

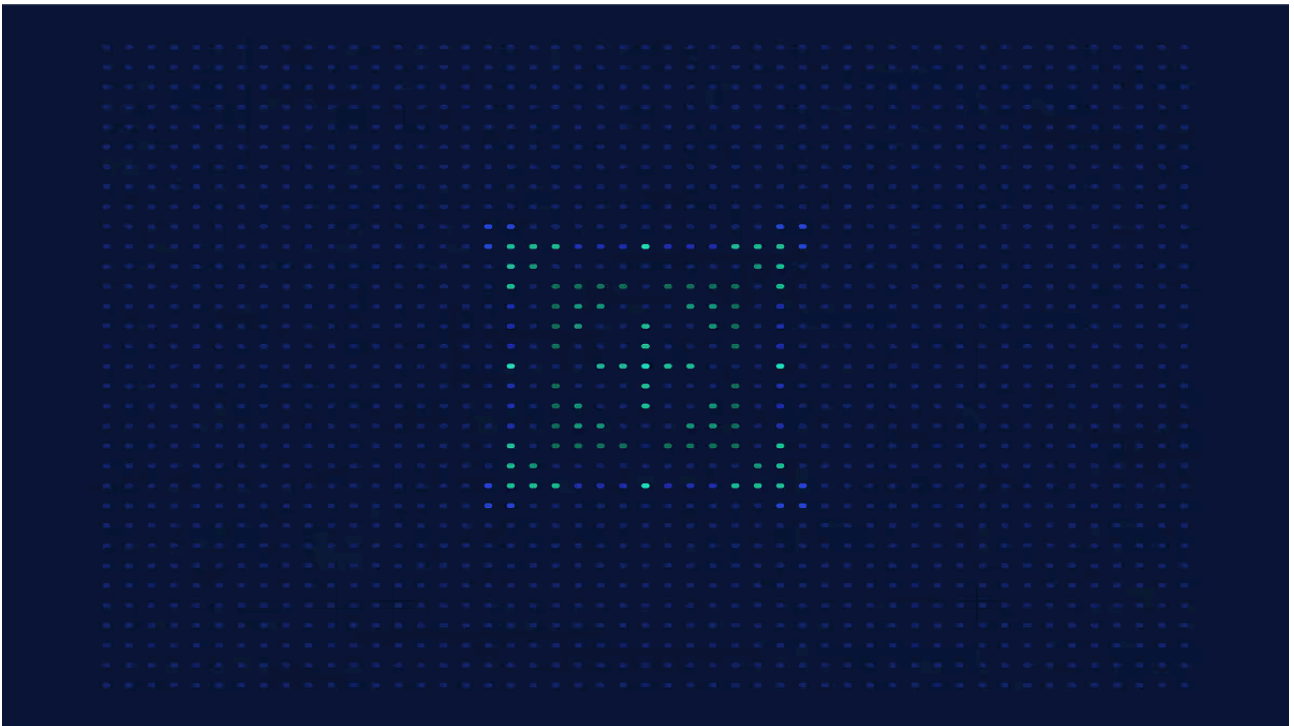
# Anomali Threat Research Releases First Public Analysis of Smaug Ransomware as a Service

By Anomali Threat Research

Published: 2026-03-12 · Archived: 2026-04-05 13:20:18 UTC

Anomali Threat Research publishes a public analysis of Smaug RaaS, including TTPs, indicators, and practical guidance to reduce ransomware risk.

- [IntroductionThreat ActorSmaug\\_PanelThe RansomwareAnalysisConclusionHow Anomali HelpsMitre ATT&CKEndnotes](#)



*Authored by: Joakim Kennedy and Rory Gould*

Anomali ThreatStream customers can find Indicators of Compromise (IOCs), signatures, and [more information about this threat here](#).

## **Introduction**

Threat actors and cybercriminals that don't have the ability to develop their own ransomware for malicious campaigns can turn to the Smaug Ransomware as a Service (RaaS) offering, which is available via a Dark Web Onion site. At least two threat actors are operating the site, providing ransomware that can be used to target

Windows, macOS, and Linux machines. The site is built with ease of use in mind. To launch an attack, threat actors simply need to sign up, create a campaign, and then start distributing the malware. The site also handles decryption key purchasing and tracking for victims.


## Threat Actor

The Smaug RaaS appears to be operated by one to two threat actors that are both active on the criminal underground. One of the threat actor's online handles on the forums has been identified. The second operator still is unknown.

