Mar 25 2024

NETMANAGEIT

Intelligence Report

APT29 Uses WINELOADER to Target German Political Parties

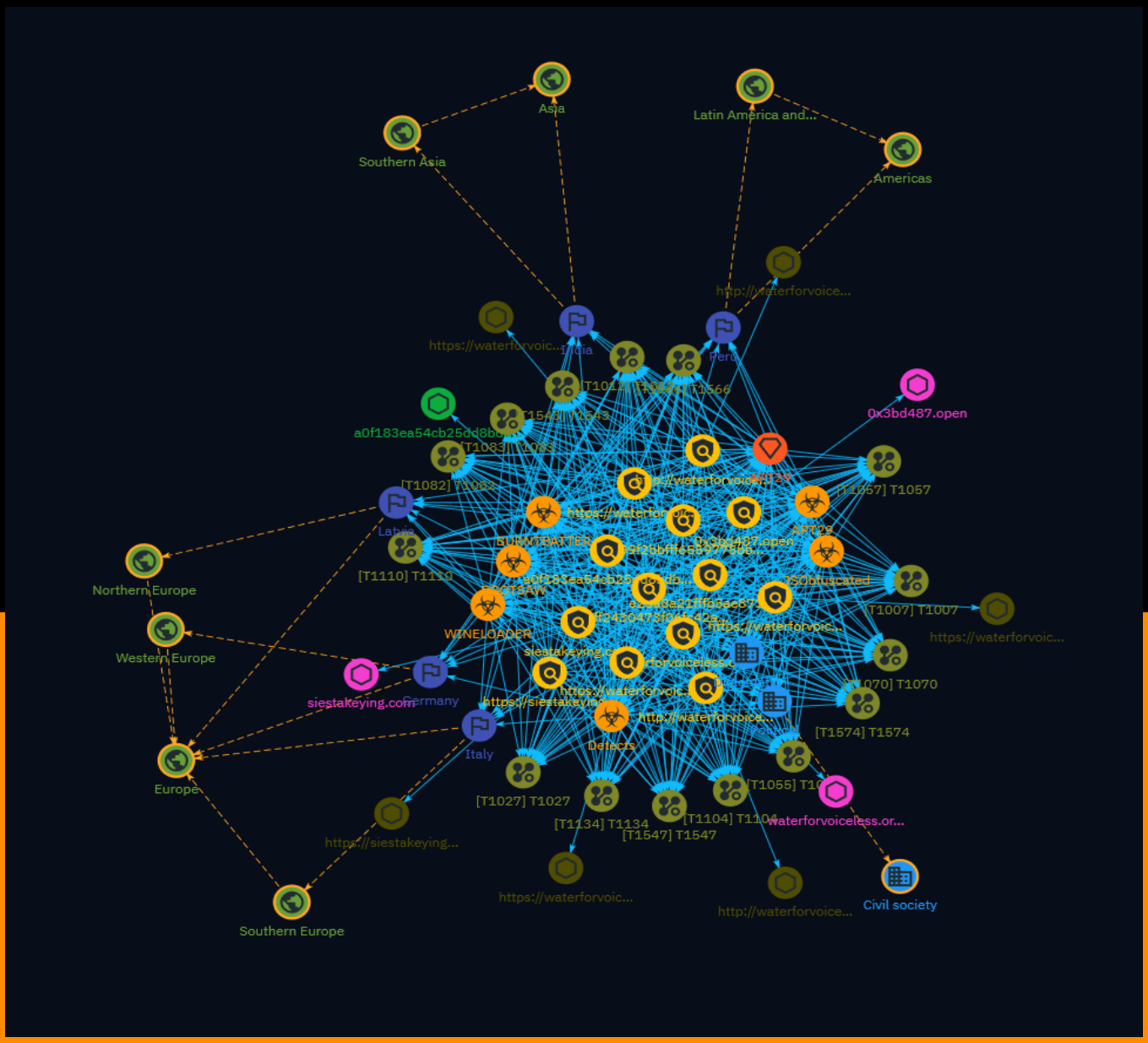

Table of contents

Overview

Entities

Observables

External References

Overview

Description

In late February, APT29 used a new backdoor variant publicly tracked as WINELOADER to target German political parties with a CDU-themed lure. This is the first time the APT29 cluster has been observed targeting political parties, indicating a possible area of emerging operational focus beyond the typical targeting of diplomatic missions. This activity presents a broad threat to European and other Western political parties from across the political spectrum.

