BLACKSWAN CYBERSECURITY

THREAT ADVISORY

Deep#Door Python-based Backdoor Framework

May 7, 2026

Background/Summary

DEEP#DOOR (also referred to as Deep#Door or DeepDoor), has been identified as a sophisticated, stealthy Python-based backdoor framework targeting Windows systems. It is delivered primarily through phishing emails or malicious documents/scripts that contain an obfuscated batch script (install_obf.bat) that embeds the full Python payload (svc.py) directly within it, eliminating the need for external downloads.

The malware establishes multi-layered persistence and uses advanced evasion techniques to maintain long-term access. It supports extensive espionage capabilities (surveillance, credential theft) and can pivot to destructive actions. Communication relies on the public bore.pub TCP tunneling service, allowing attackers to avoid maintaining dedicated C2 infrastructure.

The objectives are targeted espionage, credential harvesting (especially browser/cloud/SSH), and potential disruption.

Threat

- **Actors**: Unattributed but consistent with advanced persistent threats (APTs) or sophisticated cybercrime groups focused on long-term espionage and data theft.
- **Primary Objective**: Persistent remote access for intelligence collection (credentials, surveillance) and potential sabotage.
- **Key Innovation**: Self-contained dropper with embedded payload + public tunneling (bore.pub) reduces infrastructure footprint and detection surface.

Risk

- **High** for organizations with Windows endpoints exposed to phishing or supply-chain risks.
- Particularly dangerous due to credential theft (browsers, cloud providers like AWS/Azure/GCP, SSH keys) enabling lateral movement and further compromise.
- Stealth features make it a “detection nightmare” for traditional antivirus/EDR without strong behavioral monitoring.

Impact

- Full remote command execution (RAT capabilities).
- Comprehensive surveillance: keylogging, clipboard monitoring, screenshots, webcam/microphone access.
- Broad credential harvesting and system reconnaissance.
- Destructive potential: Master Boot Record (MBR) overwrite, forced crashes, resource exhaustion.
- Long-term persistence with self-healing mechanisms, complicating remediation.

IOCs

File/Directory Indicators:

- **Dropper**: install_obf.bat (heavily obfuscated batch script).

- **Payload**: svc.py (dropped to %LOCALAPPDATA%\SystemServices\svc.py — mimics legitimate Windows services).
- **Other common paths**: Startup folder scripts, registry Run keys.

Network / C2:
- Domain: bore.pub (and subdomains/variants used for tunneling).
- Ports: Dynamically generated range 41234–41243.
- Traffic: TCP tunneling with challenge-response authentication; blends with legitimate outbound connections.

Behavioral / Registry:
- PowerShell commands referencing %~f0 (self-file reading).
- Modifications disabling Windows Defender (real-time monitoring, tamper protection), AMSI, ETW, PowerShell logging, Event Logs, SmartScreen, and firewall logging.
- Creation of scheduled tasks, WMI event subscriptions, and watchdog threads for persistence.

Hashes: Specific SHA256 hashes are detailed in the full Securonix report (not publicly enumerated in summaries; hunt for files matching the described behavior and paths).

TTPs

- **Initial Access & Execution**: Obfuscated batch script via phishing; self-referential payload extraction (reads own file contents using regex delimiters #PYTHON_START / #PYTHON_END).
- **Defense Evasion**: Disables Defender, AMSI/ETW patching, ntdll unhooking, sandbox/VM/debugger detection, command-line wiping, timestamp stomping, log clearing. Process and path exclusions for Python executables.
- **Persistence**: Multi-layered — Startup folder, Registry Run keys, Scheduled Tasks, WMI subscriptions, watchdog/self-healing threads.
- **Command & Control**: Public bore.pub TCP tunneling; dynamic ports; resilient connection retries.
- **Discovery & Collection**: System/network reconnaissance, credential dumping (browsers, Cloud tokens, SSH keys, Wi-Fi, Credential Manager).
- **Impact**: Surveillance (keylogger, screen/webcam/mic), file operations, destructive commands (MBR overwrite, crashes).
- **MITRE ATT&CK** (approximate mapping):
  - T1059 (Command and Scripting Interpreter)
  - T1547/T1053 (Persistence mechanisms)
  - T1562 (Impair Defenses)
  - T1572 (Protocol Tunneling)
  - T1115/T1113/T1123/T1125 (Surveillance/collection)
  - T1555 (Credentials from Password Stores)
  - T1485/T1489 (Data/Endpoint Destruction)

THREAT ADVISORY

Deep#Door Python-based Backdoor Framework

May 7, 2026

Recommendations

1. **Immediate Detection**:
   - Block or alert on outbound connections to bore.pub and ports 41234–41243.
   - Hunt for install_obf.bat, %LOCALAPPDATA%\SystemServices\svc.py, and unusual Python processes.
   - Monitor for PowerShell disabling Defender/logging and registry changes.

2. **Prevention**:
   - Strengthen phishing defenses and script execution controls (e.g., Applocker, WDAC).
   - Enable and monitor AMSI, ETW, and full Defender protections.
   - Restrict unnecessary outbound traffic and public tunneling services.

3. **Response & Hardening**:
   - Investigate endpoints for the described persistence artifacts.
   - Enforce least-privilege; monitor privileged credential usage.
   - Update EDR/SIEM rules with behavioral indicators from the Securonix report.
   - Regular credential rotation (especially cloud/SSH) and endpoint hygiene.

Organizations should prioritize behavioral detection and network egress controls, as signature-based methods are less effective against this framework.

Contact Blackswan Cybersecurity at 855-BLK-SWAN or
Contact@BlackswanCybersecurity.com