## Forum Activity

On May 5, 2020, an actor named **corinda** posted on the 'Exploit.in' forum advertising a new RaaS dubbed 'Smaug.' The post (figure 1.) showcased Smaug's features and included screenshots of the Smaug UI, this is detailed in the **Panel** section below. The post also directed users as to how they could avail of the service: contacting [smaug-ransomware@protonmail.com](mailto:smaug-ransomware@protonmail.com) with a registration fee of 0.2 BTC (appx \$1,900 (USD) at time of posting) and subsequent service fees of 20%. The actor was willing to waive the registration fee for the first five users who could demonstrate their skills with the product; this was likely to garner reputation on the forum given that **corinda's** reputation score was zero.

**Corinda** continued to interact with posters before the topic was locked by a forum admin on May 14, as **corinda** had not deposited money on the forum. There has been no subsequent activity from **corinda** and the profile has not been active since June 6.

 **Initial Smaug offering on Exploit.in** **Figure 1 - Initial Smaug offering on Exploit.in**

## Account History

**Corinda's** Exploit account was created on May 17, 2019 as a paid registration (paid registrations are completely anonymous and do not link to other accounts on 'friendly forums'), and remained dormant until Oct. 2019. On Oct. 6, 2019 **corinda** created their first post:

*'I look for front end dev. Must be able to design professional websites  
Must be able to show previous work or prove skills otherwise  
Fulent english mandatory  
Budget \$2000 in BTC  
PM for details'*

The only reply to the topic was from **corinda** three days later on 10/9:

*'I am still looking. Do not be shy'*

These posts were recovered through Sixgill as the original topic has since been deleted and is no longer available on the forum.

## Smaug Panel

Smaug developers shared screenshots of the product on their Onion site, it offers some insight as to how the product is to be used.

## Registration

The site offers a registration form replete with captcha. Upon completion, users are directed to pay 0.2 BTC to a specified Bitcoin wallet. The actor claims that accounts will be activated within 10 minutes of payment after the payment has received six confirmations.

 **Registration form**

*Figure 2 - Registration form*

## Dashboard

### Create Campaign

After successful registration, the user is taken to the Smaug dashboard, which appears as a clean, utilitarian UI with clear directions. From this page, a user can create campaigns with custom ransom messages and expiration dates; Smaug also allows users to distinguish between 'Regular' and 'Company' campaigns. A 'Regular' campaign will necessitate a decryption key for every victim infected, while a 'Company' campaign will allow a single key to decrypt all computers affected by the campaign.

 **Campaign creation**

*Figure 3 - Campaign creation*

After the expiration date set by the user has passed, the campaign will expire and the victims can no longer recover their files.

### Manage Campaign

With a campaign created, a user can download the relevant payload for the system(s) they are attempting to infect; Mac, Linux and Windows are all available. The dashboard also will keep track of how many systems have been infected, this is done through allowing a victim one free decryption of a single file, this is then recorded and updates the dashboard. The interface allows the user to track the campaign dates and profits they have accumulated.

 **Campaign management**

*Figure 4 - Campaign management*

## Fund Withdrawal

Once a victim has paid the ransom, the user is able to navigate to the 'Withdrawal' page. This dashboard will present the user with their current balance (the 20% fee has been automatically withheld) and number of payments made to them. From here, the user simply creates a withdrawal to a specified BTC wallet and removes the funds.

## Fund overview

*Figure 5 - Fund overview*

## Support

The Smaug developers claim to run a support service to assist with bugs and other issues. They also claim to take feature requests, they will create them for a fee or incorporate them for free if enough users request it.

## Support service

*Figure 6 - Support service*

## Settings

The settings page is bare-boned and allows for users to change passwords.

## Settings page

*Figure 7 - Settings page*

## The Ransomware

The Smaug ransomware is written in Go (Golang). Anomali Threat Research has identified samples targeting both Windows and Linux machines. The malware has a relative simple design implemented in about 300 lines of code. Its source code structure is shown in Figure 8 below. The malware creates a unique key for each machine that is used to encrypt the files. The key is encrypted by the threat actor's RSA public key which means the machine key can only be encrypted by a threat actor's private key. The ransomware is designed to run completely offline without the requirement of a network connection.

## Extracted source code structure from a malware sample

*Figure 8 - Extracted source code structure from a malware sample.*

Go binaries do not include a compilation timestamp. Consequently, it is difficult to gauge the age of a sample. The analyzed sample was compiled by version 1.14.2 of the official Go compiler. This compiler was released on April 8, 2020 and was uploaded to a public malware repository on May 31, 2020. At the time of it being uploaded to the public repository, approximately 11% of the antivirus engines detected it as malicious. The earliest sample was uploaded on May 11, 2020 and was only detected by one antivirus engine, while the Linux sample was uploaded on May 19, 2020 and as of this writing date is not detected by any engines.