Confidence

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

100 / 100

Content

N/A

Indicator

Name

e25a8a21fffb5ae871022f4342db2a0e6561191e

Description

Detects rc4 decryption logic in WINELOADER samples

Pattern Type

yara

Pattern

rule M_APT_Downloader_WINELOADER_1 { meta: author = "Mandiant" disclaimer = "This rule is meant for hunting and is not tested to run in a production environment." description = "Detects rc4 decryption logic in WINELOADER samples" strings: $ = {B9 00 01 00 00 99 F7 F9 8B 44 24 [50-200] 0F B6 00 3D FF 00 00 00} // Key initialization $ = {0F B6 00 3D FF 00 00 00} // Key size condition: all of them }

Name

d61ff2430473f06fc42a1d452597c610027aace2

Description

Detects obfuscated ROOTSAW payloads

Pattern Type

yara

Pattern

rule M_APT_Dropper_Rootsaw_Obfuscated { meta: author = "Mandiant" disclaimer = "This rule is meant for hunting and is not tested to run in a production environment." description = "Detects obfuscated ROOTSAW payloads" strings: $ = "function _" $ = "new XMLHttpRequest();" $ = "'\\x2e\\x7a\\x69\\x70'" $ = "'\\x4f\\x70\\x65\\x6e'" $ = "\\x43\\x3a\\x5c\\x57" condition: all of them }

Name

9809f2bbfff6559775bbe3f2656155515e3cd137

Description

Detects payload invocation stub in WINELOADER

Pattern Type

yara

Pattern

rule M_APT_Downloader_WINELOADER_2 { meta: author = "Mandiant" disclaimer = "This rule is meant for hunting and is not tested to run in a production environment." description = "Detects payload invocation stub in WINELOADER" strings: // 48 8D 0D ?? ?? 00 00 lea rcx, module_start (Pointer to encrypted resource) // 48 C7 C2 ?? ?? 00 00 mov rdx, ???? (size of encrypted source) // E8 [4] call decryption // 48 8D 05 [4] lea rcx, ?? // 48 8D 0D [4] lea rax, module_start (decrypted resource) // 48 89 05 [4] mov ptr_mod, rax // $ = {48 8D 0D ?? ?? 00 00 48 C7 C2 ?? ?? 00 00 E8 [4] 48 8d 0D [4] 48 8D 05 [4] 48 89 05 } condition: all of them }

Name

waterforvoiceless.org

Pattern Type

stix

Pattern

[domain-name:value = 'waterforvoiceless.org']

Name

siestakeying.com

Pattern Type

stix

Pattern

[domain-name:value = 'siestakeying.com']

Name

0x3bd487.open

Pattern Type

stix

Pattern

[domain-name:value = '0x3bd487.open']

Name

https://waterforvoiceless.org/util.php

Pattern Type

stix

Pattern

[url:value = 'https://waterforvoiceless.org/util.php']

Name

https://waterforvoiceless.org/invite.xn--php-9o0a.

Pattern Type

stix

Pattern

[url:value = 'https://waterforvoiceless.org/invite.xn--php-9o0a.']

Name

https://waterforvoiceless.org/invite.php

Pattern Type

stix

Pattern

[url:value = 'https://waterforvoiceless.org/invite.php']

Name

https://siestakeying.com/auth.php

Pattern Type

stix

Pattern

[url:value = 'https://siestakeying.com/auth.php']

Name

http://waterforvoiceless.org/util.xn--php-9o0a.

