Aug 24 2023

NETMANAGEIT

Intelligence Report

Why LaZagne Makes D-Bus API Vigilance Crucial

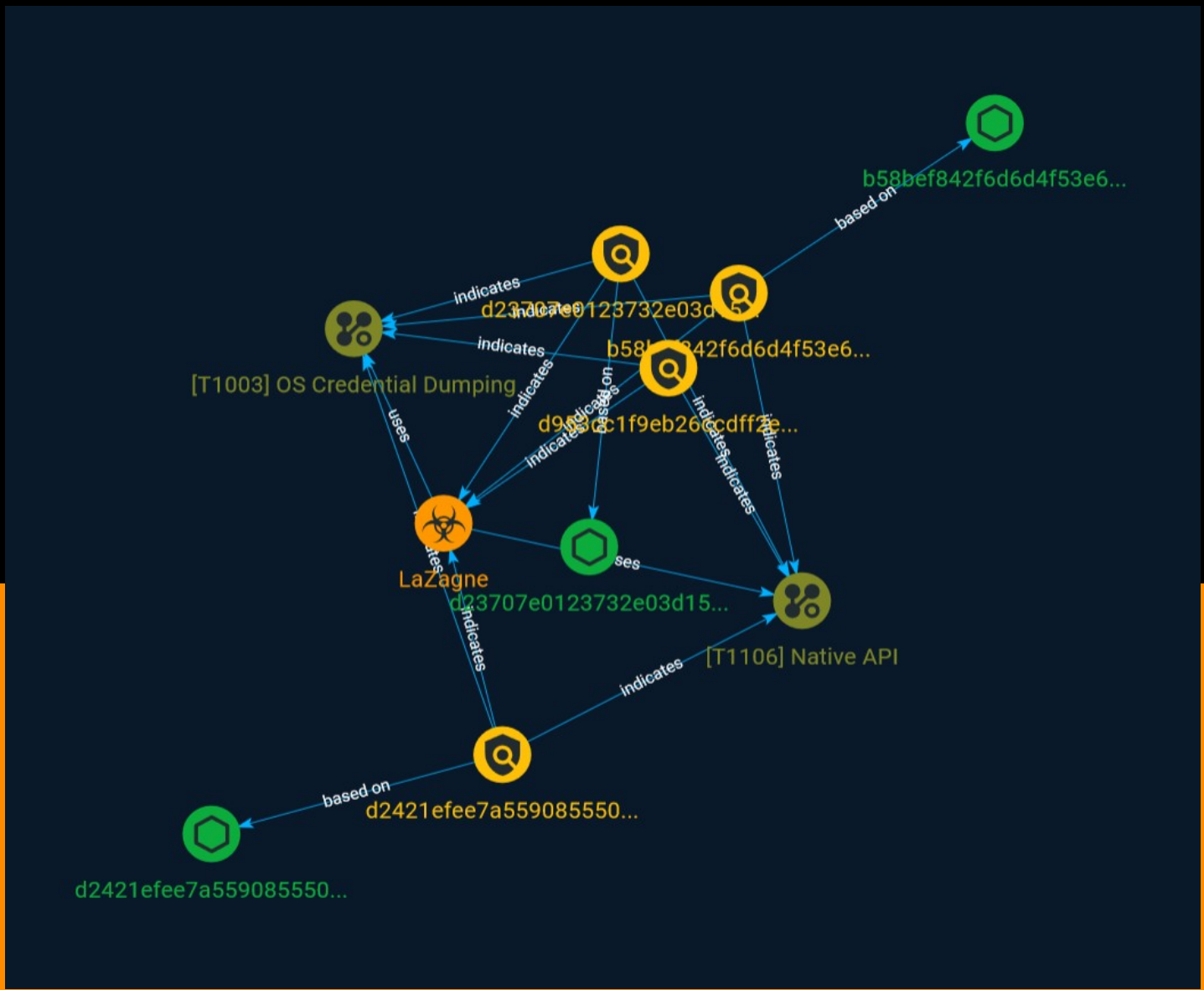

Table of contents

Overview

Entities

Observables

External References

Overview

Description

A popular Linux chat software like Pidgin can be used to extract usernames, passwords and other sensitive information from its D-Bus APIs, according to a Palo Alto Networks security tool.

