made we'll follow up with a transaction to the same address, this transaction will include the **decryption key** as part of the transaction details. [more information]

You can enter the **decryption key** below to start the decryption process and get access to all your files again.

important message for [redacted]

Figure 2. The DeadBolt ransom note that appears onscreen when victims try to access the web administration page of their NAS devices

The links included in the ransom note open the following pop-up pages:



🔑 Obtaining Decryption Key 🔓

Our decryption key delivery process is **100% transparent and honest.**

The decryption key will be delivered to the bitcoin blockchain inside the **OP_RETURN** field. You can retrieve it by monitoring the address you made your payment to for new transactions containing the **OP_RETURN** field. An easy way to do this is using a public blockchain explorer like blockchain.com.

Outputs

Index	0
Address	📋
Pkscript	OP_RETURN [redacted]

example of decryption key as found on **blockchain.com explorer**.

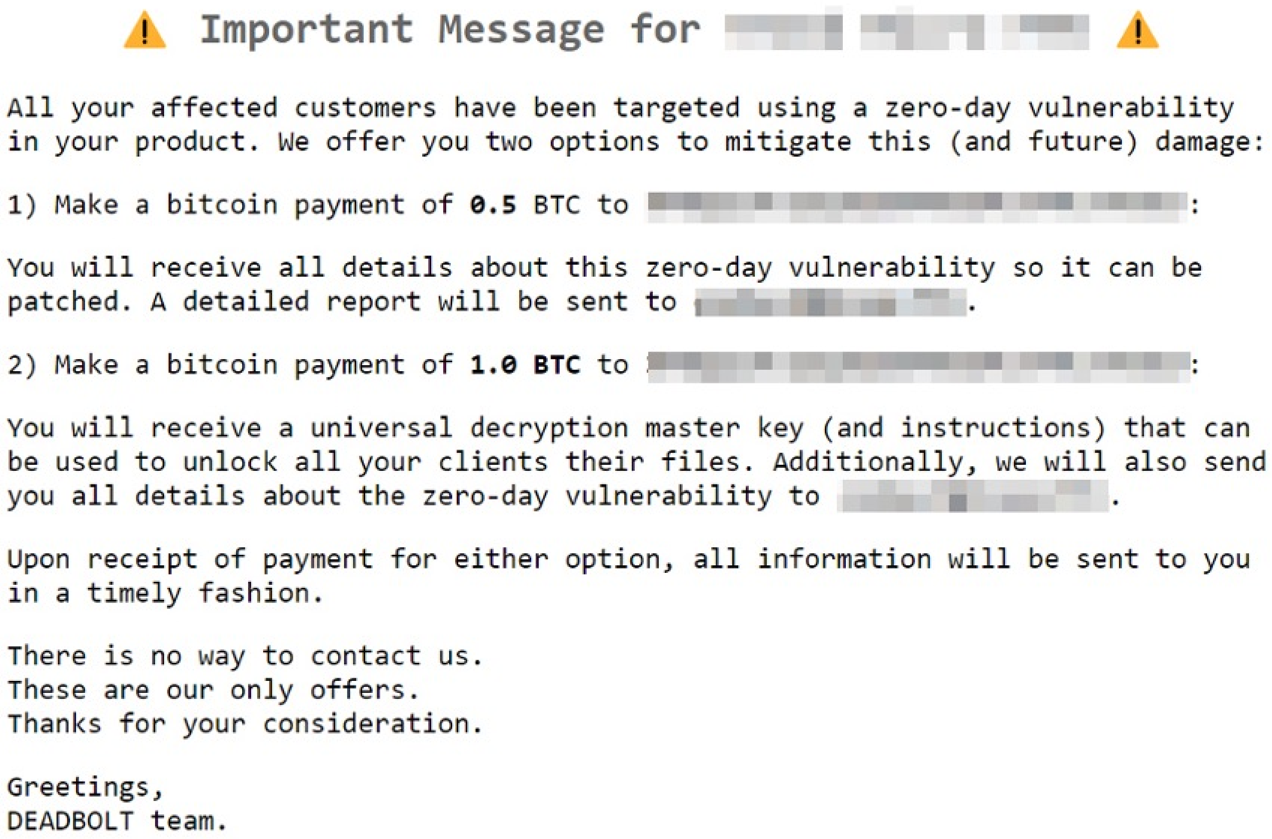
The decryption key always has an exact length of **32 characters**.