## Key Generation

Smaug generates a unique encryption key for each infected machine. First, the malware gets a timestamp of the current time. The Unix time in nanoseconds of this timestamp is used to set the global seed for the “math/rand” package in the standard library for Go.<sup>[1]</sup> This action has no effect on the randomness as the malware, after this action, generates a 32 byte slice of randomness via the call to “crypt/rand.Read”.<sup>[2]</sup> This function on Windows uses the “CryptGenRandom” API to generate the random data. The random data is encrypted with the threat actor’s 2048 bit key using RSA Optimal Asymmetric Encryption Padding (OAEP).


## File Encryption

The malware uses a pair of “goroutines“ for each disk partition for the encryption of files. A goroutine is Go’s implementation of coroutines for concurrency. The first routine scans the partition for all files and compares the file extension against the list of extensions shown in Figure 9 below. If the file has an extension that is in the list, the file is added to a list of files to be encrypted.

```
*.3fr *.acddb *.aes *.ai *.ARC *.arw *.asc *.asf *.asm *.asp *.avi *.backup *.bak *.bat *.bay *.bmp  
*.brd *.c *.cdr *.cer *.cgm *.class *.cmd *.cpp *.cr2 *.crt *.crw *.csr *.CSV *.dbf *.dch *.dcr *.der  
*.dif *.dip *.djv *.djvu *.dng *.doc *.DOC *.docb *.docm *.docx *.dot *.DOT *.dotm *.dotx *.dwg *.dxf  
*.dxg *.eps *.erf *.fla *.flv *.frm *.gif *.go *.gpg *.gz *.h *.html *.hwp *.ibd *.indd *.jar *.java  
*.jpe *.jpeg *.jpegm *.jpg *.kdc *.key *.lay *.lay6 *.ldf *.max *.mdb *.mdf *.mef *.mid *.mkv *.mml  
*.mov *.mpeg *.mpg *.mrw *.ms11 *.MYD *.MYI *.nef *.NEF *.nrw *.odb *.odg *.odm *.odp *.ods *.odt  
*.orf *.otg *.otp *.ots *.ott *.p12 *.p7b *.p7c *.PAQ *.pas *.pdd *.pdf *.pef *.pem *.pfx *.php *.png  
*.pot *.potm *.potx *.ppam *.pps *.ppsm *.ppsx *.ppt *.PPT *.pptm *.pptx *.psd *.pst *.ptx *.py  
*.qcow2 *.r3d *.raf *.rar *.raw *.rb *.rtf *.RTF *.rw2 *.rwl *.sch *(Security copy) *.sldm *.sldx  
*.slk *.sql *.sqlite3 *.SQLITE3 *.sqlitedb *.SQLITEDB *.srf *.srw *.stc *.std *.sti *.stw *.svg *.swf  
*.sxc *.sxd *.sxi *.sxm *.sxw *.tar *.tar.bz2 *.tbk *.tgz *.tif *.tiff *.txt *.uop *.uot *.vbs *.vdi  
*.vmdk *.vmx *.vob *.wav *.wb2 *.wks *.wma *.wmv *.wpd *.wps *.xlc *.xlk *.xlm *.xls *.XLS *.xlsb  
*.xlsm *.xlsx *.xlt *.xltm *.xltx *.xlw *.xml *.zip
```

**Figure 9** - List of file extensions the malware uses to determine if the file should be encrypted or not.

The second routine encrypts all the files in the list identified by the other routine. Each file is encrypted with AES in CBC mode. Figure 10 below shows the setup of the encryption handler for each file. A new AES cipher handler is constructed, an Initialization Vector (IV) is generated by reading 16 bytes from “crypt/rand”, before a CBC handler is created.

 **Screenshot of the routine creating the encryption handler. The files are encrypted with AES in CBC mode. The last four instructions create a 4k buffer that is used to encrypt the file 4k bytes at the time.**

**Figure 10** - Screenshot of the routine creating the encryption handler. The files are encrypted with AES in CBC mode. The last four instructions create a 4k buffer that is used to encrypt the file 4k bytes at the time.

The files are encrypted 4k bytes at a time. If the file is smaller or the read returns less than 4k, the malware adds padding to the end of the file to make the total bytes a multiple of 16 bytes as required by AES. The padding format used is PKCS#7. The encrypted blocks are written to a new file that has the original filename with an unique identifier added to the end of the filename. The original file is removed after the file has been encrypted.

