

# NETMANAGEIT

## Intelligence Report

# A pernicious potpourri of Python packages in PyPI



# Table of contents

---

## Overview

---

● Description	4
● Confidence	4
● Content	5

---

## Entities

---

● Attack-Pattern	6
● Indicator	15
● Malware	18

---

## Observables

---

● StixFile	19
● Hostname	20
● IPv4-Addr	21



## External References

- External References

22

# Overview

## Description

Researchers have discovered more than 100 malicious packages being distributed in the official Python package repository, PyPI, over the past year, in which 80 victims have downloaded files every day, including malware.

## Confidence

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

15 / 100

# Content

N/A

# Attack-Pattern

**Name**

Boot or Logon Autostart Execution

**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**

Masquerading

**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 abusible 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**

Scheduled Task/Job

**ID**

T1053

**Description**

Adversaries may abuse task scheduling functionality to facilitate initial or recurring execution of malicious code. Utilities exist within all major operating systems to schedule programs or scripts to be executed at a specified date and time. A task can also be scheduled on a remote system, provided the proper authentication is met (ex: RPC and file and printer sharing in Windows environments). Scheduling a task on a remote system typically may require being a member of an admin or otherwise privileged group on the remote system. (Citation: TechNet Task Scheduler Security) Adversaries may use task scheduling to execute programs at system startup or on a scheduled basis for persistence. These mechanisms can also be abused to run a process under the context of a specified account (such as one with elevated permissions/privileges). Similar to [System Binary Proxy Execution](https://attack.mitre.org/techniques/T1218), adversaries have also abused task scheduling to potentially mask one-time execution under a trusted system process. (Citation: ProofPoint Serpent)

**Name**

Phishing

**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**

Credentials from Password Stores

**ID**

T1555

**Description**



Adversaries may search for common password storage locations to obtain user credentials. Passwords are stored in several places on a system, depending on the operating system or application holding the credentials. There are also specific applications and services that store passwords to make them easier for users to manage and maintain, such as password managers and cloud secrets vaults. Once credentials are obtained, they can be used to perform lateral movement and access restricted information.

**Name**

Obfuscated Files or Information

**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**

Non-Application Layer Protocol

**ID**

T1095

**Description**

Adversaries may use an OSI non-application layer protocol for communication between host and C2 server or among infected hosts within a network. The list of possible protocols is extensive.(Citation: Wikipedia OSI) Specific examples include use of network layer protocols, such as the Internet Control Message Protocol (ICMP), transport layer protocols, such as the User Datagram Protocol (UDP), session layer protocols, such as Socket Secure (SOCKS), as well as redirected/tunneled protocols, such as Serial over LAN (SOL). ICMP communication between hosts is one example.(Citation: Cisco Synful Knock Evolution) Because ICMP is part of the Internet Protocol Suite, it is required to be implemented by all IP-compatible hosts.(Citation: Microsoft ICMP) However, it is not as commonly monitored as other Internet Protocols such as TCP or UDP and may be used by adversaries to hide communications.

**Name**

Command and Scripting Interpreter

**ID**

T1059

**Description**

Adversaries may abuse command and script interpreters to execute commands, scripts, or binaries. These interfaces and languages provide ways of interacting with computer systems and are a common feature across many different platforms. Most systems come with some built-in command-line interface and scripting capabilities, for example, macOS and Linux distributions include some flavor of [Unix Shell](<https://attack.mitre.org/techniques/T1059/004>) while Windows installations include the [Windows Command Shell]

(<https://attack.mitre.org/techniques/T1059/003>) and [PowerShell](<https://attack.mitre.org/techniques/T1059/001>). There are also cross-platform interpreters such as [Python](<https://attack.mitre.org/techniques/T1059/006>), as well as those commonly associated with client applications such as [JavaScript](<https://attack.mitre.org/techniques/T1059/007>) and [Visual Basic](<https://attack.mitre.org/techniques/T1059/005>). Adversaries may abuse these technologies in various ways as a means of executing arbitrary commands. Commands and scripts can be embedded in [Initial Access](<https://attack.mitre.org/tactics/TA0001>) payloads delivered to victims as lure documents or as secondary payloads downloaded from an existing C2. Adversaries may also execute commands through interactive terminals/shells, as well as utilize various [Remote Services](<https://attack.mitre.org/techniques/T1021>) in order to achieve remote Execution. (Citation: Powershell Remote Commands)(Citation: Cisco IOS Software Integrity Assurance - Command History)(Citation: Remote Shell Execution in Python)