Entering the **wrong** decryption key **will not** harm your files. This page will tell you if the entered key is invalid.

After the decryption has finished successfully, this page will disappear and you can access the management interface again. However, it is **strongly advised** to migrate all your data to a more secure platform.

i If you struggle with this process, please contact an IT professional to help you.

Figure 3. A pop-up message that provides more information about DeadBolt’s decryption key



⚠ Important Message for [redacted] ⚠

All your affected customers have been targeted using a zero-day vulnerability in your product. We offer you two options to mitigate this (and future) damage:

1) Make a bitcoin payment of **0.5** BTC to [redacted]:

You will receive all details about this zero-day vulnerability so it can be patched. A detailed report will be sent to [redacted].

2) Make a bitcoin payment of **1.0 BTC** to [redacted]:

You will receive a universal decryption master key (and instructions) that can be used to unlock all your clients their files. Additionally, we will also send you all details about the zero-day vulnerability to [redacted].

Upon receipt of payment for either option, all information will be sent to you in a timely fashion.

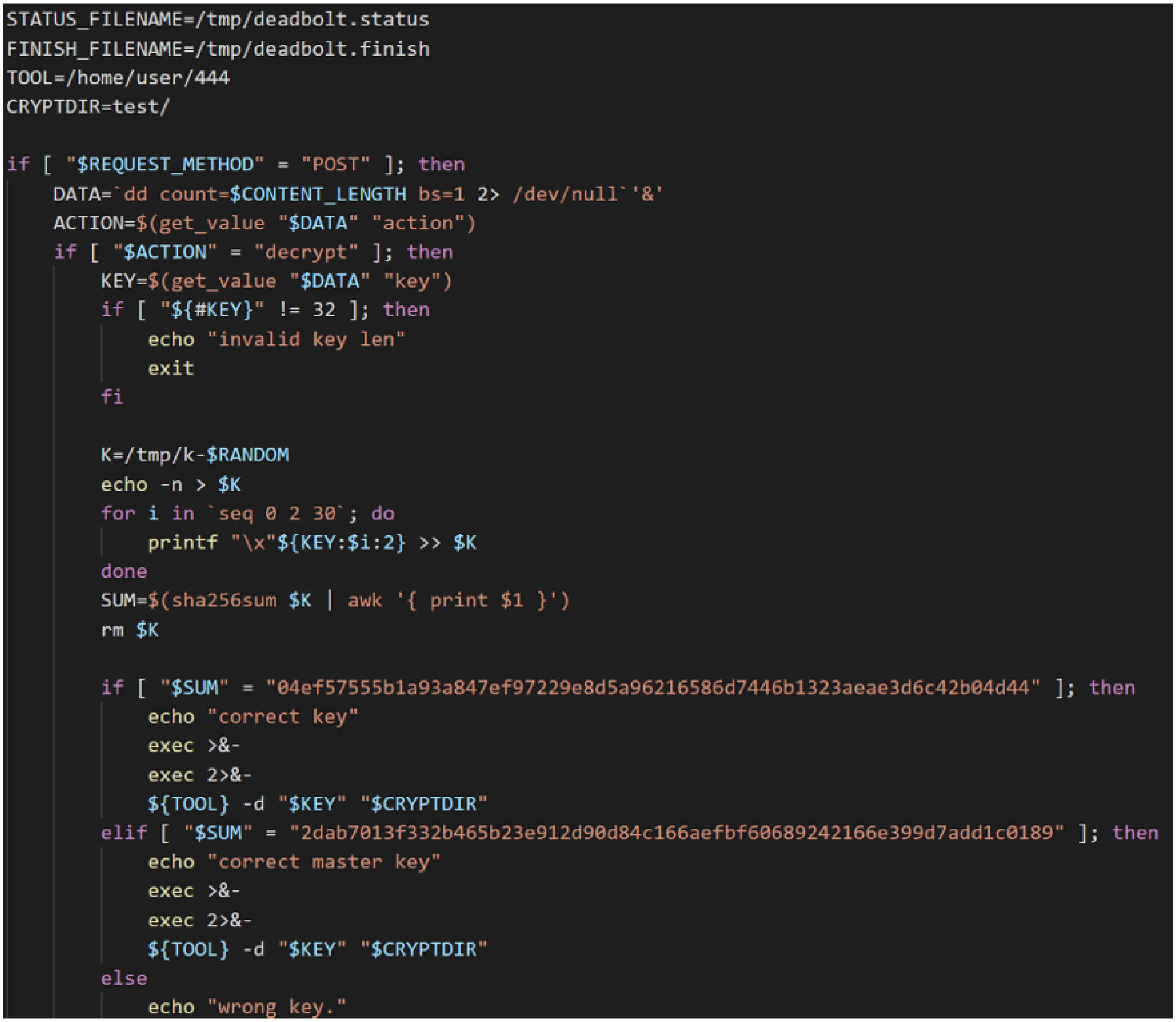
There is no way to contact us.
These are our only offers.
Thanks for your consideration.

