Ebury, Software S0377 | MITRE ATT&CK®

Archived: 2026-04-05 16:23:22 UTC

Enterprise T1071 .004 Application Layer Protocol: DNS

Ebury has used DNS requests over UDP port 53 for C2.[1]

Enterprise T1020 Automated Exfiltration

If credentials are not collected for two weeks, Ebury encrypts the credentials using a public key and sends them via UDP to an IP address located in the DNS TXT record.[5][4]

Enterprise T1059 .004 Command and Scripting Interpreter: Unix Shell

Ebury can use the commands `Xcsh` or `Xcls` to open a shell with Ebury level permissions and `Xxsh` to open a shell with root level.[4]

.006 Command and Scripting Interpreter: Python

Ebury has used Python to implement its DGA.[3]

Enterprise T1554 Compromise Host Software Binary

Ebury modifies the `keyutils` library to add malicious behavior to the OpenSSH client and the curl library.[1][4]

Enterprise T1132 .001 Data Encoding: Standard Encoding

Ebury has encoded C2 traffic in hexadecimal format.[1]

Enterprise T1140 Deobfuscate/Decode Files or Information

Ebury has verified C2 domain ownership by decrypting the TXT record using an embedded RSA public key.[3]

Enterprise T1568 .002 Dynamic Resolution: Domain Generation Algorithms

Ebury has used a DGA to generate a domain name for C2.[1][3]

Enterprise T1573 .001 Encrypted Channel: Symmetric Cryptography

Ebury has encrypted C2 traffic using the client IP address, then encoded it as a hexadecimal string.[1]

Enterprise T1041 Exfiltration Over C2 Channel

Ebury exfiltrates a list of outbound and inbound SSH sessions using OpenSSH's `known_host` files and `wtmp` records. Ebury can exfiltrate SSH credentials through custom DNS queries or use the command `Xcat` to send the process's ssh session's credentials to the C2 server.[5][4]

Enterprise T1008 Fallback Channels

Ebury has implemented a fallback mechanism to begin using a DGA when the attacker hasn't connected to the infected system for three days.[3]

Enterprise T1574 .006 Hijack Execution Flow: Dynamic Linker Hijacking

When Ebury is running as an OpenSSH server, it uses LD_PRELOAD to inject its malicious shared module in to programs launched by SSH sessions. Ebury hooks the following functions from `libc` to inject into subprocesses; `system`, `popen`, `execve`, `execvpe`, `execv`, `execvp`, and `execl`.[3][4]

Enterprise T1562 .001 Impair Defenses: Disable or Modify Tools

Ebury can disable SELinux Role-Based Access Control and deactivate PAM modules.[3]

.006 Impair Defenses: Indicator Blocking

Ebury hooks system functions to prevent the user from seeing malicious files (`readdir`, `realpath`, `readlink`, `stat`, `open`, and variants), hide process activity (`ps` and `readdir64`), and socket activity (`open` and `fopen`).[1][4]

.012 Impair Defenses: Disable or Modify Linux Audit System

Ebury disables OpenSSH, system (`systemd`), and audit logs (`/sbin/auditd`) when the backdoor is active.[4]

Enterprise T1556 Modify Authentication Process

Ebury can intercept private keys using a trojanized `ssh-add` function.[1]

.003 Pluggable Authentication Modules

Ebury can deactivate PAM modules to tamper with the sshd configuration.[3]

Enterprise T1027 Obfuscated Files or Information

Ebury has obfuscated its strings with a simple XOR encryption with a static key.[1]

Enterprise T1014 Rootkit

Ebury acts as a user land rootkit using the SSH service.[3][4]

Enterprise T1129 Shared Modules

Ebury is executed through hooking the keyutils.so file used by legitimate versions of `OpenSSH` and `libcurl`.[4]

Enterprise T1553 .002 Subvert Trust Controls: Code Signing

Ebury has installed a self-signed RPM package mimicking the original system package on RPM based systems.[1]

Enterprise T1552 .004 Unsecured Credentials: Private Keys

Ebury has intercepted unencrypted private keys as well as private key pass-phrases.[1]
