P2PInfect Worm Evolves, Targeting a New Platform

nozominetworks.com/blog/p2pinfect-worm-evolves-to-target-a-new-platform

Nozomi Networks

P2PInfect Worm Evolves to Target a New Platform

by

Nozomi Networks Labs

|

January 16, 2024

A highly sophisticated strain of malware known as P2PInfect is raising new concerns in the cybersecurity community. Developed in Rust, a language known for its safety and efficiency, this cross-platform worm uses several different methods of propagation to infect devices powered by different architectures. The malware is capable of performing Peer-to-Peer (P2P) communications without relying on a single Command and Control server (C&C) to propagate attackers' commands. Some strains attempt to abuse SSH while others exploit vulnerabilities in Redis, a popular in-memory database, to spread rapidly across networks.

The most intriguing aspect of P2PInfect is its evolving nature; as researchers dig deeper to understand its goals, the worm continues to adapt, and expand to target new architectures. Nozomi Networks Labs has identified a strain of P2Pinfect that targets a new IoT architecture

- ARM.

This blog provides a comprehensive overview of recent P2PInfect worm operations and behavior, along with how they have changed over time. We analyze a new set of samples to investigate the worm's defence techniques, and provide detections at the end of the blog. By delving into the technical details and ongoing efforts to mitigate its threats, we aim to inform and educate readers about this emerging cybersecurity threat and the broader implications the malware holds for digital security in our interconnected world.

Tracking the Timeline of P2PInfect Strains

According to our data, the first strains of this threat date back to at least July 2023. Malware authors have been gradually introducing support for more and more platforms mainly focusing on the x86-64 platform of CPUs. Redis is only officially supported in Linux, but P2PInfect creators are designing malware to spread to as many platforms as possible. It is also feasible to install Redis in Windows machines using the Windows Subsystem for Linux (WSL) and the P2PInfect developers are aware of this. For this reason, the actors also ship Windows DLLs inside the ELF binaries to increase the number of devices they can infect.

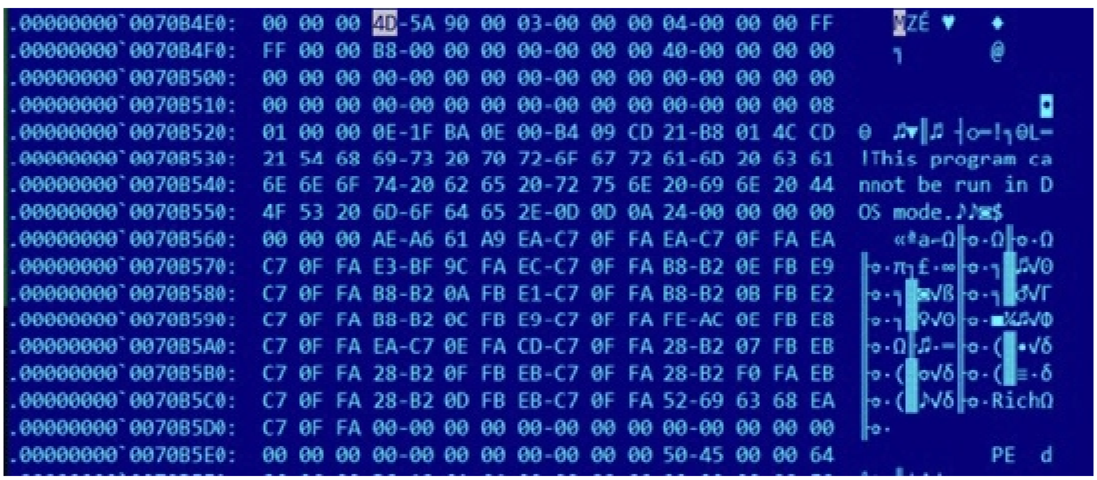


Figure 1. MZ-PE executable inside an ELF file.

Recently, Cado Security Labs made a noteworthy discovery. They came across the first MIPS samples in the wild, determining that including Windows, DLLS is just one part of the strategy to broaden the potential range of targeted devices. This revelation has expanded the scope to potentially affect a wider array of devices.

To this discovery, we add a new set of samples that target another typical IoT architecture: ARM, something that hasn't been reported by other researchers so far. The corresponding example SHA-256 hashes found by Nozomi Networks Labs are the following:

- 4421298c97f245f4e7eafb4f3873b0a95fe22682766c5dfb9c22ccfef8b91ad1
- 8ca16968634b5c7bb0343fff806da827ad00866748fce022da9fb0addc50ee99

A version of this malware that utilizes SSH protocol for propagation was intercepted by our chain of globally distributed honeypots on November 6, 2023, something that hasn't been observed by the researchers when the threat was first analyzed back in July 2023. Our honeypots began recording the initial infection commands in October 2023, and this trend has continued to grow consistently until the current month of January 2024. The majority of these malicious connections originate from locations in China, Hong Kong and Singapore.

On Windows, the compilation timestamps stored as part of the MZ-PE header structure of executables are commonly used to build timelines, even though they can be forged by attackers and on newer versions of Windows don't necessarily represent the timestamps but rather checksums. However, the ELF format of executables commonly found on *nix systems doesn't even have such a field. Still, some conclusions can be made based on the metadata left by the compiler as well as metadata from third-party systems like VirusTotal.