Greetings,
DEADBOLT team.

Figure 4. A pop-up message for the NAS device vendor.

Decryption

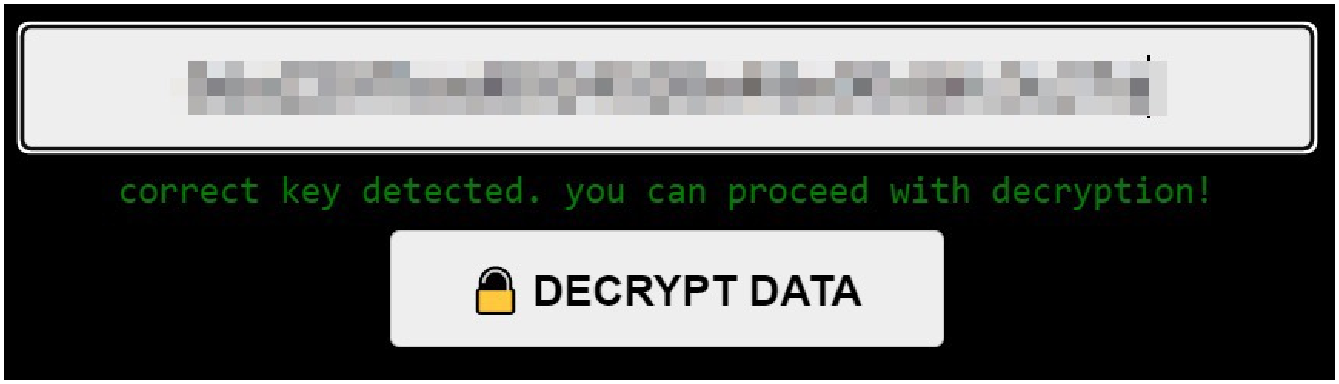
We verified that the decryption can be done with the correct key that was provided via the JSON file when the ransomware executable is run. Additionally, the previously shown web page has a feature that calls the ransomware executable by passing the provided key to it:



```
STATUS_FILENAME=/tmp/deadbolt.status
FINISH_FILENAME=/tmp/deadbolt.finish
TOOL=/home/user/444
CRYPTDIR=test/

if [ "$REQUEST_METHOD" = "POST" ]; then
    DATA=`dd count=$CONTENT_LENGTH bs=1 2> /dev/null`'&'
    ACTION=$(get_value "$DATA" "action")
    if [ "$ACTION" = "decrypt" ]; then
        KEY=$(get_value "$DATA" "key")
        if [ "${#KEY}" != 32 ]; then
            echo "invalid key len"
            exit
        fi

        K=/tmp/k-$RANDOM
        echo -n > $K
        for i in `seq 0 2 30`; do
            printf "\x"${KEY:$i:2} >> $K
        done
        SUM=$(sha256sum $K | awk '{ print $1 }')
        rm $K

        if [ "$SUM" = "04ef57555b1a93a847ef97229e8d5a96216586d7446b1323aeae3d6c42b04d44" ]; then
            echo "correct key"
            exec >&-
            exec 2>&-
            ${TOOL} -d "$KEY" "$CRYPTDIR"
        elif [ "$SUM" = "2dab7013f332b465b23e912d90d84c166aefbf60689242166e399d7add1c0189" ]; then
            echo "correct master key"
            exec >&-
            exec 2>&-
            ${TOOL} -d "$KEY" "$CRYPTDIR"
        else
            echo "wrong key."
```

Figure 5. Code that shows how the ransomware executable is called using the correct key

By using the correct key, victims can decrypt their files using the infected device’s web user interface (UI):



[redacted]

correct key detected. you can proceed with decryption!

