Saint Bot, Software S1018 | MITRE ATT&CK®

Archived: 2026-04-05 15:35:23 UTC

Enterprise T1548 .002 Abuse Elevation Control Mechanism: Bypass User Account Control

Saint Bot has attempted to bypass UAC using `fodhelper.exe` to escalate privileges.[2]

Enterprise T1071 .001 Application Layer Protocol: Web Protocols

Saint Bot has used HTTP for C2 communications.[1]

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

Saint Bot has established persistence by being copied to the Startup directory or through the `\Software\Microsoft\Windows\CurrentVersion\Run` registry key.[1][2]

Enterprise T1059 .001 Command and Scripting Interpreter: PowerShell

Saint Bot has used PowerShell for execution.[2]

.003 Command and Scripting Interpreter: Windows Command Shell

Saint Bot has used `cmd.exe` and `.bat` scripts for execution.[2]

.005 Command and Scripting Interpreter: Visual Basic

Saint Bot has used `.vbs` scripts for execution.[2]

Enterprise T1132 .001 Data Encoding: Standard Encoding

Saint Bot has used Base64 to encode its C2 communications.[1]

Enterprise T1005 Data from Local System

Saint Bot can collect files and information from a compromised host.[1]

Enterprise T1622 Debugger Evasion

Saint Bot has used `is_debugger_present` as part of its environmental checks.[1]

Enterprise T1140 Deobfuscate/Decode Files or Information

Saint Bot can deobfuscate strings and files for execution.[1]

Enterprise T1083 File and Directory Discovery

Saint Bot can search a compromised host for specific files.[2]

Enterprise T1574 Hijack Execution Flow

Saint Bot will use the malicious file `slideshow.mp4` if present to load the core API provided by `ntdll.dll` to avoid any hooks placed on calls to the original `ntdll.dll` file by endpoint detection and response or antimalware software.[2]

Enterprise T1070 .004 Indicator Removal: File Deletion

Saint Bot can run a batch script named `del.bat` to remove any Saint Bot payload-linked files from a compromise system if anti-analysis or locale checks fail.[2]

Enterprise T1105 Ingress Tool Transfer

Saint Bot can download additional files onto a compromised host.[2]

Enterprise T1036 Masquerading

Saint Bot has renamed malicious binaries as `wallpaper.mp4` and `slideshow.mp4` to avoid detection.[1][2]

.005 Match Legitimate Resource Name or Location

Saint Bot has been disguised as a legitimate executable, including as Windows SDK.[1]

Enterprise T1106 Native API

Saint Bot has used different API calls, including `GetProcAddress`, `VirtualAllocEx`, `WriteProcessMemory`, `CreateProcessA`, and `SetThreadContext`.[1][2]

Enterprise T1027 Obfuscated Files or Information

Saint Bot has been obfuscated to help avoid detection.[2]

.002 Software Packing

Saint Bot has been packed using a dark market crypter.[1]

Enterprise T1566 .001 Phishing: Spearphishing Attachment

Saint Bot has been distributed as malicious attachments within spearphishing emails.[1][2]

.002 Phishing: Spearphishing Link

Saint Bot has been distributed through malicious links contained within spearphishing emails.[2]

Enterprise T1057 Process Discovery

Saint Bot has enumerated running processes on a compromised host to determine if it is running under the process name `dfrgui.exe`.[2]

Enterprise T1055 .001 Process Injection: Dynamic-link Library Injection

Saint Bot has injected its DLL component into `EhStorAurhn.exe`.[1]

.004 Process Injection: Asynchronous Procedure Call

Saint Bot has written its payload into a newly-created `EhStorAuthn.exe` process using `ZwWriteVirtualMemory` and executed it using `NtQueueApcThread` and `ZwAlertResumeThread`.[1]

.012 Process Injection: Process Hollowing

The Saint Bot loader has used API calls to spawn `MSBuild.exe` in a suspended state before injecting the decrypted Saint Bot binary into it.[2]

Enterprise T1012 Query Registry

Saint Bot has used `check_registry_keys` as part of its environmental checks.[1]

Enterprise T1053 .005 Scheduled Task/Job: Scheduled Task

Saint Bot has created a scheduled task named "Maintenance" to establish persistence.[1]

Enterprise T1218 .004 System Binary Proxy Execution: InstallUtil

Saint Bot had used `InstallUtil.exe` to download and deploy executables.[1]

.010 System Binary Proxy Execution: Regsvr32

Saint Bot has used `regsvr32` to execute scripts.[1][2]

Enterprise T1082 System Information Discovery

Saint Bot can identify the OS version, CPU, and other details from a victim's machine.[1]

Enterprise T1614 System Location Discovery

Saint Bot has conducted system locale checks to see if the compromised host is in Russia, Ukraine, Belarus, Armenia, Kazakhstan, or Moldova.[1][2]

Enterprise T1016 System Network Configuration Discovery

Saint Bot can collect the IP address of a victim machine.[1]

Enterprise T1033 System Owner/User Discovery

Saint Bot can collect the username from a compromised host.[1]

Enterprise T1204 .001 User Execution: Malicious Link

Saint Bot has relied on users to click on a malicious link delivered via a spearphishing.[2]

.002 User Execution: Malicious File

Saint Bot has relied on users to execute a malicious attachment delivered via spearphishing.[1][2]

Enterprise T1497 .001 Virtualization/Sandbox Evasion: System Checks

Saint Bot has run several virtual machine and sandbox checks, including checking if `Sbiedll.dll` is present in a list of loaded modules, comparing the machine name to `HAL9TH` and the user name to `JohnDoe`, and checking the BIOS version for known virtual machine identifiers.[2]

.003 Virtualization/Sandbox Evasion: Time Based Checks

Saint Bot has used the command `timeout 20` to pause the execution of its initial loader.[2]