In addition, here we actually do have Windows executables embedded into ELF files, the only thing that is left is to automatically extract all of them. Here is an example SHA-256 hash of the same DLL found inside x86-64, ARM and MIPS samples:

> a29cb8da788a5ebaa7b8f6a6016d7233f81f9f478cf4a52e021a6d6060d09f7e

We have also uploaded it to VirusTotal to share it with the community. If its compilation timestamps (Figure 2) were not forged by attackers, they may indicate that attackers finished developing Windows payloads months before the first widespread distribution in July.

Count of sections	6	Machine	AMD64
Symbol table	00000000[00000000]		Sun May 21 15:16:44 2023
Size of optional header	00F0	Magic optional header	020B
Linker version	14.29	OS version	6.00
Image version	0.00	Subsystem version	6.00
Entry point	00001E60	Size of code	00001A00
Size of init data	00002A00	Size of uninit data	00000000
Size of image	00009000	Size of header	00000400
Base of code	00001000		
Image base	00000001`80000000	Subsystem	GUI
Section alignment	00001000	File alignment	00000200
Stack	00000000`00100000	Heap	00000000`00100000
Stack commit	00000000`00001000	Heap commit	00000000`00001000
Checksum	00000000	Number of dirs	16

Figure 2. The compilation timestamp of the embedded Windows payload states May 2023.

Turning our attention to the technical aspects of this identified threat, we've uncovered several self-defense techniques detailed in the following section.

Tracing the Transformation of Malware Executables

High-level Functionality

Written in Rust, the P2PInfect malware mainly consists of two executable modules: the smaller (~0.5Mb) auxiliary module commonly having a filename “bash” and a larger, main payload. The purpose of the former is to ensure the correct work of the main payload performing all the required validations, updates and revivals where necessary. The main payload is capable of performing various operations, including propagating and delivering other modules with filenames that speak for themselves like miner and winminer. As its name suggests, the malware is capable of performing Peer-to-Peer (P2P) communications without relying on a single Command and Control server (C&C) to propagate attackers’ commands.

As we will see below, some parts of it are less static than others.

Self-Defense Techniques

Samples packed with higher versions of the UPX tool generally can’t be unpacked by lower versions of it, complicating the analysis and detection if automated analysis systems are using outdated versions of UPX. In general, attackers will use the latest version available. Interestingly, the latest variants of this malware tend to utilize a particular version of the UPX packer: 4.0.2. This version is newer than 3.94 and 3.95 which are widely used by other attackers, but well below the most recent version available, which is 4.2.2. The use of version 4.0.2. provides an idea of how the project was developing as this version became available to the public on January 30, 2023, while the next version 4.1.0 not used by attackers was released on August 8, 2023. It could mean that the main part of the development was taking place starting from early in 2023, and after the toolset was set up, the attackers didn’t bother updating it.

The authors' commitment to safeguarding the samples from dynamic analysis and remediation becomes clear when we examine the various security measures employed in P2PInfect. In addition to disabling core dumps through the 'setrlimit' syscall (Figure 3) and the process debugging protection noted by Cado Security, this family of malware employs additional security measures across various layers.