Figure 6. DeadBolt victims can decrypt data on the DeadBolt web UI by encoding the correct key.

This is another example of how much effort DeadBolt actors have put into the development of this ransomware family. While other ransomware families use hard-to-follow steps that victims would need to take to get their data back, DeadBolt creators built a web UI that can decrypt victim data after ransom is paid and a decryption key is provided. The OP_RETURN field of the blockchain transaction automatically provides the decryption key to the victim once the ransomware payment is done. . This is a unique process wherein victims do not need to contact the ransomware actors — in fact, there is no way of doing so. Other ransomware families (such as CTB-Locker) have previously used this technique in its campaigns.

It should be noted that we were not able to verify how the alleged master key decryption works. In our tests, we found no evidence that such a decryption is even possible for files encrypted by DeadBolt. This is because AES is a symmetric encryption scheme and we have not seen any other data being added to the encrypted files. Notably, that the “master key” supplied via the configuration file is never used in the encryption process.

DeadBolt over time

Censys stated that they originally saw almost 5,000 infected services from DeadBolt. We looked into this data and saw that the number of infected DeadBolt systems has been decreasing. According to our data, the highest number of infections in March 2022. However, we observed that some systems replied with multiple HTTP titles. This indicates a ransomware infection, so it is possible to have more than one infection noted per device. For example, if a NAS device has both HTTP port 80 and HTTPS port 443 open, this single device would count for two infections.

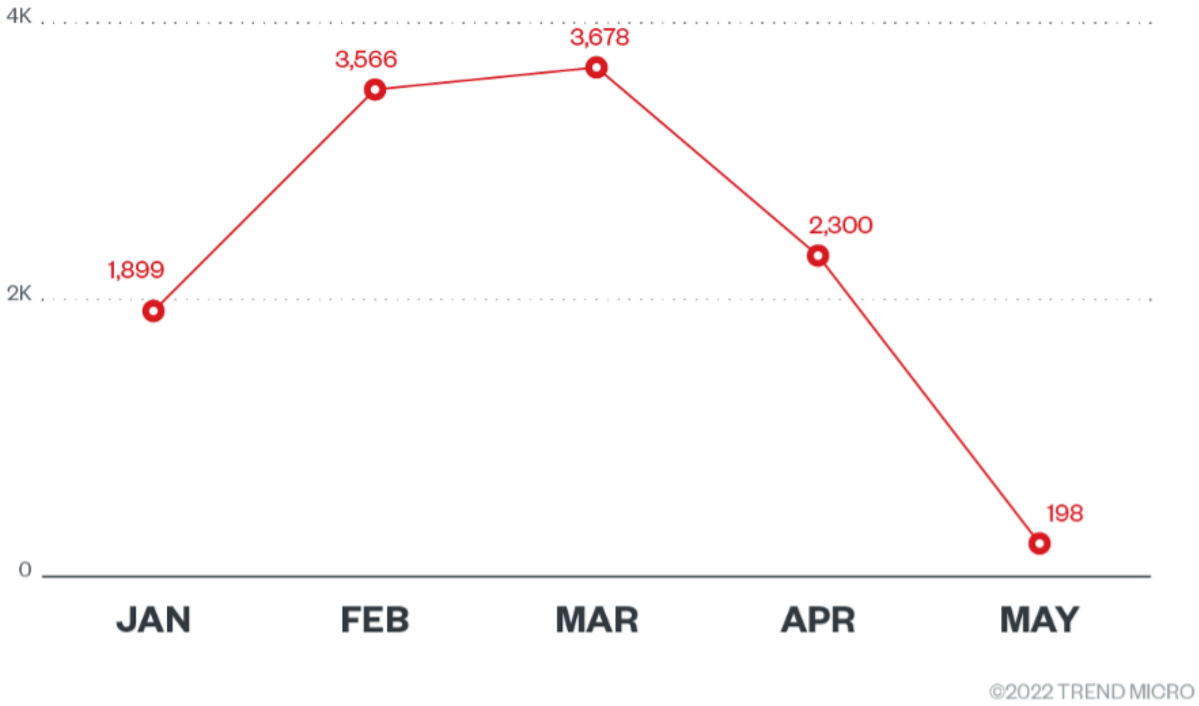


Figure 7. The total number of DeadBolt-infected services between January 1, 2022 and April 30, 2022 based on our telemetry

As we kept looking into the data, although both QNAP and ASUSTOR were targeted by DeadBolt, we found that most of the infections were on QNAP devices. There were only around 350 devices that were infected on ASUSTOR devices at the peak of the infections, and this number had gone down to 95 ASUSTOR internet-connected devices that are currently infected by DeadBolt. It’s worth remembering that a NAS infection does not equate to an endpoint infection. NAS devices frequently hold significant amounts of storage for their users, much of which might not be recoverable in the event of an attack.