**Name**

Supply Chain Compromise

**ID**

T1195

**Description**

Adversaries may manipulate products or product delivery mechanisms prior to receipt by a final consumer for the purpose of data or system compromise. Supply chain compromise can take place at any stage of the supply chain including: \* Manipulation of development tools \* Manipulation of a development environment \* Manipulation of source code repositories (public or private) \* Manipulation of source code in open-source dependencies \* Manipulation of software update/distribution mechanisms \* Compromised/infected system images (multiple cases of removable media infected at the factory)(Citation: IBM Storwize)(Citation: Schneider Electric USB Malware) \* Replacement of legitimate software with modified versions \* Sales of modified/counterfeit products to legitimate distributors \* Shipment interdiction While supply chain compromise can impact any component of hardware or software, adversaries looking to gain execution have often focused on malicious additions to legitimate software in software distribution or update channels.(Citation: Avast CCleaner3 2018)(Citation: Microsoft Dofail 2018)(Citation: Command Five SK 2011) Targeting may be specific to a desired victim set or malicious software may be distributed to a broad set of consumers but only move on to additional tactics on specific victims.(Citation: Symantec Elderwood Sept 2012)(Citation: Avast CCleaner3 2018)(Citation: Command Five SK 2011) Popular open source projects that are

used as dependencies in many applications may also be targeted as a means to add malicious code to users of the dependency.(Citation: Trendmicro NPM Compromise)

**Name**

Remote Services

**ID**

T1021

**Description**

Adversaries may use [Valid Accounts](<https://attack.mitre.org/techniques/T1078>) to log into a service that accepts remote connections, such as telnet, SSH, and VNC. The adversary may then perform actions as the logged-on user. In an enterprise environment, servers and workstations can be organized into domains. Domains provide centralized identity management, allowing users to login using one set of credentials across the entire network. If an adversary is able to obtain a set of valid domain credentials, they could login to many different machines using remote access protocols such as secure shell (SSH) or remote desktop protocol (RDP).(Citation: SSH Secure Shell)(Citation: TechNet Remote Desktop Services) They could also login to accessible SaaS or IaaS services, such as those that federate their identities to the domain. Legitimate applications (such as [Software Deployment Tools](<https://attack.mitre.org/techniques/T1072>) and other administrative programs) may utilize [Remote Services](<https://attack.mitre.org/techniques/T1021>) to access remote hosts. For example, Apple Remote Desktop (ARD) on macOS is native software used for remote management. ARD leverages a blend of protocols, including [VNC](<https://attack.mitre.org/techniques/T1021/005>) to send the screen and control buffers and [SSH](<https://attack.mitre.org/techniques/T1021/004>) for secure file transfer. (Citation: Remote Management MDM macOS)(Citation: Kickstart Apple Remote Desktop commands)(Citation: Apple Remote Desktop Admin Guide 3.3) Adversaries can abuse applications such as ARD to gain remote code execution and perform lateral movement. In versions of macOS prior to 10.14, an adversary can escalate an SSH session to an ARD session which enables an adversary to accept TCC (Transparency, Consent, and Control) prompts without user interaction and gain access to data.(Citation: FireEye 2019 Apple Remote Desktop)(Citation: Lockboxx ARD 2019)(Citation: Kickstart Apple Remote Desktop commands)

**Name**

## Deobfuscate/Decode Files or Information

**ID**

T1140

**Description**

Adversaries may use [Obfuscated Files or Information](<https://attack.mitre.org/techniques/T1027>) to hide artifacts of an intrusion from analysis. They may require separate mechanisms to decode or deobfuscate that information depending on how they intend to use it. Methods for doing that include built-in functionality of malware or by using utilities present on the system. One such example is the use of [certutil](<https://attack.mitre.org/software/S0160>) to decode a remote access tool portable executable file that has been hidden inside a certificate file. (Citation: Malwarebytes Targeted Attack against Saudi Arabia) Another example is using the Windows `copy /b`` command to reassemble binary fragments into a malicious payload. (Citation: Carbon Black Obfuscation Sept 2016) Sometimes a user's action may be required to open it for deobfuscation or decryption as part of [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)

**Name**

Screen Capture

**ID**

T1113

**Description**

Adversaries may attempt to take screen captures of the desktop to gather information over the course of an operation. Screen capturing functionality may be included as a feature of a remote access tool used in post-compromise operations. Taking a screenshot is also typically possible through native utilities or API calls, such as `CopyFromScreen``,

``xwd``, or ``screenshot``.(Citation: CopyFromScreen .NET)(Citation: Antiquated Mac Malware)

