Apr 01 2024

NETMANAGEIT

Intelligence Report

Malware Spotlight: Linodas aka DinodasRAT for Linux

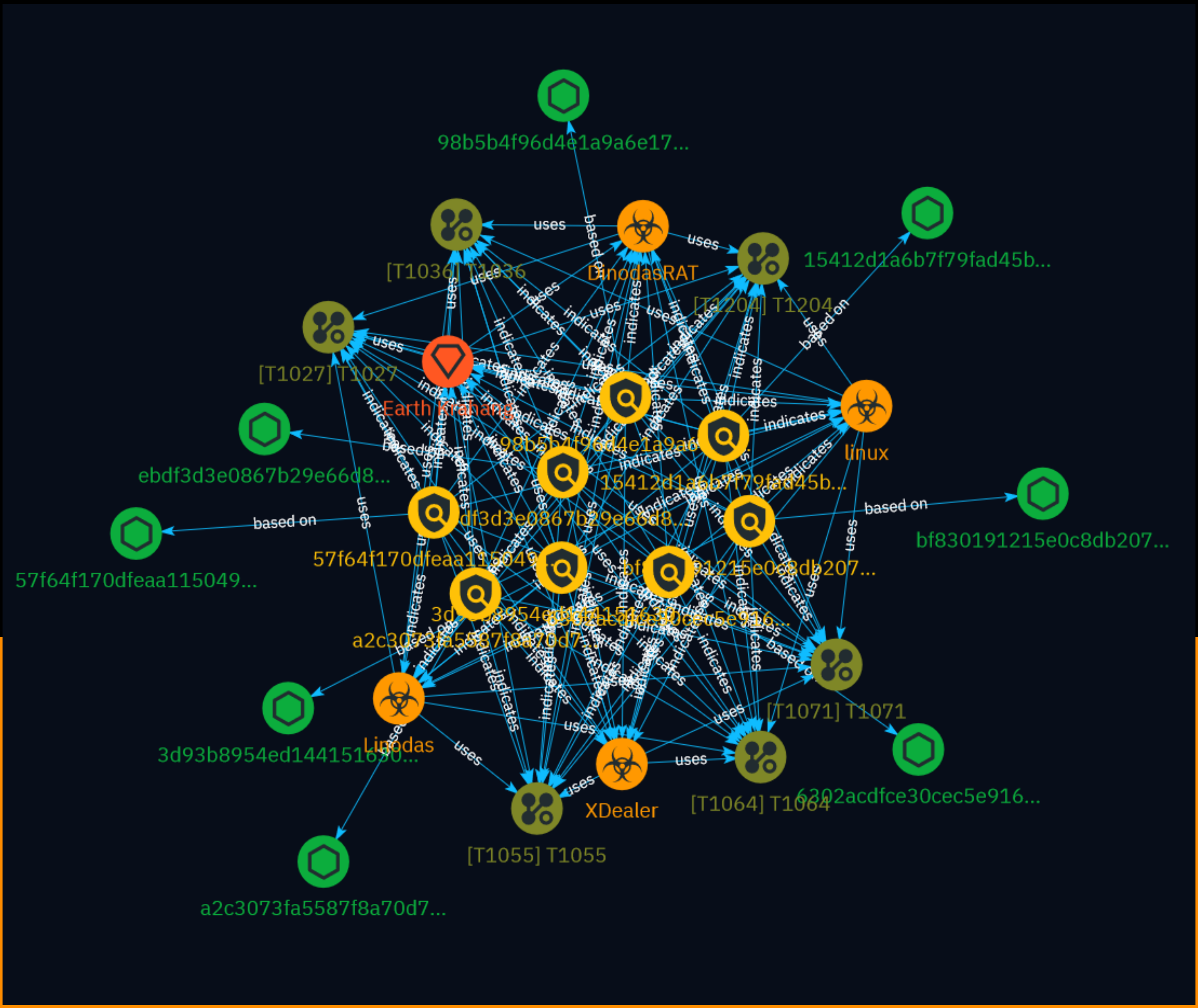

Table of contents

Overview

Entities

Observables

External References

Overview

Description

A Chinese-nexus cyber espionage threat actor is focusing on Southeast Asia, Africa, and South America, aligning with insights on threat actor Earth Krahang. The actor uses a cross-platform backdoor DinodasRAT, aka XDealer, linking it to Chinese actor LuoYu. While the Windows version is analyzed, the Linux version is not. Here we analyze Linux version 11 of DinodasRAT, called Linodas. It adds Linux-specific capabilities like reverse shells and logs monitoring. The latest version hides malware via a module proxying/modifying system binaries. Linodas shows continued targeting of Linux servers as pivot points in networks.

Confidence

This value represents the confidence in the correctness of the data contained within this report.