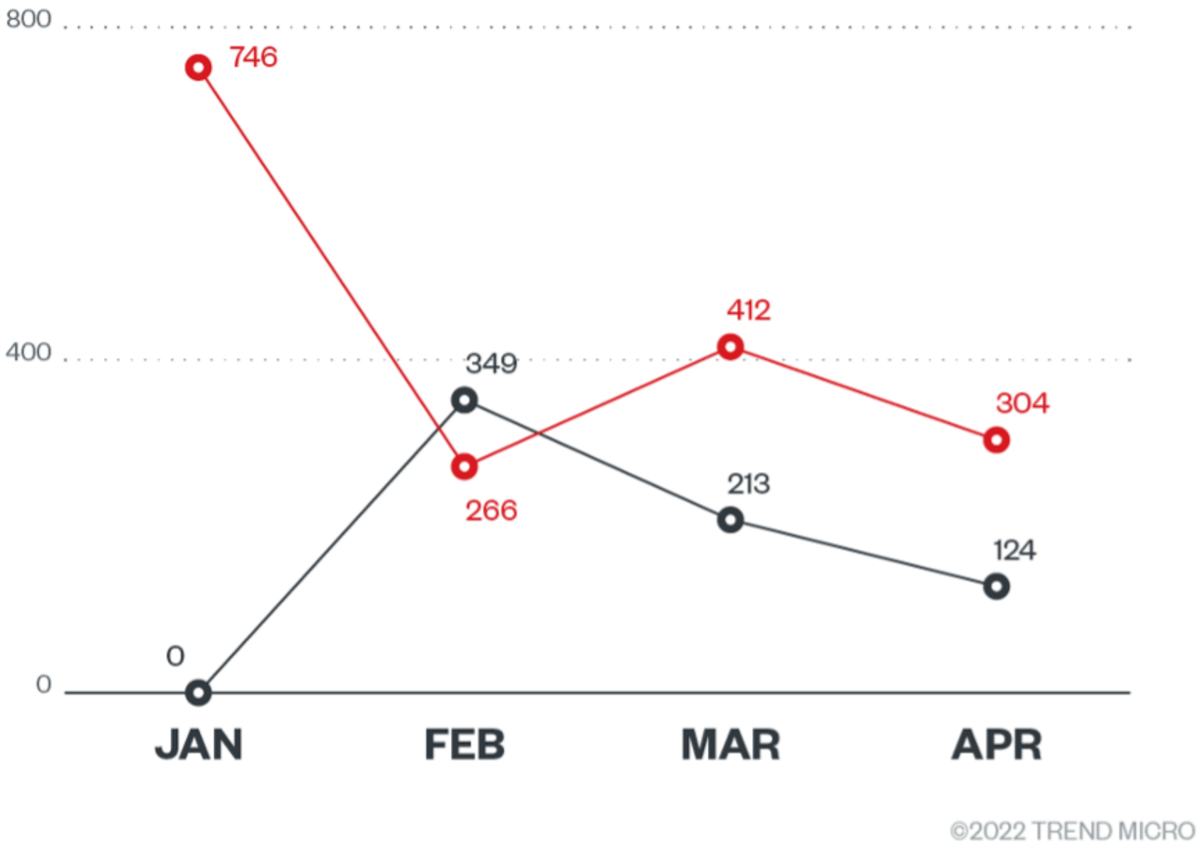


Figure 8. The number of DeadBolt infections by vendor SSL certificate from Jan. 1, 2022, to April 30, 2022

Even with at least 2,300 infected QNAP and ASUSTOR devices that are still connected to the internet, it should be noted that the number of infected devices is going down. This is probably because users are either taking their systems offline or are paying the ransom amount to get their files back. However, with an increasing number of ransomware families being used to attack NAS devices, the number of NAS devices exposed to the internet is becoming even more alarming. At the time of this writing, we found that there are over 2,500 ASUSTOR and over 83,000 QNAP internet-exposed services.

What can the economics and statistics tell us?