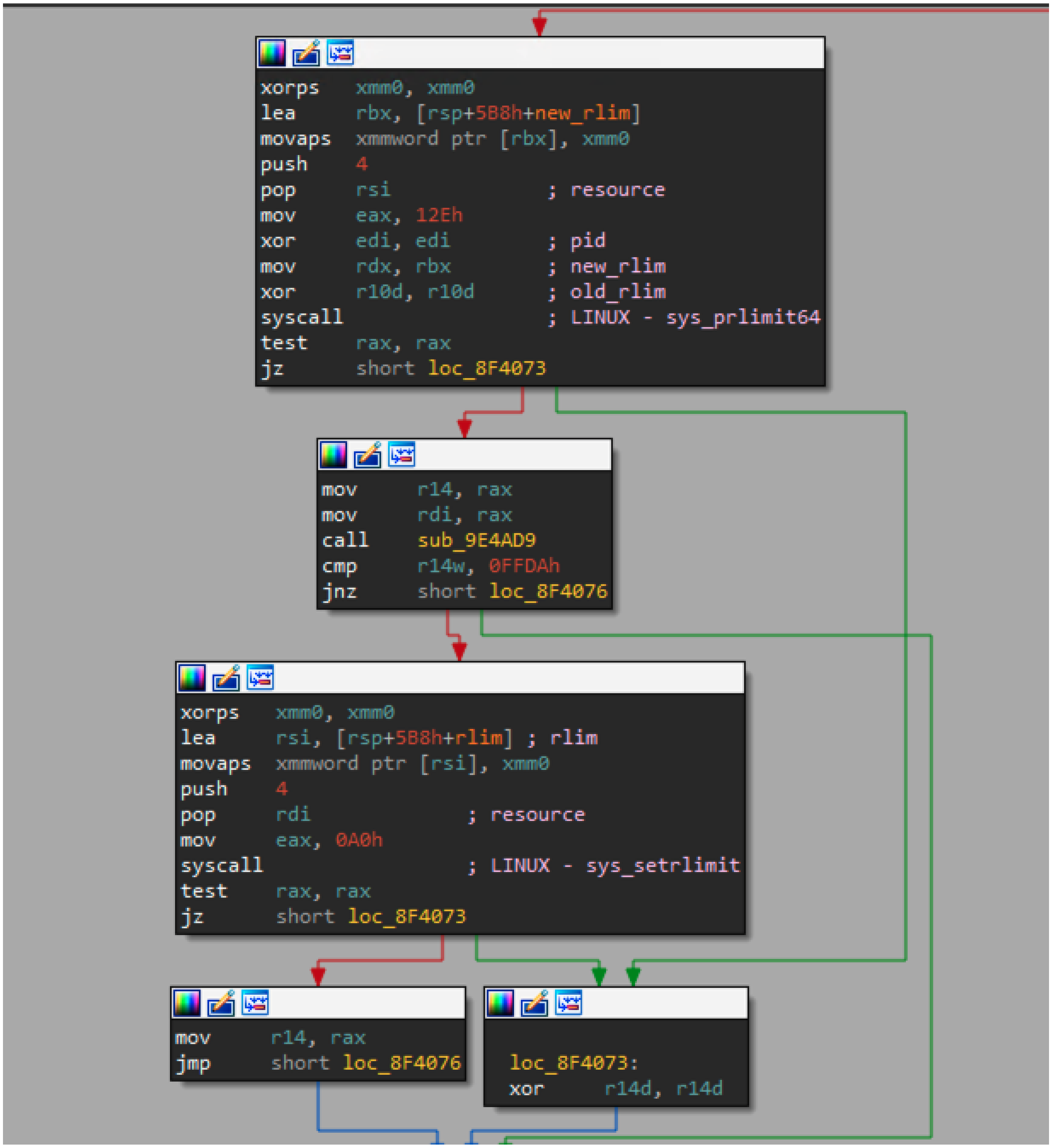


Figure 3. 'setrlimit' syscall to disable core dumps.

As mentioned earlier, the "bash" binary is dropped into the system to ensure the continued "health" of the P2PInfect sample in case a new version is released or its integrity is compromised (e.g., binary modification or the main process is not active). This "bash" binary is not downloaded from the botnet, it's dropped from the sample. To establish effective communication with it, the primary sample listens on the loopback address via a TCP port. This approach can lead to a consistent behavior pattern that analysts can readily identify, and there is also the potential risk that the selected port may already be in use. To mitigate these concerns, the dropped executable is intentionally made non-functional until the malware modifies it during runtime. This measure also prevents the extraction of a functional payload from the primary executable.

To find the memory to be modified inside the “template bash” executable to make it ready to be executed, P2PInfect looks for two markers “thisisport” and “password12345678” to dynamically replace them with two random values.