100 / 100

Content

N/A

Indicator

Name
3d93b8954ed1441516302681674f4989bd0f20232ac2b211f4b601af0fcfc13b
Pattern Type
stix
Pattern
[file:hashes.'SHA-256' = '3d93b8954ed1441516302681674f4989bd0f20232ac2b211f4b601af0fcfc13b']
Name
ebdf3d3e0867b29e66d8b7570be4e6619c64fae7e1fbd052be387f736c980c8e
Pattern Type
stix
Pattern
[file:hashes.'SHA-256' = 'ebdf3d3e0867b29e66d8b7570be4e6619c64fae7e1fbd052be387f736c980c8e']
Name

bf830191215e0c8db207ea320d8e795990cf6b3e6698932e6e0c9c0588fc9eff

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'bf830191215e0c8db207ea320d8e795990cf6b3e6698932e6e0c9c0588fc9eff']

Name

a2c3073fa5587f8a70d7def7fd8355e1f6d20eb906c3cd4df8c744826cb81d91

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'a2c3073fa5587f8a70d7def7fd8355e1f6d20eb906c3cd4df8c744826cb81d91']

Name

98b5b4f96d4e1a9a6e170a4b2740ce1a1dfc411ada238e42a5954e66559a5541

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '98b5b4f96d4e1a9a6e170a4b2740ce1a1dfc411ada238e42a5954e66559a5541']

Name

6302acdfce30cec5e9167ff7905800a6220c7dda495c0aae1f4594c7263a29b2

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '6302acdfce30cec5e9167ff7905800a6220c7dda495c0aae1f4594c7263a29b2']

Name

57f64f170dfeaa1150493ed3f63ea6f1df3ca71ad1722e12ac0f77744fb1a829

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = '57f64f170dfeaa1150493ed3f63ea6f1df3ca71ad1722e12ac0f77744fb1a829']

Name

15412d1a6b7f79fad45bcd32cf82f9d651d9ccca082f98a0cca3ad5335284e45

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' =
'15412d1a6b7f79fad45bcd32cf82f9d651d9ccca082f98a0cca3ad5335284e45']

Malware

Name
Linodas
Name
DinodasRAT
Name
XDealer
Name
linux

Intrusion-Set

Name
Earth Krahang

Attack-Pattern

Name
T1064
ID
T1064
Description
This technique has been deprecated. Please use [Command and Scripting Interpreter](https://attack.mitre.org/techniques/T1059) where appropriate. Adversaries may use scripts to aid in operations and perform multiple actions that would otherwise be manual. Scripting is useful for speeding up operational tasks and reducing the time required to gain access to critical resources. Some scripting languages may be used to bypass process monitoring mechanisms by directly interacting with the operating system at an API level instead of calling other programs. Common scripting languages for Windows include VBScript and [PowerShell](https://attack.mitre.org/techniques/T1086) but could also be in the form of command-line batch scripts. Scripts can be embedded inside Office documents as macros that can be set to execute when files used in [Spearphishing Attachment](https://attack.mitre.org/techniques/T1193) and other types of spearphishing are opened. Malicious embedded macros are an alternative means of execution than software exploitation through [Exploitation for Client Execution](https://attack.mitre.org/techniques/T1203), where adversaries will rely on macros being allowed or that the user will accept to activate them. Many popular offensive frameworks exist which use forms of scripting for security testers and adversaries alike. Metasploit (Citation: Metasploit_Ref), Veil (Citation: Veil_Ref), and PowerSploit (Citation: Powersploit) are three examples that are popular among penetration testers for exploit and post-compromise operations and include many features for evading defenses. Some adversaries are known to use PowerShell. (Citation: Alperovitch 2014)

Name

T1027

ID

T1027

Description

Adversaries may attempt to make an executable or file difficult to discover or analyze by encrypting, encoding, or otherwise obfuscating its contents on the system or in transit. This is common behavior that can be used across different platforms and the network to evade defenses. Payloads may be compressed, archived, or encrypted in order to avoid detection. These payloads may be used during Initial Access or later to mitigate detection. Sometimes a user's action may be required to open and [Deobfuscate/Decode Files or Information](https://attack.mitre.org/techniques/T1140) for [User Execution](https://attack.mitre.org/techniques/T1204). The user may also be required to input a password to open a password protected compressed/encrypted file that was provided by the adversary. (Citation: Volexity PowerDuke November 2016) Adversaries may also use compressed or archived scripts, such as JavaScript. Portions of files can also be encoded to hide the plain-text strings that would otherwise help defenders with discovery. (Citation: Linux/Cdorked.A We Live Security Analysis) Payloads may also be split into separate, seemingly benign files that only reveal malicious functionality when reassembled. (Citation: Carbon Black Obfuscation Sept 2016) Adversaries may also abuse [Command Obfuscation](https://attack.mitre.org/techniques/T1027/010) to obscure commands executed from payloads or directly via [Command and Scripting Interpreter](https://attack.mitre.org/techniques/T1059). Environment variables, aliases, characters, and other platform/language specific semantics can be used to evade signature based detections and application control mechanisms. (Citation: FireEye Obfuscation June 2017) (Citation: FireEye Revoke-Obfuscation July 2017)(Citation: PaloAlto EncodedCommand March 2017)

