EvilBunny, Software S0396 | MITRE ATT&CK®

Domain	ID		Name	Use
Enterprise	T1071	.001	Application Layer Protocol: Web Protocols	EvilBunny has executed C2 commands directly via HTTP.[1]
Enterprise	T1547	.001	Boot or Logon Autostart Execution: Registry Run Keys / Startup Folder	EvilBunny has created Registry keys for persistence in `[HKLM|HKCU]\…\CurrentVersion\Run`.[1]
Enterprise	T1059	.003	Command and Scripting Interpreter: Windows Command Shell	EvilBunny has an integrated scripting engine to download and execute Lua scripts.[1]
		.011	Command and Scripting Interpreter: Lua	EvilBunny has used Lua scripts to execute payloads.[2]
Enterprise	T1203		Exploitation for Client Execution	EvilBunny has exploited CVE-2011-4369, a vulnerability in the PRC component in Adobe Reader.[1]
Enterprise	T1070	.004	Indicator Removal: File Deletion	EvilBunny has deleted the initial dropper after running through the environment checks.[1]
Enterprise	T1105		Ingress Tool Transfer	EvilBunny has downloaded additional Lua scripts from the C2.[1]
Enterprise	T1106		Native API	EvilBunny has used various API calls as part of its checks to see if the malware is running in a sandbox.[1]

Domain	ID		Name	Use
Enterprise	T1057		Process Discovery	EvilBunny has used EnumProcesses() to identify how many process are running in the environment.[1]
Enterprise	T1053	.005	Scheduled Task/Job: Scheduled Task	EvilBunny has executed commands via scheduled tasks.[1]
Enterprise	T1518	.001	Software Discovery: Security Software Discovery	EvilBunny has been observed querying installed antivirus software.[1]
Enterprise	T1124		System Time Discovery	EvilBunny has used the API calls NtQuerySystemTime, GetSystemTimeAsFileTime, and GetTickCount to gather time metrics as part of its checks to see if the malware is running in a sandbox.[1]
Enterprise	T1497	.001	Virtualization/Sandbox Evasion: System Checks	EvilBunny's dropper has checked the number of processes and the length and strings of its own file name to identify if the malware is in a sandbox environment.[1]
		.003	Virtualization/Sandbox Evasion: Time Based Checks	EvilBunny has used time measurements from 3 different APIs before and after performing sleep operations to check and abort if the malware is running in a sandbox.[1]
Enterprise	T1047		Windows Management Instrumentation	EvilBunny has used WMI to gather information about the system.[1]