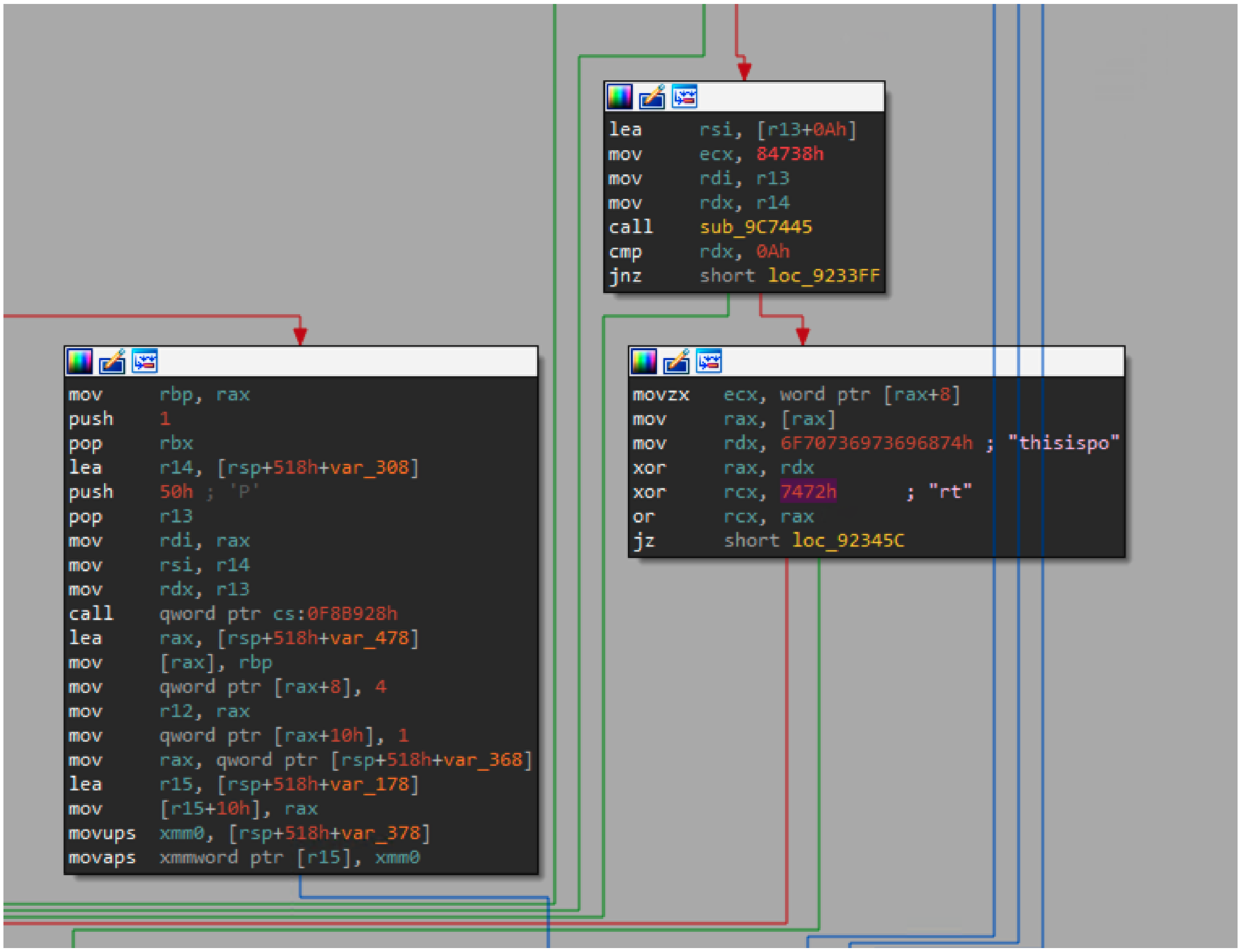


Figure 4. TCP port marker search.

When the offset of the marker is found, a random port between 60100 and 60150 is generated and then an underscore ‘_’ is appended to allow the bash executable to know where the port string ends. This TCP port range is the same as the range of ports used to communicate with other peers within the P2P network.

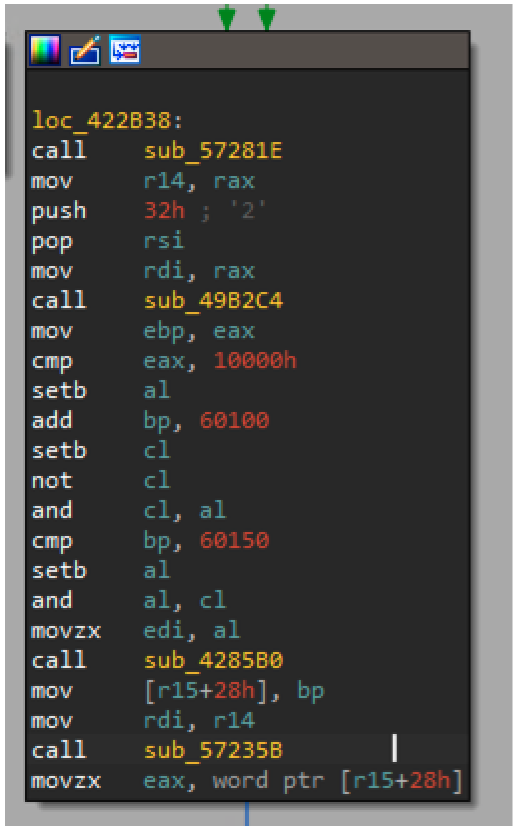


Figure 5. Random TCP port generation between 60100 and 60150.

The password marker is replaced with 16 random characters that will be used to encrypt the communication between both executables. This additional layer of protection provides an additional level of protection to conceal the transmitted content from curious eyes.

Finally, lots of main ELF payloads are distributed inside wrapper ELF files that just print a famous “Hello, World!” message pretending this is its only functionality.

```
; Attributes: bp-based frame

; int __fastcall main(int, char **, char **)
main proc near

var_10= qword ptr -10h
var_4= dword ptr -4

; __unwind {
push    rbp
mov     rbp, rsp
sub     rsp, 10h
mov     [rbp+var_4], edi
mov     [rbp+var_10], rsi
lea     rdi, s          ; "Hello, World!"
call    _puts
mov     eax, 0
leave
retn
; } // starts at 163A
main endp
```

Figure 6. A short main function of the wrapper ELF sample.

An example of it would be a sample d1ad42ab5289447cbd803e186d2115e1b2ea3bf3486dc92c7ef5153f572dfd65 containing the main P2PInfect payload 32365440cbe93909b1dfd4364bcdb0c31953f4d6be97a675eb3984126cc49295 inside (Figure 9).

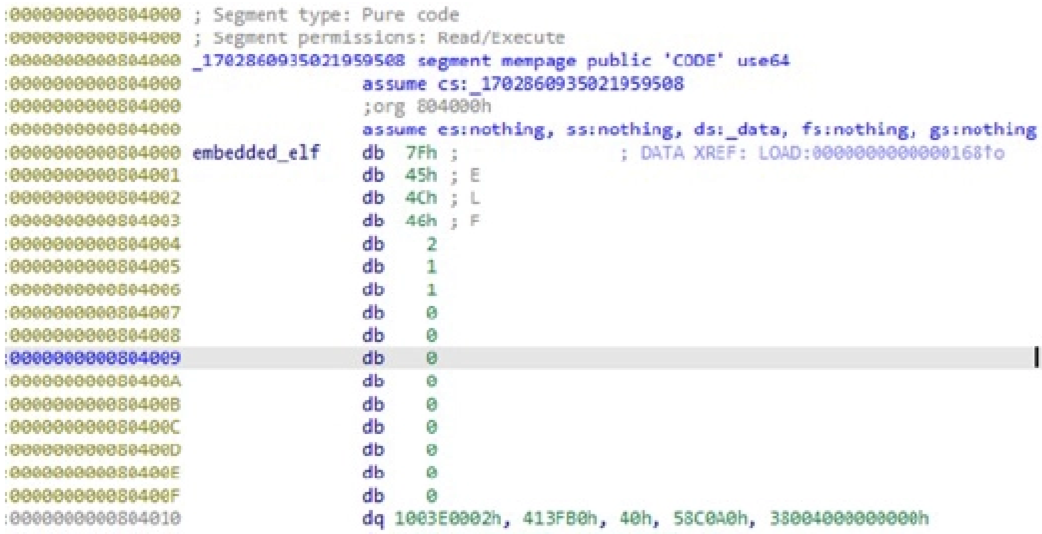


Figure 7. Embedded main P2Pinfect playload.