One unique facet of DeadBolt operations is that when victims pay the ransom, the decryption information is automatically put into the blockchain as part of the OP_RETURN section of a transaction. This is interesting because it allows us to see exactly when and for how much these payments were made.

For example, we observed DeadBolt actors charging 0.03 bitcoins for individual keys, 5 or 7.5 bitcoins for giving out vulnerability details, and 50 bitcoins for full vulnerability information and the master key. We also observed that unlike the more targeted business model of “big-game hunting” that most well-known ransomware families use, negotiating a ransom amount is not possible with Deadbolt. This is more common among other volume-focused ransomware because it’s simply not economical to directly interact with many victims. While the economics might be a bit dry, the amounts are worth detailing because they give us an idea of how these groups operate.

The fact that the price of 50 bitcoins (around US$1.9 million as of this publishing) is listed shows us the price that the ransomware group is aiming to obtain for this operation. This reveals that they never expected to make the US$4.4 million maximum amount that Censys projected. Let's take that logic a bit further and analyze DeadBolt’s success in pure business terms.

1,900,000 ÷ 4,400,000 = 0.43

Note: This percentage was calculated using the bitcoin to US dollar rate at the time of Deadbolt's peak in January.

It therefore appears that DeadBolt actors would have been more than happy if 43% of their victims paid ransom — or they never expected more than 40% of their victims to pay. In reality, only 8% of victims have paid to date. Based on our analysis, victims who paid DeadBolt's ransom did so within the first 20 days, and the number of victims who paid the ransom tapered off during the last 80 days. This data shows that the chances of people paying ransom decreases over time, so it is increasingly unlikely that more DeadBolt victims will pay the ransom amount after a certain period.