Pattern Type

stix

Pattern

[url:value = 'http://waterforvoiceless.org/util.xn--php-9o0a.']

Name

http://waterforvoiceless.org/invite.xn--php-9o0a

Pattern Type

stix

Pattern

[url:value = 'http://waterforvoiceless.org/invite.xn--php-9o0a']

Name

a0f183ea54cb25dd8bdba586935a258f0ecd3cba0d94657985bb1ea02af8d42c

Description

SUSP_obfuscated_JS_obfuscatorio SHA256 of efafcd00b9157b4146506bd381326f39

Pattern Type

stix

Pattern

[file:hashes.'SHA-256' = 'a0f183ea54cb25dd8bdba586935a258f0ecd3cba0d94657985bb1ea02af8d42c']

Malware

Name
ROOTSAW
Name
APT29
Name
Detects
Name
BURNTBATTER
Name
JSObfuscated
Name
WINELOADER

Intrusion-Set

Name

APT29

Description

[APT29](https://attack.mitre.org/groups/G0016) is threat group that has been attributed to Russia's Foreign Intelligence Service (SVR).(Citation: White House Imposing Costs RU Gov April 2021)(Citation: UK Gov Malign RIS Activity April 2021) They have operated since at least 2008, often targeting government networks in Europe and NATO member countries, research institutes, and think tanks. [APT29](https://attack.mitre.org/groups/G0016) reportedly compromised the Democratic National Committee starting in the summer of 2015.(Citation: F-Secure The Dukes)(Citation: GRIZZLY STEPPE JAR)(Citation: Crowdstrike DNC June 2016)(Citation: UK Gov UK Exposes Russia SolarWinds April 2021) In April 2021, the US and UK governments attributed the [SolarWinds Compromise](https://attack.mitre.org/campaigns/C0024) to the SVR; public statements included citations to [APT29](https://attack.mitre.org/groups/G0016), Cozy Bear, and The Dukes.(Citation: NSA Joint Advisory SVR SolarWinds April 2021)(Citation: UK NSCS Russia SolarWinds April 2021) Industry reporting also referred to the actors involved in this campaign as UNC2452, NOBELIUM, StellarParticle, Dark Halo, and SolarStorm.(Citation: FireEye SUNBURST Backdoor December 2020)(Citation: MSTIC NOBELIUM Mar 2021)(Citation: CrowdStrike SUNSPOT Implant January 2021)(Citation: Volexity SolarWinds)(Citation: Cybersecurity Advisory SVR TTP May 2021)(Citation: Unit 42 SolarStorm December 2020)

Attack-Pattern

Name
T1134
ID
T1134
Description
Adversaries may modify access tokens to operate under a different user or system security context to perform actions and bypass access controls. Windows uses access tokens to determine the ownership of a running process. A user can manipulate access tokens to make a running process appear as though it is the child of a different process or belongs to someone other than the user that started the process. When this occurs, the process also takes on the security context associated with the new token. An adversary can use built-in Windows API functions to copy access tokens from existing processes; this is known as token stealing. These token can then be applied to an existing process (i.e. [Token Impersonation/Theft](https://attack.mitre.org/techniques/T1134/001)) or used to spawn a new process (i.e. [Create Process with Token](https://attack.mitre.org/techniques/T1134/002)). An adversary must already be in a privileged user context (i.e. administrator) to steal a token. However, adversaries commonly use token stealing to elevate their security context from the administrator level to the SYSTEM level. An adversary can then use a token to authenticate to a remote system as the account for that token if the account has appropriate permissions on the remote system.(Citation: Pentestlab Token Manipulation) Any standard user can use the `runas` command, and the Windows API functions, to create impersonation tokens; it does not require access to an administrator account. There are also other mechanisms, such as Active Directory fields, that can be used to modify access tokens.

