Turla, IRON HUNTER, Group 88, Waterbug, WhiteBear, Snake, Krypton, Venomous Bear, Secret Blizzard, BELUGASTURGEON, Group G0010

Archived: 2026-04-05 13:28:58 UTC

Enterprise T1134 .002 Access Token Manipulation: Create Process with Token

Turla RPC backdoors can impersonate or steal process tokens before executing commands.[11]

Enterprise T1087 .001 Account Discovery: Local Account

Turla has used `net user` to enumerate local accounts on the system.[14][15]

.002 Account Discovery: Domain Account

Turla has used `net user /domain` to enumerate domain accounts.[14]

Enterprise T1583 .006 Acquire Infrastructure: Web Services

Turla has created web accounts including Dropbox and GitHub for C2 and document exfiltration.[15]

Enterprise T1071 .001 Application Layer Protocol: Web Protocols

Turla has used HTTP and HTTPS for C2 communications.[4][16]

.003 Application Layer Protocol: Mail Protocols

Turla has used multiple backdoors which communicate with a C2 server via email attachments.[17]

Enterprise T1560 .001 Archive Collected Data: Archive via Utility

Turla has encrypted files stolen from connected USB drives into a RAR file before exfiltration.[18]

Enterprise T1547 .001 Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder

A Turla Javascript backdoor added a local_update_check value under the Registry key `HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run` to establish persistence. Additionally, a Turla custom executable containing Metasploit shellcode is saved to the Startup folder to gain persistence.[4][16][19]

.004 Boot or Logon Autostart Execution: Winlogon Helper DLL

Turla established persistence by adding a Shell value under the Registry key `HKCU\Software\Microsoft\Windows NT\CurrentVersion\Winlogon`.[4]

Enterprise T1110 Brute Force

Turla may attempt to connect to systems within a victim's network using `net use` commands and a predefined list or collection of passwords.[1]

Enterprise T1059 .001 Command and Scripting Interpreter: PowerShell

Turla has used PowerShell to execute commands/scripts, in some cases via a custom executable or code from Empire's PSInject.[16][11][18] Turla has also used PowerShell scripts to load and execute malware in memory.

.003 Command and Scripting Interpreter: Windows Command Shell

Turla RPC backdoors have used cmd.exe to execute commands.[11][18]

.005 Command and Scripting Interpreter: Visual Basic

Turla has used VBS scripts throughout its operations.[18]

.006 Command and Scripting Interpreter: Python

Turla has used IronPython scripts as part of the IronNetInjector toolchain to drop payloads.[20]

.007 Command and Scripting Interpreter: JavaScript

Turla has used various JavaScript-based backdoors.[4]

Enterprise T1584 .003 Compromise Infrastructure: Virtual Private Server

Turla has used the VPS infrastructure of compromised Iranian threat actors.[21]

.004 Compromise Infrastructure: Server

Turla has used compromised servers as infrastructure.[22][13][10]

.006 Compromise Infrastructure: Web Services

Turla has frequently used compromised WordPress sites for C2 infrastructure.[22]

Enterprise T1555 .004 Credentials from Password Stores: Windows Credential Manager

Turla has gathered credentials from the Windows Credential Manager tool.[18]

Enterprise T1213 .006 Data from Information Repositories: Databases

Turla has used a custom .NET tool to collect documents from an organization's internal central database.[14]

Enterprise T1005 Data from Local System

Turla RPC backdoors can upload files from victim machines.[11]

Enterprise T1025 Data from Removable Media

Turla RPC backdoors can collect files from USB thumb drives.[11][18]

Enterprise T1140 Deobfuscate/Decode Files or Information

Turla has used a custom decryption routine, which pulls key and salt values from other artifacts such as a WMI filter or PowerShell Profile, to decode encrypted PowerShell payloads.[11]

Enterprise T1587 .001 Develop Capabilities: Malware

Turla has developed its own unique malware for use in operations.[22]

Enterprise T1189 Drive-by Compromise

Turla has infected victims using watering holes.[14][6]

Enterprise T1546 .003 Event Triggered Execution: Windows Management Instrumentation Event Subscription

Turla has used WMI event filters and consumers to establish persistence.[11]

.013 Event Triggered Execution: PowerShell Profile

Turla has used PowerShell profiles to maintain persistence on an infected machine.[11]

Enterprise T1567 .002 Exfiltration Over Web Service: Exfiltration to Cloud Storage

Turla has used WebDAV to upload stolen USB files to a cloud drive.[18] Turla has also exfiltrated stolen files to OneDrive and 4shared.[14]

Enterprise T1068 Exploitation for Privilege Escalation

Turla has exploited vulnerabilities in the VBoxDrv.sys driver to obtain kernel mode privileges.[23]

Enterprise T1083 File and Directory Discovery

Turla surveys a system upon check-in to discover files in specific locations on the hard disk %TEMP% directory, the current user's desktop, the Program Files directory, and Recent.[1][14] Turla RPC backdoors have also searched for files matching the `lPH*.dll` pattern.[11]

Enterprise T1615 Group Policy Discovery

Turla surveys a system upon check-in to discover Group Policy details using the `gpresult` command.[14]

Enterprise T1564 .012 Hide Artifacts: File/Path Exclusions

Turla has placed LunarWeb install files into directories that are excluded from scanning.[19]

Enterprise T1562 .001 Impair Defenses: Disable or Modify Tools

Turla has used a AMSI bypass, which patches the in-memory amsi.dll, in PowerShell scripts to bypass Windows antimalware products.[11]

Enterprise T1105 Ingress Tool Transfer

Turla has used shellcode to download Meterpreter after compromising a victim.[16]

Enterprise T1570 Lateral Tool Transfer

Turla RPC backdoors can be used to transfer files to/from victim machines on the local network.[11][18]