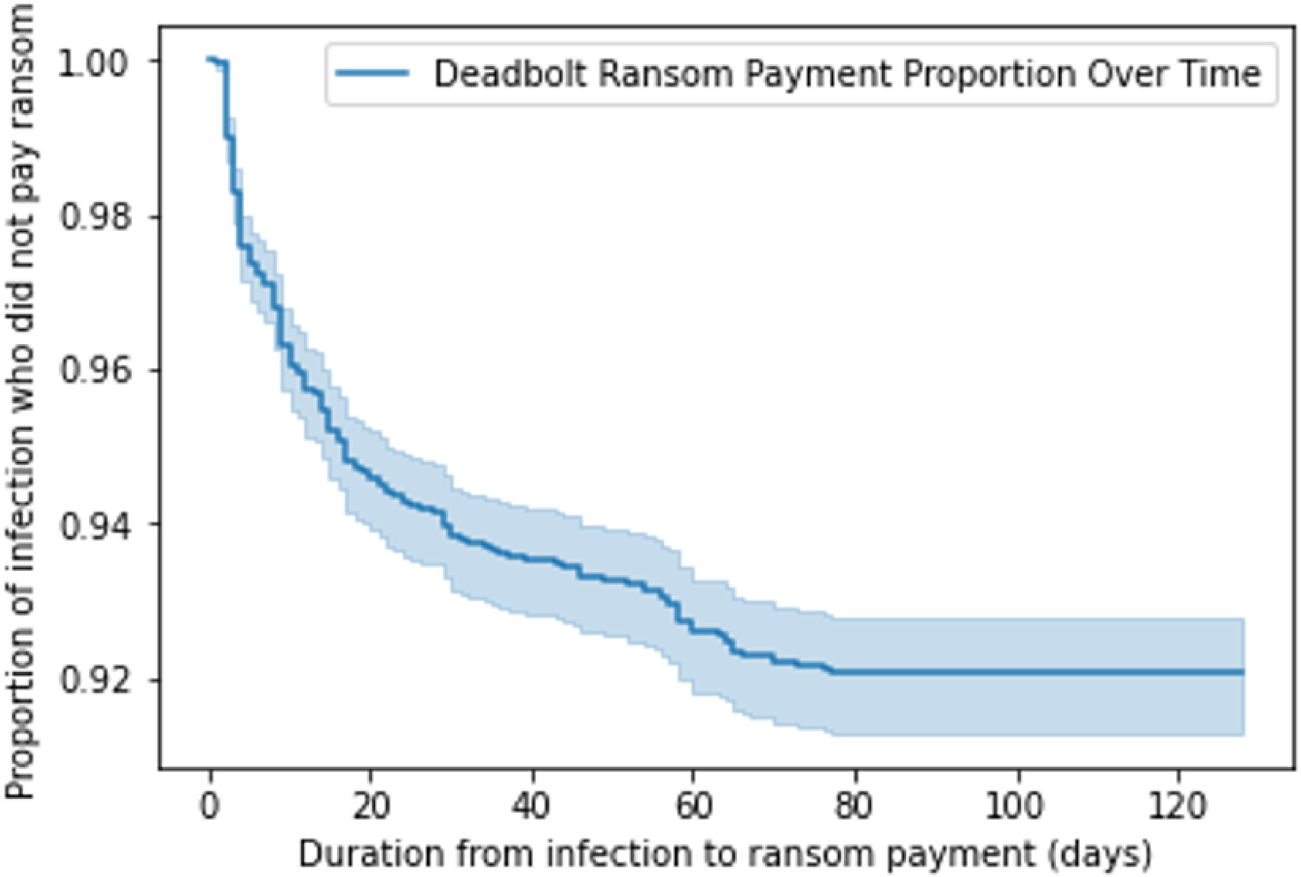


Figure 9. DeadBolt ransom payment proportion curve over time

The dark blue line in the survival analysis in Figure 8 shows the date range when victims paid the ransom amount. In this analysis, the victims that do not pay the ransom amount are referred to as survivors, while those who do are referred to as terminal. This analysis allows us to better understand the science of ransomware and ransom payout prevention.

We can go further and say that for about 5 to 7.5 bitcoins (roughly US$200,000 to US$300,000 as of this publishing), they would be willing to give away their methods — we are, however, only taking them for their word, which admittedly is on the charitable side. On the other hand, the charitable assumption on our end allows for this analysis. It's also possible that DeadBolt actors think that a conversion ratio of 6% (300,000 divided by 4,400,000) is substantial enough to cash out. They obviously know a lot more about payment ratios than we do, because they eventually topped out at 8%.

It's also clear that they knew in advance that US$300,000 would have been a good, low-risk deal. That in turn suggests that the entire operation cost them less than US$150,000, otherwise their profit margins would be undesirable. However, it's worth noting that the fact that 92% of victims have chosen not to pay ransom is an enormous success in cybersecurity — one that we often choose to ignore; instead, we tend to focus on how much ransomware actors have earned in their attacks.

Let’s try to understand the economic damage that DeadBolt has caused as best as we can. Presumably, for those who paid ransom, their financial losses would have been greater than 0.03 bitcoins (roughly US$1,000 at that time of publishing). For those who didn’t pay ransom, we can reasonably assume that their losses were lower, between zero to US$1,000. We can simplify the matter and suggest that their financial losses could be US$500 on the average.

(0.08 4,988 × 1,000) + (0.92 × 4,988 × 500) = 2,693,520

Based on this calculation, DeadBolt causes about US$2,693,520 worth of economic damage to earn US$300,000. It’s also interesting to think that the US$300,000 amount that they are asking for in exchange of the vulnerability details would probably be split among multiple members of the DeadBolt operation. Based on these numbers, DeadBolt actors are running the risk of incarceration for demanding millions of dollars from their victims, for a chance to earn only thousands, which doesn’t seem to be a sensible risk quantification.

Is it about the money, therefore, or about the damage caused? Are DeadBolt actors punishing society at large or just specific vendors? Or does this represent a refined business model that focuses on automation and volume, along with a chance to get a large single payout from affected vendors? These are some of the questions that we are left with after investigating ransomware groups such as DeadBolt.

Security recommendations

Users and organizations can keep their NAS devices secure by implementing the following security recommendations:

- **Regularly update your NAS devices.** Make sure that the latest patches have been installed as soon as they are available.
- **Keep NAS devices offline.** If you need to access your NAS device remotely, do it securely by opting to use either your NAS vendor’s remote access services (which most major NAS vendors offer) or use a virtual private network (VPN) solutions.
- **Use a strong password and two-factor authentication (2FA).** Do not use weak passwords or default credentials. If your NAS device supports 2FA, enable it to add an extra layer of protection against brute force attacks.
- **Keep your connection and ports secure.** Keep incoming and outgoing traffic secure by enabling HTTPs instead of HTTP. Remember to close all unused communication ports and change default ports.
- **Shut down or uninstall unused and out-of-date services.** Remove unused or out-of-date services to reduce the risk of NAS device compromise.