Once the device is infected it's ready to spread to other devices. To infect new targets, P2PInfect executes a bruteforce attack against the device with the SSH server and, after a successful login, different commands are executed to download and run the malware (Figure 8).

Figure 8. Bash command executed to infect the device after successful login.

An example of such a sample is the following, illustrated in Figure 9:

9ec9d3f720a752b9ab928e1c395c778d3da652442d0f0ae09552efc2f57ee6de

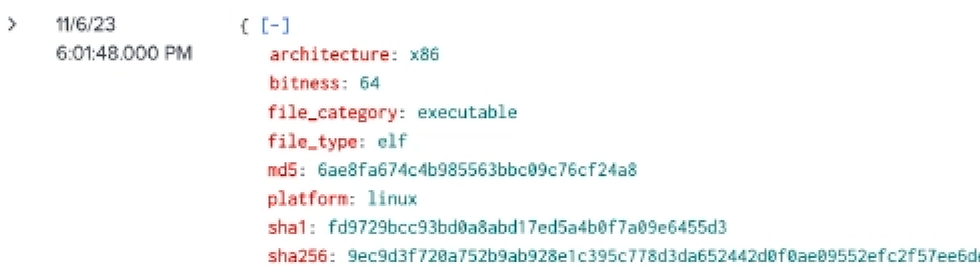


Figure 9. A record from one of the honeypots spotting this sample of P2PInfect.

Conclusion

The P2PInfect malware family continues to expand its reach across various architectures and platforms, representing a significant and evolving threat in the realm of cybersecurity. Its unique characteristics, such as its use of Rust and exploitation of Redis vulnerabilities, highlight the ongoing challenges presented by modern malware.

While researchers and cybersecurity experts are making strides in understanding and combating this worm, P2PInfect serves as a stark reminder of the dynamic nature of cyber threats and the importance of staying ahead in the ever-evolving cybersecurity landscape. As we forge ahead, it is crucial for individuals and organizations alike to remain informed and proactive in their cybersecurity practices to mitigate the risks posed by such sophisticated threats.

Detections

YARA

```
// Created by Nozomi Networks Labs

rule p2pinfect_linux {

meta:

  author = "Nozomi Networks Labs"

  date = "2023-12-13"

  x_threat_name = "P2PInfect"

  name = "P2PInfect - WORM"

  description = "Multiplatform worm that targets Redis servers."

  reference = "https://www.cadosecurity.com/redis-p2pinfect/"

  x_mitre_technique = "T1190, T1059, T1107, T1068, T1071"

  tlp = "green"

  hash = "6d0e4c03cf4731b9b05c3e575a92db9beabccf243263d703c7b332597c8ed591"

strings:

  $str_0 = "\x00need_slow_targets" ascii

  $str_1 = "\x00need_fast_targets" ascii

  $str_2 = "\x00attacker_not_valid_secs" ascii

  $str_3 = "\x00file_servers_online_check_delay_secs" ascii

  $str_4 = "ReloadConfstruct" ascii fullword
```

```
  $str_5 = "Failed to disable core-dumps via rlimit" ascii

  $str_6 = "Failed to check tracer presence via /proc/self/status" ascii

  $str_7 = "AWS_ACCESS_KEY_ID=\"?(.*?)\"?[" ascii

  $big_num = {433ce8229fc5f04b}

condition:
  uint32(0) == 0x464c457f and (3 of ($str_*) or $big_num)
}

// Created by Nozomi Networks Labs

rule p2pinfect_dll {

meta:

   author = "Nozomi Networks Labs"

   date = "2023-12-20"

   x_threat_name = "P2PInfect"

    name = "P2PInfect - WORM"

    description = "Multiplatform worm that targets Redis servers."

    reference = "https://www.cadosecurity.com/redis-p2pinfect/"

    x_mitre_technique = "T1190, T1059, T1107, T1068, T1071"

    tlp = "green"

    hash = "a29cb8da788a5ebaa7b8f6a6016d7233f81f9f478cf4a52e021a6d6060d09f7e"

strings:

     $exp_dll = "\x00exp.dll\x00"

     $rm_onload = "\x00RedisModule_OnLoad\x00"

     $system_exec = "\x00system.exec\x00"

     $redis_module = "\x00RedisModule_"

     $onexit_table = "onexit_table" fullword
```

```
    $lookup_func = "RtlLookupFunctionEntry" fullword

condition:

    uint16(0) == 0x5a4d and all of them and #redis_module > 20

}
```