**Name**

Clipboard Data

**ID**

T1115

**Description**

Adversaries may collect data stored in the clipboard from users copying information within or between applications. For example, on Windows adversaries can access clipboard data by using ``clip.exe`` or ``Get-Clipboard``.(Citation: MSDN Clipboard)(Citation: clip\_win\_server)(Citation: CISA\_AA21\_200B) Additionally, adversaries may monitor then replace users' clipboard with their data (e.g., [Transmitted Data Manipulation](<https://attack.mitre.org/techniques/T1565/002>)).(Citation: mining\_ruby\_reversinglabs) macOS and Linux also have commands, such as ``pbpaste``, to grab clipboard contents.(Citation: Operating with EmPyre)

# Indicator

## Name

104a5192cf032cee44b732d33458a27909cef45d7391e092b9c13acd5779bb39

## Description

is\_\_elf SHA256 of ef59c159d3fd668c3963e5ade3c726b8771e6f54

## Pattern Type

stix

## Pattern

[file:hashes:'SHA-256' =  
'104a5192cf032cee44b732d33458a27909cef45d7391e092b9c13acd5779bb39']

## Name

204.152.203.78

## Description

\*\*ISP:\*\* QuadraNet Enterprises LLC \*\*OS:\*\* None ----- Hostnames: -  
buzz-203078.buzzford.com ----- Domains: - buzzford.com  
----- Services: \*\*22:\*\* ~~~ SSH-2.0-OpenSSH\_8.9p1 Ubuntu-3ubuntu0.4 Key  
type: ecdsa-sha2-nistp256 Key:  
AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBjXjYDvUyfibD+nBWzJXsS12  
oPCb60rKqzRQK6dwDiuQfLK2dhMYuCOzbzN9EkbntY6UmTVvz0nXEdC5+l0d8Y= Fingerprint:

```

89:37:18:3d:8d:27:b1:21:93:bb:c6:ce:00:25:c2:c4 Kex Algorithms: curve25519-sha256 curve25519-
sha256@libssh.org ecdh-sha2-nistp256 ecdh-sha2-nistp384 ecdh-sha2-nistp521
sntrup761x25519-sha512@openssh.com diffie-hellman-group-exchange-sha256 diffie-
hellman-group16-sha512 diffie-hellman-group18-sha512 diffie-hellman-group14-sha256
Server Host Key Algorithms: rsa-sha2-512 rsa-sha2-256 ecdsa-sha2-nistp256 ssh-ed25519
Encryption Algorithms: chacha20-poly1305@openssh.com aes128-ctr aes192-ctr aes256-ctr
aes128-gcm@openssh.com aes256-gcm@openssh.com MAC Algorithms: umac-64-
etm@openssh.com umac-128-etm@openssh.com hmac-sha2-256-etm@openssh.com
hmac-sha2-512-etm@openssh.com hmac-sha1-etm@openssh.com umac-64@openssh.com
umac-128@openssh.com hmac-sha2-256 hmac-sha2-512 hmac-sha1 Compression
Algorithms: none zlib@openssh.com ~~~ ----- **80:** ~~~ HTTP/1.1 404 Not Found
Content-Type: text/plain; charset=utf-8 Vary: Accept-Encoding X-Content-Type-Options:
nosniff Date: Thu, 07 Dec 2023 11:23:59 GMT Content-Length: 19 ~~~ ----- **631:**
~~~ HTTP/1.1 403 Forbidden Connection: close Content-Language: en Content-Length: 370
Content-Type: text/html; charset=utf-8 Date: Thu, 07 Dec 2023 19:35:14 GMT Accept-
Encoding: gzip, deflate, identity Server: CUPS/2.4 IPP/2.1 X-Frame-Options: DENY Content-
Security-Policy: frame-ancestors 'none' ~~~ ----- **3389:** ~~~ Remote Desktop
Protocol
\x03\x00\x00\x13\x0e\xd0\x00\x00\x124\x00\x02\x01\x08\x00\x00\x00\x00\x00 ~~~
-----

```

**Pattern Type**

stix

**Pattern**

[ipv4-addr:value = '204.152.203.78']

**Name**

blazywound.ignorelist.com

**Pattern Type**

stix

**Pattern**



[hostname:value = 'blazywound.ignorelist.com']

# Malware

**Name**

W4SP

# StixFile

## Value

104a5192cf032cee44b732d33458a27909cef45d7391e092b9c13acd5779bb39

# Hostname

## Value

blazywound.ignorelist.com

# IPv4-Addr

## Value

204.152.203.78

# External References

- 
- <https://otx.alienvault.com/pulse/657b710c1b313b8547fa4145>
- 
- <https://www.welivesecurity.com/en/eset-research/ Pernicious-potpourri-python-packages-pypi/>