Name

T1204

ID

T1204

Description

An adversary may rely upon specific actions by a user in order to gain execution. Users may be subjected to social engineering to get them to execute malicious code by, for example, opening a malicious document file or link. These user actions will typically be observed as follow-on behavior from forms of [Phishing](https://attack.mitre.org/techniques/T1566). While [User Execution](https://attack.mitre.org/techniques/T1204) frequently occurs shortly after Initial Access it may occur at other phases of an intrusion, such as when an adversary places a file in a shared directory or on a user's desktop hoping that a user will click on it. This activity may also be seen shortly after [Internal Spearphishing](https://attack.mitre.org/techniques/T1534). Adversaries may also deceive users into performing actions such as enabling [Remote Access Software](https://attack.mitre.org/techniques/T1219), allowing direct control of the system to the adversary, or downloading and executing malware for [User Execution](https://attack.mitre.org/techniques/T1204). For example, tech support scams can be facilitated through [Phishing](https://attack.mitre.org/techniques/T1566), vishing, or various forms of user interaction. Adversaries can use a combination of these methods, such as spoofing and promoting toll-free numbers or call centers that are used to direct victims to malicious websites, to deliver and execute payloads containing malware or [Remote Access Software](https://attack.mitre.org/techniques/T1219).(Citation: Telephone Attack Delivery)

Name

T1055

ID

T1055

Description

Adversaries may inject code into processes in order to evade process-based defenses as well as possibly elevate privileges. Process injection is a method of executing arbitrary code in the address space of a separate live process. Running code in the context of another process may allow access to the process's memory, system/network resources, and possibly elevated privileges. Execution via process injection may also evade detection from security products since the execution is masked under a legitimate process. There are many different ways to inject code into a process, many of which abuse legitimate functionalities. These implementations exist for every major OS but are typically platform specific. More sophisticated samples may perform multiple process injections to segment modules and further evade detection, utilizing named pipes or other inter-process communication (IPC) mechanisms as a communication channel.

Name
T1036

ID
T1036

Description
Adversaries may attempt to manipulate features of their artifacts to make them appear legitimate or benign to users and/or security tools. Masquerading occurs when the name or location of an object, legitimate or malicious, is manipulated or abused for the sake of evading defenses and observation. This may include manipulating file metadata, tricking users into misidentifying the file type, and giving legitimate task or service names. Renaming abusable system utilities to evade security monitoring is also a form of [Masquerading](https://attack.mitre.org/techniques/T1036).(Citation: LOLBAS Main Site) Masquerading may also include the use of [Proxy](https://attack.mitre.org/techniques/T1090) or VPNs to disguise IP addresses, which can allow adversaries to blend in with normal network traffic and bypass conditional access policies or anti-abuse protections.

Name
T1071

ID
T1071

Description

Adversaries may communicate using OSI application layer protocols to avoid detection/ network filtering by blending in with existing traffic. Commands to the remote system, and often the results of those commands, will be embedded within the protocol traffic between the client and server. Adversaries may utilize many different protocols, including those used for web browsing, transferring files, electronic mail, or DNS. For connections that occur internally within an enclave (such as those between a proxy or pivot node and other nodes), commonly used protocols are SMB, SSH, or RDP.

StixFile

Value
3d93b8954ed1441516302681674f4989bd0f20232ac2b211f4b601af0fcfc13b
ebdf3d3e0867b29e66d8b7570be4e6619c64fae7e1fbd052be387f736c980c8e
bf830191215e0c8db207ea320d8e795990cf6b3e6698932e6e0c9c0588fc9eff
a2c3073fa5587f8a70d7def7fd8355e1f6d20eb906c3cd4df8c744826cb81d91
98b5b4f96d4e1a9a6e170a4b2740ce1a1dfc411ada238e42a5954e66559a5541
6302acdfce30cec5e9167ff7905800a6220c7dda495c0aae1f4594c7263a29b2
57f64f170dfeaa1150493ed3f63ea6f1df3ca71ad1722e12ac0f77744fb1a829
15412d1a6b7f79fad45bcd32cf82f9d651d9ccca082f98a0cca3ad5335284e45

External References

- https://research.checkpoint.com/2024/29676/
- https://otx.alienvault.com/pulse/660a811d68237351c3984e43