Name

T1012

ID

T1012

Description

Adversaries may interact with the Windows Registry to gather information about the system, configuration, and installed software. The Registry contains a significant amount of information about the operating system, configuration, software, and security.(Citation: Wikipedia Windows Registry) Information can easily be queried using the [Reg](https://attack.mitre.org/software/S0075) utility, though other means to access the Registry exist. Some of the information may help adversaries to further their operation within a network. Adversaries may use the information from [Query Registry](https://attack.mitre.org/techniques/T1012) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.

Name

T1574

ID

T1574

Description

Adversaries may execute their own malicious payloads by hijacking the way operating systems run programs. Hijacking execution flow can be for the purposes of persistence, since this hijacked execution may reoccur over time. Adversaries may also use these mechanisms to elevate privileges or evade defenses, such as application control or other restrictions on execution. There are many ways an adversary may hijack the flow of execution, including by manipulating how the operating system locates programs to be executed. How the operating system locates libraries to be used by a program can also be intercepted. Locations where the operating system looks for programs/resources, such as

file directories and in the case of Windows the Registry, could also be poisoned to include malicious payloads.

Name

T1110

ID

T1110

Description

Adversaries may use brute force techniques to gain access to accounts when passwords are unknown or when password hashes are obtained. Without knowledge of the password for an account or set of accounts, an adversary may systematically guess the password using a repetitive or iterative mechanism. Brute forcing passwords can take place via interaction with a service that will check the validity of those credentials or offline against previously acquired credential data, such as password hashes. Brute forcing credentials may take place at various points during a breach. For example, adversaries may attempt to brute force access to [Valid Accounts](https://attack.mitre.org/techniques/T1078) within a victim environment leveraging knowledge gathered from other post-compromise behaviors such as [OS Credential Dumping](https://attack.mitre.org/techniques/T1003), [Account Discovery](https://attack.mitre.org/techniques/T1087), or [Password Policy Discovery](https://attack.mitre.org/techniques/T1201). Adversaries may also combine brute forcing activity with behaviors such as [External Remote Services](https://attack.mitre.org/techniques/T1133) as part of Initial Access.

Name

T1057

ID

T1057

Description

Adversaries may attempt to get information about running processes on a system. Information obtained could be used to gain an understanding of common software/applications running on systems within the network. Adversaries may use the information from [Process Discovery](https://attack.mitre.org/techniques/T1057) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions. In Windows environments, adversaries could obtain details on running processes using the [Tasklist](https://attack.mitre.org/software/S0057) utility via [cmd](https://attack.mitre.org/software/S0106) or `Get-Process` via [PowerShell](https://attack.mitre.org/techniques/T1059/001). Information about processes can also be extracted from the output of [Native API](https://attack.mitre.org/techniques/T1106) calls such as `CreateToolhelp32Snapshot`. In Mac and Linux, this is accomplished with the `ps` command. Adversaries may also opt to enumerate processes via /proc. On network devices, [Network Device CLI](https://attack.mitre.org/techniques/T1059/008) commands such as `show processes` can be used to display current running processes.(Citation: US-CERT-TA18-106A)(Citation: show_processes_cisco_cmd)

Name

T1104

ID

T1104

Description

Adversaries may create multiple stages for command and control that are employed under different conditions or for certain functions. Use of multiple stages may obfuscate the command and control channel to make detection more difficult. Remote access tools will call back to the first-stage command and control server for instructions. The first stage may have automated capabilities to collect basic host information, update tools, and upload additional files. A second remote access tool (RAT) could be uploaded at that point to redirect the host to the second-stage command and control server. The second stage will likely be more fully featured and allow the adversary to interact with the system through a reverse shell and additional RAT features. The different stages will likely be hosted separately with no overlapping infrastructure. The loader may also have backup first-stage callbacks or [Fallback Channels](https://attack.mitre.org/techniques/T1008) in case the original first-stage communication path is discovered and blocked.