IPs

117.45.170[.]79

118.122.1[.]20

120.222.158[.]89

124.127.58[.]234

124.88.250[.]55

183.233.174[.]44

193.151.148[.]30

218.56.32[.]85

35.220.253[.]187

36.110.27[.]178

36.7.171[.]21

43.128.15[.]83

43.133.238[.]3

43.134.161[.]34

43.134.225[.]133

43.135.173[.]88

43.155.137[.]204

43.155.142[.]210

43.155.169[.]55

43.155.183[.]210

43.155.85[.]140

43.156.18[.]95

43.159.50[.]150

47.106.228[.]20

47.113.222[.]202

61.157.177[.]227

61.49.105[.]174

62.234.11[.]186

8.134.178[.]4

8.137.14[.]175

8.217.135[.]13

8.218.146[.]78

8.219.52[.]90

URLs

hxxp://103.219.60.221:60146/linux

hxxp://110.191.238.10:60110/linux

hxxp://110.39.11.163:60108/linux

hxxp://118.44.95.82:60104/linux

hxxp://133.242.68.165:60144/linux

hxxp://150.138.83.155:60124/linux

hxxp://154.0.31.161:60106/linux

hxxp://159.65.54.223:60104/linux

hxxp://18.183.15.88:60103/linux

hxxp://193.148.252.194:60119/linux

hxxp://20.191.185.44:60101/linux

hxxp://27.191.237.5:60127/linux

hxxp://43.132.150.184:60134/linux

hxxp://43.155.153.147:60118/linux

hxxp://45.138.174.199:60117/linux

hxxp://47.236.101.172:60119/linux

hxxp://47.245.92.210:60147/linux

hxxp://50.17.152.237:60129/linux

hxxp://61.160.213.239:60147/linux

hxxp://74.208.103.29:60116/linux

hxxp://8.134.144.81:60147/linux

hxxp://8.217.0.228:60140/linux

hxxp://8.218.146.1:60115/linux

hxxp://89.168.78.92:60130/linux

SHA-256 hashes

- 000bf4ef861996b4b11451beec52c79ef4ec68ec56ec38dfd1481b7fbea96911
- 006aaab764e9b249c98cae072872a8b2b1bd8c6a8a44fb9682722cc2cc830ce0
- 0111f06b27c95361a8222cd1e80957fb232c799ba93950dfae65ab1c972f7b3a
- 030789780e91092f1934239ec4d5c2c2ca17d9e3889daac76178107b15b199f6
- 04c3d68ccc274b82bdd59e7fbdc1d314b9ece6dc5ba8e96ed383a159284036d4
- 070bab71b39062c686a40856a2d2198642e4c4d565636ab0cc52d5bdb4395fea
- 07501540d43527f9c0417629162edeeeb66cbb6bb545d20cec86dcad296f621d
- 0c1045eaa5241ae599ce551d2f618e1d3648cd647ecf5b4f5f59fd4da1d1cd03
- 0ecfa32eaa13cc010784d46ba1e68e7d951dc7d95a38da1a2a54e7c22fa7c89b
- 103e2ba056d4b2a074e82902f31b17100aece6d0e0e9dfb0b6e2d102fe6f9dcb
- 10dc62048cfd3392178fc351f89271a98bdc3df750a4beb60a0f9763ef6cc70f
- 150df742a24ef6b146d468d67ae453b73a05e0275c995afcf05654f8dca3b9df
- 15542fe695b3beabe329934b4be5bafcc6a7bed70a4b9b46a2777ae07032b569
- 1c2363532c8b83243bb1933d7213ceb80e9a095280a26175421a70ef3b7834e9

