Trapping the Netwire RAT on Linux

By susannah.matt@redcanary.com

Published: 2020-01-30 · Archived: 2026-04-05 23:17:15 UTC

Adversaries today have a slew of remote access trojans (RAT) to choose from, ranging from .NET tools for Windows to cross-platform RATs that work across multiple operating systems, such as CrossRAT, Pupy, and Netwire. While public research abounds examining the Windows and macOS versions of Netwire, the Linux variety is considerably less well known. Today we want to shed some extra light on Netwire for Linux.

Intro to Netwire

Netwire is a RAT distributed by World Wired Labs and marketed as a remote management tool. It allows remote access to Windows, macOS, Linux, and Solaris systems, and is primarily used to transfer files and conduct system management in multiple ways.

Once you go beyond the initial veneer of legitimacy, you may notice some additional features that aren't as benign. These include:

- keylogging
- masquerading network traffic with a browser user-agent string
- capturing screenshots
- accessing credentials in web browsers.

Netwire allows the people using it to pivot their traffic through proxies and listen on whatever ports they need to receive encrypted command and control. A quick summary of the ATT&CK techniques used by Netwire are listed in a table at the end of this post, with the Linux-specific ones in bold.

The notes section in the table includes links to Atomic Red Team tests, which can help you test your security controls with commands similar to the ones Netwire would actually use.

The Linux Specifics

Most of the functionality of Netwire is the same across platforms, with some minor exceptions. The Windows Registry doesn't exist on non-Windows systems, so the persistence mechanisms have to change. In addition, the binary formats will be different across platforms. The Windows and macOS versions use Portable Executable and Mach-O, respectively. For Linux and Solaris, the binaries are in Executable and Linkable Format (ELF).

The persistence mechanisms also change, offering the options to use XDG Autostart Entries and crontabs for persistence. We've waxed lyrical about crontabs before, but we haven't explored XDG Autostart Entries in detail. These artifacts are similar to the Start Menu Startup Items for Windows.

On any Unix-like system that uses a Freedesktop.org XDG-compliant desktop, you can add an autostart entry to execute an application or script when the desktop loads. These files usually exist under these folders, although you can change them with environment variables:

- `/etc/xdg/autostart`
- `~/.config/autostart`

The Autostart Entry itself usually has a file extension of '.desktop.' Its contents will look something like this:

```
[Desktop Entry]
Type=Application
Exec=/home/user/.config/dbus-notifier/dbus-inotifier
Name=system service d-bus notifier
```

Netwire isn't the only tool that uses this persistence mechanism: it has been used by Fysbis, Pupy, jRAT, and CrossRAT. In the case of Netwire, users can specify the name of the Autostart Entry and make it masquerade as something like a Linux system process if desired.

Into the Lab!

We ran across this Netwire sample on VirusTotal, with all of the other malware in the world.

We decided to download it and throw it into the test lab. At the bottom of the page we'll include a link to the sample we used for analysis.

After executing the sample, we noticed the sample copied itself to a hidden folder and launched from the hidden folder. This is a good first step to hide itself from casual observation on disk.

This functionality corresponds to a configuration by the operator to install a copy of the RAT to a local folder for longer-term residence.

Next, additional telemetry shows the creation of Netwire's .desktop persistence mechanism and the creation of two additional files.

The file `/tmp/.rOuYXzdOF` was most likely used as a mutex, ensuring only one copy of Netwire could run at a time. Next, `.default.conf` was a configuration file storing data required for Netwire to communicate with command and control. On the Windows side, this is usually stored in the Registry. Finally, network connections were established for control by an adversary.

Attributing to Netwire

We worked with this sample from VirusTotal.

For attribution to Netwire, we relied on a few data points. First, we took note of the antivirus detection rate and classification in VT. Next, we leaned on Patrick Wardle's analysis of a Netwire variant for macOS. In the post, he provided several strings of interest that were extracted from the macOS Netwire sample. When we compared these strings against the Linux variant, we found 14 common strings that correspond with a user-agent string, network configuration discovery, and a rather unique string that may correspond to a session ID or password. The strings that matched are below:

- `/bin/bash`

- `/bin/sh`
- `/tmp/.%s`
- `Accept-Language: en-US,en;q=0.8`
- `Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8`
- `CONNECT %s:%d HTTP/1.0`
- `Current IP Address:`
- `GET / HTTP/1.1`
- `Host: %s:%d`
- `Host: checkip.dyndns.org`
- `User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; Trident/7.0; rv:11.0) like Gecko`
- `checkip.dyndns.org`
- `exit`
- `hyd7u5jdi8`

Finally, we matched up the remaining strings with the functionality of Netwire mentioned in a user manual available online. The release notes for a newer version of Netwire mentioned crontab support for persistence, and we observed a string indicating the agent might use a `crontab -l` command, although none was observed during execution earlier. We also observed strings indicating the use of HTTP and SOCKS proxy functions that were described in the Netwire manual. We assert with high confidence that this sample is a Netwire variant and is representative of a newer version.

Bringing a lesser known variant of malware to the public is always fun, and hopefully the details shared here will help prevent or detect RATs in your environment!

Source: https://redcanary.com/blog/netwire-remote-access-trojan-on-linux/