The ransomware writes the ransom note in all the folders with encrypted files. In Figure 11 below, the logic for writing the note is shown. In the analyzed sample, the ransom note’s name is “HACKED.txt”.

 **Logic for writing the ransom note to the disk.**

**Figure 11** - Logic for writing the ransom note to the disk.

The ransom note is stored in the binary as a Base64 encoded string. Figure 12 shows the decoded ransom note.

Your files have been encrypted using military grade encryption. They can never be accessed again without buying a decryption key. You can buy the decryption key at <http://smaugrwmaystthfxp72tldmbrzlwdp2pxtpvtzvkhv5ppg3difiwonad.onion>. To access the site you need Tor Browser.

**Figure 12** - Base64 decoded ransom note stored in the analyzed sample.

## Analysis

The marked difference in quality of English between **corinda's** contracting post and the Smaug offering would indicate two different authors. The original post was likely created by the core individual(s) who were developing Smaug, whilst the Smaug offering post was likely created by a fluent English-speaking contractor.

The spelling and grammar mistakes present in the initial post and the need for a fluent English-speaking front end dev would indicate that this actor does not come from an English-speaking country. **Corinda** specified that targeting Commonwealth of Independent States (CIS) countries was forbidden and would result in a ban from the service. However, this does not necessarily indicate that the actor's location is in a CIS state as "Exploit.in" is a Russian language cybercrime site that bans the targeting of CIS countries. It is worth noting that on the Smaug Onion site a screenshot of a ransomware message is presented (Figure 13) with Mandarin characters predominantly displayed.

 **Mandarin characters visible on ransom note**

**Figure 13** - Mandarin characters visible on ransom note

**Corinda's** offer of Smaug did not appear to gain much traction on the forum leading to a follow up discount of a 15-day trial to reputable forum members, likely to boost reputation. The only subsequent activity on the thread was other users asking if anyone could vouch for the legitimacy of the product. After ten days (5/14/20), a forum admin locked the thread and requested that **corinda** deposit \$8,000 to the forum's escrow account, this is a common practice to protect forum members from exit scams. **Corinda** did not comply and the thread has remained inactive since, however, the contact details are still available for users to purchase Smaug from **corinda**. Due to a lack of history, interactions and feedback, we cannot corroborate **corinda's** claim and rate it as *unreliable*.

The Smaug ransomware is relatively-simple compared to other active ransomware. For example, it lacks the functionality of stopping running processes which are used by ransomware to release file locks, allowing for encryption of locked files. It also does not delete backups and shadow copy files on Windows. This flaw will make it possible to recover the encrypted files from the shadow copy if the service is enabled. The simplicity of the malware allows the author to target multiple operating systems from the same code base. The Windows and Linux samples have been compiled from the same source code with no modifications, except the shim-layers provided by Go's standard library.

## Conclusion

Smaug is a RaaS that makes it easy for threat actors to use ransomware to achieve objectives. The ransomware can run on all the three major operating systems that opens up the potential for broader targeting. The actual

ransomware is relatively simple compared to common ransomware currently being used by other threat actors. The malware's only functionality is the encryption of files. Threatstream Enterprise users can find [Indicators of Compromise \(IOCs\), signature, and more information about this threat here](#).

## How Anomali Helps

Anomali Threat Research provides actionable threat intelligence that helps customers, partners, and the security community to detect and mitigate the most serious threats to their organizations. The team frequently publishes threat research in the form of white papers, blogs, and bulletins that are made available to the security community, general public, and news organizations. Intelligence and bulletins about threat actors and related Indicators of Compromise (IOCs) are integrated directly into Anomali customers' security infrastructures to enable faster and more automated detection, blocking, and response. For more information on how Anomali customers gain integrated access to threat research, visit: <https://www.anomali.com/products>.

## Mitre ATT&CK

- T1486 - Data Encrypted for Impact

## Endnotes

1. "rand package · pkg.go.dev", accessed July 7, 2020, <https://pkg.go.dev/math/rand?tab=doc#Seed>
2. "rand package · pkg.go.dev", accessed July 7, 2020, <https://pkg.go.dev/crypto/rand?tab=doc#Read>



## Iran's IRGC Names Western Tech Giants as "Legitimate Targets": What CISOs Must Do Now



**When 766 Systems Fall in 24 Hours: The Threats Bearing Down on State Government Networks**



**The Iran Cyber Threat Machine Isn't Slowing Down — Here's What CISOs Need to Know Now**

Source: <https://www.anomali.com/blog/anomali-threat-research-releases-first-public-analysis-of-smaug-ransomware-as-a-service>