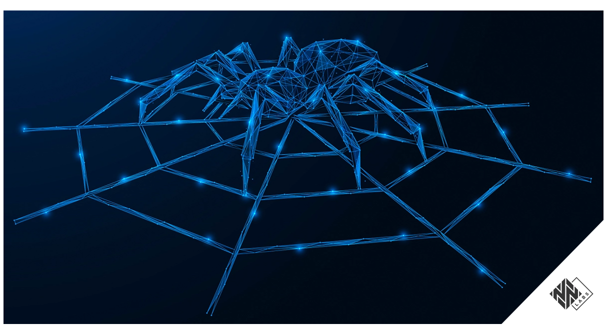
- 1e2686c1a674630311fdab9b74df54605309076b6d2c3acb4dbc0e7c0080bfa4
- 1e963c97bfa850522984b09e80c1be7bcd372ad527bcfdd0787922d27fdb253d
- 22fe687809d73835d943deb396a84152615cec56d483072de68c0e734446b8b1
- 23fd0e7b1560cdd41df020456fc7ca8ada49c80fdcaa73880779f18b5dc6d954
- 24e9524c3c2f23f9cfd34919881f06c6312a0d3c96459bb995b325aa24caeb69
- 25620e3539baa874e66cd3e87a225d88378abacdf4c2b962acdb3e86ceb936eb
- 286f87b793f6f86d24d500d22cf103285c1e3aad5193ac0027d5c826ff9952ef
- 2ce525ef39f57a3e232eb7a08bec5afde05ebbe5f1794c6260b812ddffebb7e8
- 2e895366e84312413a157d5cd1e01440d54d0108fcb95ec4c455cdc019b31200
- 3034fee705440af41d7ed3f25e5c5af4d11d2ca066dbd797032bd63f5f433626
- 317115f0f0e11c6693cc010f260138d3364cefc41662d5fce825517ae3675080
- 32365440cbe93909b1dfd4364bcdb0c31953f4d6be97a675eb3984126cc49295
- 3814554e9b418c25ed1548d3282d56a96e1021608cb5133e6875b104c5e7113e
- 39c6e11c8a50d28536423cd34153c323c34d0166805c4157490c834fe3b75ae3
- 3a43116d507d58f3c9717f2cb0a3d06d0c5a7dc29f601e9c2b976ee6d9c8713f
- 3d8712914d2f1525ef89cc15002c63e3ec533db38c87c272ff704218b68c737c
- 3e7233085678ca016ce6228cb50d735e4175533843be0b36012c853083938906
- 40cc94f348dca65b36ce74b6be9f225b25491f04f4d764ab7b0214b59d7407de
- 412f87a2a4245c371e428607dec3cb6c22c15d387dc5db69cd47146e15090216
- 42fb6c13c0ece7133e79dc055b0acb96007a50255ceba6861cc31a987b3e5987
- 43ca01e3123ebdfc546d4edfc7eb06d020757ad9b98b2fef3f49e4632362a672
- 4421298c97f245f4e7eafb4f3873b0a95fe22682766c5dfb9c22ccfef8b91ad1
- 44e6811fa8a44b2f5f1b60acbbc3f116e90f2d35ff418551c034fc9d9b96bb4c
- 470bc66a0e13ce093e1aef9843b7903e7c01b252b9bd4b11551d25437c6210e1
- 47714c45c9172f914e783b49c143daa433356a828526f38071db11743cc96ef7
- 491b859987de3835f77b21d737df5c8c201ccc1d25f48e07803cc73d9e9e336e
- 4b259143cf710d62b5d8767fcc4c5434fd8684334519dc4045c430ee73a9398b
- 4cb3c2f676c158d1d84ed1c2b4600157bb848e29e13fde615ab4675e433d4905
- 4e5bb1db1e306c9e7324195d083abcdb55b4ed1689a498e35ad7ef46650c45e4
- 4f4b6391f66b388c0e0d5b0c2ba775e9402434af115dfa9455423c72efc0f27f
- 50f76c9dea3255b9e4d33c02df0ae8d32cf8909bae6103770afdb44bb2f78f64
- 532dd10bde4bfa9a5032231449ec15d94ed8cbf128b24cacdcef3370039e8ded
- 537a0f4c1a490f769780b9a378414e06c6c4bc1ec1e70109adafadb43fe5648f
- 54ecbd099384f39fd1055133b7a6ef8f64a6c1a76b1bef712da4f69c9bfa3f77
- 55385b542cbe730199159ef403d7292efd8f5d439da4a3f8fff84e94fb0b6abe
- 59c49add8a988087c2cc8dc6c783465990c94ce82608b1eee1dadd30bf2604aa
- 5cf9b01fb73e16f1b8cd2bd4846a829d9d4d9f3270d8644065ce9b67e32b85c9
- 68f7104eb6591175866b79784b14d81f587508574537a376a7c92f3563241973
- 69d83816f675f70f3e703a260504c50df341acd46f8470bb4357a9fde8a7d3d2
- 6a22025d0309d9cf0923fe633b15d6a73ff7b66000e8797a637224269dcddde2
- 6ae60971d812d7dad63af579faeee3fd2aa03bc0528765b3477ccef6cc92010f
- 6d0e4c03cf4731b9b05c3e575a92db9beabccf243263d703c7b332597c8ed591