Confidence

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

15 / 100

Indicator

Name

b58bef842f6d6d4f53e6821f9ac1b63780267cc81006b649b56c263efeab1306

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'b58bef842f6d6d4f53e6821f9ac1b63780267cc81006b649b56c263efeab1306']

Name

d953cc1f9eb26ccdff2effbfc9ed5546ae69682f

Description

the lazagne hacktool.

Pattern Type

yara

Pattern

rule elf_hacktool_lazagne { meta: author = "Siddharth Sharma - PaloAltoNetworks" description = "the lazagne hacktool." strings: $str1="lazagne" ascii wide nocase $str2="softwares.chats.pidgin" ascii wide nocase $str3="softwares.wallet.gnome" ascii wide nocase $str4="softwares.sysadmin.shadow" ascii wide nocase $str5="libdbus" ascii wide nocase condition: uint32(0) == 0x464c457f and all of them }

Name

d23707e0123732e03d156a0fd474a1384e1b3deee3235df9e96ff5d21a4d440c

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'd23707e0123732e03d156a0fd474a1384e1b3deee3235df9e96ff5d21a4d440c']

Name

d2421efee7a559085550b5575e2301a7c2ed9541b9e861a23e57361c0cdbdbdb

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'd2421efee7a559085550b5575e2301a7c2ed9541b9e861a23e57361c0cdbdbdb']

Malware

Name
LaZagne

Attack-Pattern

Name
OS Credential Dumping
ID
T1003
Description
Adversaries may attempt to dump credentials to obtain account login and credential material, normally in the form of a hash or a clear text password, from the operating system and software. Credentials can then be used to perform [Lateral Movement](https://attack.mitre.org/tactics/TA0008) and access restricted information. Several of the tools mentioned in associated sub-techniques may be used by both adversaries and professional security testers. Additional custom tools likely exist as well.
Name
Native API
ID
T1106
Description
Adversaries may interact with the native OS application programming interface (API) to execute behaviors. Native APIs provide a controlled means of calling low-level OS services

within the kernel, such as those involving hardware/devices, memory, and processes. (Citation: NT API Windows)(Citation: Linux Kernel API) These native APIs are leveraged by the OS during system boot (when other system components are not yet initialized) as well as carrying out tasks and requests during routine operations. Native API functions (such as `NtCreateProcess`) may be directed invoked via system calls / syscalls, but these features are also often exposed to user-mode applications via interfaces and libraries.(Citation: OutFlank System Calls)(Citation: CyberBit System Calls)(Citation: MDSec System Calls) For example, functions such as the Windows API `CreateProcess()` or GNU `fork()` will allow programs and scripts to start other processes.(Citation: Microsoft CreateProcess)(Citation: GNU Fork) This may allow API callers to execute a binary, run a CLI command, load modules, etc. as thousands of similar API functions exist for various system operations. (Citation: Microsoft Win32)(Citation: LIBC)(Citation: GLIBC) Higher level software frameworks, such as Microsoft .NET and macOS Cocoa, are also available to interact with native APIs. These frameworks typically provide language wrappers/abstractions to API functionalities and are designed for ease-of-use/portability of code.(Citation: Microsoft NET)(Citation: Apple Core Services)(Citation: MACOS Cocoa)(Citation: macOS Foundation) Adversaries may abuse these OS API functions as a means of executing behaviors. Similar to [Command and Scripting Interpreter](https://attack.mitre.org/techniques/T1059), the native API and its hierarchy of interfaces provide mechanisms to interact with and utilize various components of a victimized system. While invoking API functions, adversaries may also attempt to bypass defensive tools (ex: unhooking monitored functions via [Disable or Modify Tools](https://attack.mitre.org/techniques/T1562/001)).

StixFile

Value
d2421efee7a559085550b5575e2301a7c2ed9541b9e861a23e57361c0cdbdbdb
b58bef842f6d6d4f53e6821f9ac1b63780267cc81006b649b56c263efeab1306
d23707e0123732e03d156a0fd474a1384e1b3deee3235df9e96ff5d21a4d440c

External References

- https://otx.alienvault.com/pulse/64e775a5cc20b64c5a0c6726
- https://unit42.paloaltonetworks.com/lazagne-leverages-d-bus/