Name

T1083

ID

T1083

Description

Adversaries may enumerate files and directories or may search in specific locations of a host or network share for certain information within a file system. Adversaries may use the information from [File and Directory Discovery](https://attack.mitre.org/techniques/T1083) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions. Many command shell utilities can be used to obtain this information. Examples include `dir`, `tree`, `ls`, `find`, and `locate`.(Citation: Windows Commands JPCERT) Custom tools may also be used to gather file and directory information and interact with the [Native API](https://attack.mitre.org/techniques/T1106). Adversaries may also leverage a [Network Device CLI](https://attack.mitre.org/techniques/T1059/008) on network devices to gather file and directory information (e.g. `dir`, `show flash`, and/or `nvram`).(Citation: US-CERT-TA18-106A)

Name

T1070

ID

T1070

Description

Adversaries may delete or modify artifacts generated within systems to remove evidence of their presence or hinder defenses. Various artifacts may be created by an adversary or something that can be attributed to an adversary's actions. Typically these artifacts are used as defensive indicators related to monitored events, such as strings from downloaded files, logs that are generated from user actions, and other data analyzed by defenders. Location, format, and type of artifact (such as command or login history) are

often specific to each platform. Removal of these indicators may interfere with event collection, reporting, or other processes used to detect intrusion activity. This may compromise the integrity of security solutions by causing notable events to go unreported. This activity may also impede forensic analysis and incident response, due to lack of sufficient data to determine what occurred.

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

T1566

ID

T1566

Description

Adversaries may send phishing messages to gain access to victim systems. All forms of phishing are electronically delivered social engineering. Phishing can be targeted, known as spearphishing. In spearphishing, a specific individual, company, or industry will be targeted by the adversary. More generally, adversaries can conduct non-targeted phishing, such as in mass malware spam campaigns. Adversaries may send victims emails containing malicious attachments or links, typically to execute malicious code on victim systems. Phishing may also be conducted via third-party services, like social media platforms. Phishing may also involve social engineering techniques, such as posing as a trusted source, as well as evasive techniques such as removing or manipulating emails or metadata/headers from compromised accounts being abused to send messages (e.g., [Email Hiding Rules](https://attack.mitre.org/techniques/T1564/008)).(Citation: Microsoft OAuth Spam 2022)(Citation: Palo Alto Unit 42 VBA Infostealer 2014) Another way to accomplish this is by forging or spoofing(Citation: Proofpoint-spoof) the identity of the sender which can be used to fool both the human recipient as well as automated security tools.(Citation: cyberproof-double-bounce) Victims may also receive phishing messages that instruct them to call a phone number where they are directed to visit a malicious URL, download malware,(Citation: sygnia Luna Month)(Citation: CISA Remote Monitoring and Management Software) or install adversary-accessible remote management tools onto their computer (i.e., [User Execution](https://attack.mitre.org/techniques/T1204)).(Citation: Unit42 Luna Moth)

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

T1082

ID

T1082

Description

An adversary may attempt to get detailed information about the operating system and hardware, including version, patches, hotfixes, service packs, and architecture. Adversaries may use the information from [System Information Discovery](https://attack.mitre.org/techniques/T1082) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions. Tools such as [Systeminfo](https://attack.mitre.org/software/S0096) can be used to gather detailed system information. If running with privileged access, a breakdown of system data can be gathered through the `systemsetup` configuration tool on macOS. As an example, adversaries with user-level access can execute the `df -aH` command to obtain currently mounted disks and associated freely available space. Adversaries may also leverage a [Network Device CLI](https://attack.mitre.org/techniques/T1059/008) on network devices to gather detailed system information (e.g. `show version`).(Citation: US-CERT-TA18-106A) [System Information Discovery](https://attack.mitre.org/techniques/T1082) combined with information gathered from other forms of discovery and reconnaissance can drive payload development and concealment.(Citation: OSX.FairyTale)(Citation: 20 macOS Common Tools and Techniques) Infrastructure as a Service (IaaS) cloud providers such as AWS, GCP, and Azure allow access to instance and virtual machine information via APIs. Successful authenticated API calls can return data such as the operating system platform and status

of a particular instance or the model view of a virtual machine.(Citation: Amazon Describe Instance)(Citation: Google Instances Resource)(Citation: Microsoft Virutal Machine API)