- 6fa48f060e313c108cc74da76a6b4ab3073a4f352c7ff4a4650eec7fd9042b1c
- 71f7b35c831207eb2f479d3c3eaf37cc54717cf55e6be00b85a6a8376d61839f
- 759d79f7f5ac7fd8bab551bf78c6e28c169c72bdb463642e3732b5dcfa0dd753
- 7819a5217c6d91e828128afdae760147714604c74c5c315ee51dc65d4aaecf72
- 796e84ed257f40cd8b77968839051ea815a027a9e5210298cc491c2ebfa9f4c3
- 79f99b9df8626b3bd28ccea20e111081cc886b4d41a0eafb53412231a0c44da7
- 7ca878dc7c8a9b9e9b54faf9b1b354e657ad890ba2cf67b26f3d66d1a7757479
- 7d22d02813997c6403159593589efb00d172e22143bbe436ea2df06a130a3445
- 83216df7dcb8760e451c4bb8db5a520a64c5bae589d0712203c3ab03086eeae0
- 8594830664b030dd772449279a1738ebbbb419798b0ec5dc547518b954d38877
- 8ca16968634b5c7bb0343fff806da827ad00866748fce022da9fb0addc50ee99
- 8df14ce418fcede6be48ddaa16517c6f5249888f654680a85c43850914f0a959
- 9659f5e59c17163b48431bd98b34d5185ae534d27e0bb5f42ff2edc9335931b3
- 9883db5f74aa42af6a8ae387e9f54c5f1f2c6a7dd032b58c88ab742670ce00f9
- 99afa8c5b0a61669fb9c960b97568aca175a35fadd60129ee441dd55ee27c20f
- 9b35e4bb82b36c81a064b1178f38926f82b767ccd07b0838be536876b56bc852
- 9dd1b238c27ec151bf76da9a825875ff190480813876ad68cebf6a1cddd552a6
- 9ec9d3f720a752b9ab928e1c395c778d3da652442d0f0ae09552efc2f57ee6de
- 9f2686ea42fdbb538ed55efa73896996005b5649cedb91be8907e3143be43d8a
- 9f2efc2cb95ca881b534c55e54b57ef1992fddd3078035e15a8ba3f9a41a2f33
- a29cb8da788a5ebaa7b8f6a6016d7233f81f9f478cf4a52e021a6d6060d09f7e
- a6cc37de6c8e5b83b9f70b1332ec24e3913c97d511ac21eae2141053ba926508
- aa9d34bbd2743dd5299d8cc1e7f24a54955d7d54cc3bfd45424e32fbcc540d62
- ab3163775d27efa38a60b1f4fdcedcf0d3c493ccd72a06ee3660d97a7725328f
- aba5eac208c82899a9b0abe4215e42ddce8eefcd5f4186b47fb1188e306972e7
- ad32d8b2c7bc9eec9fd183a00ebb78cbf886760347ff5e826a071fba7d5f5219
- b02b49b4d4c7e6fe3abae02ccfc2524347700f40e1856f6d0ec40dd247fd24f4
- b0a1cc6968bf9bf7f8a7f3f15a5ceace28cc0899604f2122a1e16e8c225f224a
- b445789bea083a1a8264ab6d8478d38a78a4dd5a32cefcc3b9367097cabfd807
- b46a466bef3078ee336a1fadcf62a439656ab9f68f6b3dd84ba341c9442bb96f
- b697d6ee591e4409548db6dfd0c4b32e1640bea01e2001d9ebd413e8f6d7dfbc
- b6d2a14ac7e1158c1d4b7152b7fcdb1409be7ea9866363682256f15514f037b3
- b9013d8429073c3abebc45d8bfb3881243e2bb6115860c947837a0563f6a6bf7
- bb4ef9da07041f8665d76b90dce3e7a26b3695fb0e68b94b4077a544133b8a7b
- bb9f1507f2fb1eeb31a79ce14728c5788a1c847b51281428a965e36d7dc24224
- bbfeeb48447c5861dde7cc5860920d9381492ad7a7d03242dd36f863cf50c26f
- bcb193953a4d3ba76ab0b516e2394a41443d53064ad06cd3b9b40bb8a3a98dfe
- bfd3c1a4655079a8c8a15067964b0b5b3a133c8c8e778863b70fbb4769a837b6
- c1ecc8c4a6a00aa00e59b23bf9cf08983aa8c4313ae62be39e91272edcdff6b4
- c2e6a50d7354c20d020710f9a07ea5c4ccf21a9030743bec69fe2f8538884ab0
- c3ef4d4441362483f9cd3cc22524f85f2c022178bdf8070f05b03a5f6177fbb2
- c61645672143c1369353642df218181c1843821600c095671b56aff73c082b3d

- cda9251a687b9629587a852deddf1d3d8d830d474a6787fa0ef123f557084a7c
- cfb461e10de99a2c08fe985ecaf73ea357a2dc1c47ca8c3749bc613db7a3fc02
- d03f1bd2a8cf7067db9a92c91cc7620dece14473ad27747488cb4491be8f6f18
- d14f22cb372690b2f6e77260a3717fdbd2a80346496ba1279704a4d2004df3ed
- d1ad42ab5289447cbd803e186d2115e1b2ea3bf3486dc92c7ef5153f572dfd65
- d3d5b9239d35321e3ca6b75fcda62e4d81439be5804e7bd8c78c0626aabbd328
- d5e17064b884584dcd075830bc50f596e4e4d3a4264342fbd5cfa8ec036d734b
- df98a431664f1a1249719fd415a22ca51d3858b4b3f7f2eeb21ec688d20084b0
- eb99f302346405ca63cac6cce8b43d86af899ef5f1bdcfaf1f55454a92c49cdd
- efb9845889bfb16fc99cd0a44593c2482c64e0089af8cb950ee746b3e236b2da
- efeb88732ad4538aafe369d6a604f6948e302a819271b4a7f817525801a2ed18
- f04e97fb3a053f5c6a60499e99ba58ef840b1b40882119d6cea64fa052697d82
- f20703057d771f5470095e063255f9ffa9fa5caafc44995b6590d6485ad647b8
- f54586112d54a8fd9b3f94ee199d2f05da2877ec72db7496fc464620048c7bec
- f6bd75914a01074be877c4a205745f170ed5d13c0d85acbd2e74c60a982f0117
- f721c8acc4436de8d8e5bfc76e654d6e982d6bda5a969c11d53c79785b30d79a
- fa56376c093ac3a5d5b349e6c8f02b4dc5d19b11d57103039e2b9bd861b07955
- faf035ddbfc2304f7eafbc62d60f25b8460083efb21ae3642ca3e4eea0ab2fe0

No items found.

No items found.



How to Build an IoT Honeypot

We detail five steps for setting up and using an IoT Honeypot in the real world using the Winbox protocol.

View Blog

ABOUT THE AUTHOR

Nozomi Networks Labs

Nozomi Networks Labs is dedicated to reducing cyber risk for the world's industrial and critical infrastructure organizations. Through our cybersecurity research and collaboration with industry and institutions, we're helping defend the operational systems that support everyday life.

Get Updates