Enterprise T1036 .005 Masquerading: Match Legitimate Resource Name or Location

Turla has named components of LunarWeb to mimic Zabbix agent logs.[19]

Enterprise T1112 Modify Registry

Turla has modified Registry values to store payloads.[11][18]

Enterprise T1106 Native API

Turla and its RPC backdoors have used APIs calls for various tasks related to subverting AMSI and accessing then executing commands through RPC and/or named pipes.[11]

Enterprise T1027 .005 Obfuscated Files or Information: Indicator Removal from Tools

Based on comparison of Gazer versions, Turla made an effort to obfuscate strings in the malware that could be used as IoCs, including the mutex name and named pipe.[2]

.010 Obfuscated Files or Information: Command Obfuscation

Turla has used encryption (including salted 3DES via PowerSploit's `Out-EncryptedScript.ps1`), random variable names, and base64 encoding to obfuscate PowerShell commands and payloads.[11]

.011 Obfuscated Files or Information: Fileless Storage

Turla has used the Registry to store encrypted and encoded payloads.[11][18]

Enterprise T1588 .001 Obtain Capabilities: Malware

Turla has used malware obtained after compromising other threat actors, such as OilRig.[21][22]

.002 Obtain Capabilities: Tool

Turla has obtained and customized publicly-available tools like Mimikatz.[18]

Enterprise T1201 Password Policy Discovery

Turla has used `net accounts` and `net accounts /domain` to acquire password policy information.[14]

Enterprise T1120 Peripheral Device Discovery

Turla has used `fsutil fsinfo drives` to list connected drives.[14]

Enterprise T1069 .001 Permission Groups Discovery: Local Groups

Turla has used `net localgroup` and `net localgroup Administrators` to enumerate group information, including members of the local administrators group.[14]

.002 Permission Groups Discovery: Domain Groups

Turla has used `net group "Domain Admins" /domain` to identify domain administrators.[14]

Enterprise T1566 .002 Phishing: Spearphishing Link

Turla attempted to trick targets into clicking on a link featuring a seemingly legitimate domain from Adobe.com to download their malware and gain initial access.[4]

Enterprise T1057 Process Discovery

Turla surveys a system upon check-in to discover running processes using the `tasklist /v` command.[1] Turla RPC backdoors have also enumerated processes associated with specific open ports or named pipes.[11]

Enterprise T1055 Process Injection

Turla has also used PowerSploit's `Invoke-ReflectivePEInjection.ps1` to reflectively load a PowerShell payload into a random process on the victim system.[11]

.001 Dynamic-link Library Injection

Turla has used Metasploit to perform reflective DLL injection in order to escalate privileges.[16][24]

Enterprise T1090 Proxy

Turla RPC backdoors have included local UPnP RPC proxies.[11]

.001 Internal Proxy

Turla has compromised internal network systems to act as a proxy to forward traffic to C2.[10]

Enterprise T1012 Query Registry

Turla surveys a system upon check-in to discover information in the Windows Registry with the `reg query` command.[1] Turla has also retrieved PowerShell payloads hidden in Registry keys as well as checking keys associated with null session named pipes .[11]

Enterprise T1021 .002 Remote Services: SMB/Windows Admin Shares

Turla used `net use` commands to connect to lateral systems within a network.[1]

Enterprise T1018 Remote System Discovery

Turla surveys a system upon check-in to discover remote systems on a local network using the `net view` and `net view /DOMAIN` commands. Turla has also used `net group "Domain Computers" /domain`, `net group "Domain Controllers" /domain`, and `net group "Exchange Servers" /domain` to enumerate domain computers, including the organization's DC and Exchange Server.[1][14]

Enterprise T1518 .001 Software Discovery: Security Software Discovery

Turla has obtained information on security software, including security logging information that may indicate whether their malware has been detected.[14]

Enterprise T1553 .006 Subvert Trust Controls: Code Signing Policy Modification

Turla has modified variables in kernel memory to turn off Driver Signature Enforcement after exploiting vulnerabilities that obtained kernel mode privileges.[23][25]

Enterprise T1082 System Information Discovery

Turla surveys a system upon check-in to discover operating system configuration details using the `systeminfo` and `set` commands.[1][14]

Enterprise T1016 System Network Configuration Discovery

Turla surveys a system upon check-in to discover network configuration details using the `arp -a`, `nbtstat -n`, `net config`, `ipconfig /all`, and `route` commands, as well as NBTscan.[1][18][14] Turla RPC backdoors have also retrieved registered RPC interface information from process memory.[11]

.001 Internet Connection Discovery

Turla has used `tracert` to check internet connectivity.[14]

Enterprise T1049 System Network Connections Discovery

Turla surveys a system upon check-in to discover active local network connections using the `netstat -an`, `net use`, `net file`, and `net session` commands.[1][14] Turla RPC backdoors have also enumerated the IPv4 TCP connection table via the `GetTcpTable2` API call.[11]

Enterprise T1007 System Service Discovery

Turla surveys a system upon check-in to discover running services and associated processes using the `tasklist /svc` command.[1]

Enterprise T1124 System Time Discovery

Turla surveys a system upon check-in to discover the system time by using the `net time` command.[1]

Enterprise T1204 .001 User Execution: Malicious Link

Turla has used spearphishing via a link to get users to download and run their malware.[4]

Enterprise T1078 .003 Valid Accounts: Local Accounts

Turla has abused local accounts that have the same password across the victim's network.[15]

Enterprise T1102 Web Service

Turla has used legitimate web services including Pastebin, Dropbox, and GitHub for C2 communications.[13][15]

.002 Bidirectional Communication

A Turla JavaScript backdoor has used Google Apps Script as its C2 server.[4][16]