Conclusion

Overall, the total ransom amount that was paid was low in comparison to the number of infected devices, which led us to the conclusion that most people didn’t pay the ransom. It’s also worth pointing out that DeadBolt’s ransom amount costs more than the price of a brand-new NAS device, which is possibly why majority of its victims were not willing to pay to keep their data. Presumably, if the cost was higher, even more victims would be less likely to pay. The goal of DeadBolt actors is to infect as many victims as possible to get a decent payout or to get a vendor to pay one of the ransom options to get substantial financial payouts from its attacks.

Even though the vendor master decryption key did not work in DeadBolt’s campaigns, the concept of holding both the victim and the vendors ransom is an interesting approach. It’s possible that this approach will be used in future attacks, especially since this tactic requires a low amount of effort on the part of a ransomware group.

DeadBolt represents several innovations in the ransomware world: It targets NAS devices, has a multitiered payment and extortion scheme, and has a flexible configuration. But perhaps its main contribution to the ransomware ecosystem will be the legacy of its heavily automated approach. There is a lot of attention on ransomware families that focus on big-game hunting and one-off payments, but it's also important to keep in mind that ransomware families that focus on spray-and-pray types of attacks such as DeadBolt can also leave a lot of damage to end users and vendors.

Indicators of compromise (IOCs)

SHA-256	Detection
3c4af1963fc96856a77dbaba94e6fd5e13c938e2de3e97bdd76e1fca6a7ccb24	Ransom.Linux.DEADBOLT.YXCEP
80986541450b55c0352beb13b760bbd7f561886379096cf0ad09381c9e09fe5c	Ransom.Linux.DEADBOLT.YXCEP
e16dc8f02d6106c012f8fef2df8674907556427d43caf5b8531e750cf3aeed77	Ransom.Linux.DEADBOLT.YXCEP
acb3522feccc666e620a642cadd4657fdb4e9f0f8f32462933e6c447376c2178	Ransom.Linux.DEADBOLT.YXCEP
14a13534d21d9f85a21763b0e0e86657ed69b230a47e15efc76c8a19631a8d04	Ransom.Linux.DEADBOLT.YXCEP
444e537f86cbeeea5a4fcf94c485cc9d286de0ccd91718362cecf415bf362bcf	Ransom.Linux.DEADBOLT.YXCEP

Yara rules

```
rule deadbolt_cgi_ransomnote : ransomware {
meta:
description = "Looks for CGI shell scripts created by DeadBolt"
author = "Trend Micro Research"
date = "2022-03-25"
hash = "4f0063bbe2e6ac096cb694a986f4369156596f0d0f63cbb5127e540feca33f68"
hash = "81f8d58931c4ecf7f0d1b02ed3f9ad0a57a0c88fb959c3c18c147b209d352ff1"
hash = "3058863a5a169054933f49d8fe890aa80e134f0febc912f80fc0f94578ae1bcb"
hash = "e0580f6642e93f9c476e7324d17d2f99a6989e62e67ae140f7c294056c55ad27"
strings:
$= "ACTION=$(get_value \"$DATA\" \"action\")"
$= "invalid key len"
$= "correct master key"
$= "'{\"status\":\"finished\"}'"
$= "base64 -d 2>/dev/null"
condition:
uint32be(0) != 0x7F454C46 // We are not interested on ELF files here
and all of them
}
rule deadbolt_uncompressed : ransomware {
meta:
description = "Looks for configuration fields in the JSON parsing code"
author = "Trend Micro Research"
date = "2022-03-23"
hash = "444e537f86cbeeea5a4fcf94c485cc9d286de0ccd91718362cecf415bf362bcf"
hash = "80986541450b55c0352beb13b760bbd7f561886379096cf0ad09381c9e09fe5c"
hash = "e16dc8f02d6106c012f8fef2df8674907556427d43caf5b8531e750cf3aeed77"
strings:
$= "json:\"key\""
$= "json:\"cgi_path\""
$= "json:\"client_id\""
$= "json:\"vendor_name\""
$= "json:\"vendor_email\""
$= "json:\"vendor_amount\""
$= "json:\"payment_amount\""
$= "json:\"vendor_address\""
$= "json:\"master_key_hash\""
$= "json:\"payment_address\""
$= "json:\"vendor_amount_full\""
condition:
elf.type == elf.ET_EXEC
and all of them
}
```