Name

T1007

ID

T1007

Description

Adversaries may try to gather information about registered local system services. Adversaries may obtain information about services using tools as well as OS utility commands such as `sc query`, `tasklist /svc`, `systemctl --type=service`, and `net start`. Adversaries may use the information from [System Service Discovery](https://attack.mitre.org/techniques/T1007) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.

Name

T1547

ID

T1547

Description

Adversaries may configure system settings to automatically execute a program during system boot or logon to maintain persistence or gain higher-level privileges on compromised systems. Operating systems may have mechanisms for automatically running a program on system boot or account logon.(Citation: Microsoft Run Key)(Citation: MSDN Authentication Packages)(Citation: Microsoft TimeProvider)(Citation: Cylance Reg Persistence Sept 2013)(Citation: Linux Kernel Programming) These mechanisms may include automatically executing programs that are placed in specially designated

directories or are referenced by repositories that store configuration information, such as the Windows Registry. An adversary may achieve the same goal by modifying or extending features of the kernel. Since some boot or logon autostart programs run with higher privileges, an adversary may leverage these to elevate privileges.

Name

T1543

ID

T1543

Description

Adversaries may create or modify system-level processes to repeatedly execute malicious payloads as part of persistence. When operating systems boot up, they can start processes that perform background system functions. On Windows and Linux, these system processes are referred to as services.(Citation: TechNet Services) On macOS, launchd processes known as [Launch Daemon](https://attack.mitre.org/techniques/T1543/004) and [Launch Agent](https://attack.mitre.org/techniques/T1543/001) are run to finish system initialization and load user specific parameters.(Citation: AppleDocs Launch Agent Daemons) Adversaries may install new services, daemons, or agents that can be configured to execute at startup or a repeatable interval in order to establish persistence. Similarly, adversaries may modify existing services, daemons, or agents to achieve the same effect. Services, daemons, or agents may be created with administrator privileges but executed under root/SYSTEM privileges. Adversaries may leverage this functionality to create or modify system processes in order to escalate privileges.(Citation: OSX Malware Detection)

Country

Name
Germany
Name
Italy
Name
Latvia
Name
India
Name
Peru

Region

Name
Western Europe
Name
Southern Europe
Name
Northern Europe
Name
Europe
Name
Southern Asia
Name
Asia
Name
Latin America and the Caribbean

Name
Americas

Sector

Name
Political
Description
A recognized or non recognized political group or party taking part into the political life of a country, weither by being part of the majority or opposed to the ruling power
Name
Civil society
Description
The general public and all non-governmental entities, or individuals independent of governments, which may be linked by interests or activities aiming at promoting the interests and the will of citizens.
Name
Diplomatic
Description
Public or private entities which are actors of or involved in international relations activities.

Domain-Name

Value
waterforvoiceless.org
siestakeying.com
0x3bd487.open

Url

Value
https://waterforvoiceless.org/util.php
https://waterforvoiceless.org/invite.xn--php-9o0a.
https://waterforvoiceless.org/invite.php
https://siestakeying.com/auth.php
http://waterforvoiceless.org/util.xn--php-9o0a.
http://waterforvoiceless.org/invite.xn--php-9o0a

StixFile

Value
a0f183ea54cb25dd8bdba586935a258f0ecd3cba0d94657985bb1ea02af8d42c

External References

- https://www.mandiant.com/resources/blog/apt29-wineloader-german-political-parties
- https://otx.alienvault.com/pulse/66016e21eec7e935cc241bdd