Living off the Land | Dell SecureWorks Security and Compliance Blog

Published: 2015-05-28 · Archived: 2026-04-29 02:07:13 UTC

Summary

In over half of the targeted threat response engagements performed by the Dell SecureWorks Counter Threat Unit™ Special Operations (CTU-SO) team in the past year, the threat actors accessed the target environment using compromised credentials and the companies' own virtual private network (VPN) or other remote access solutions. Detecting threat actors who are "living off the land," using credentials, systems, and tools they collect along the way instead of backdoors, can be challenging for organizations that focus their instrumentation and controls primarily on the detection of malware and indicators such as command and control IP addresses, domains, and protocols. With their gaps in visibility, these organizations can have a very difficult time distinguishing adversary activity from that of legitimate users, pushing detection times out to weeks, months, or even years.

Recently, CTU researchers responded to an intrusion perpetrated by Threat Group-1314[1] (TG-1314), one of numerous threat groups that employ the "living off the land" technique to conduct their intrusions. In this case, the threat actors used compromised credentials to log into an Internet-facing Citrix server to gain access to the network. CTU researchers discovered evidence that the threat actors were not only leveraging the company's remote access infrastructure, but were also using the company's endpoint management platform, Altiris, to move laterally through the network (see Figure 1).

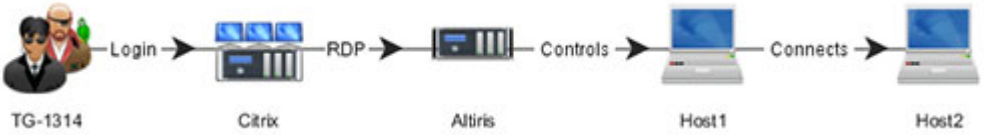

Figure 1. TG-1314 actions on objective. (Source: Dell SecureWorks)

Analysis

Memory collection and analysis can be an extremely valuable component of an incident response plan and in this case proved crucial in identifying TG-1314's actions on objective.

Memory collected from systems involved in the intrusion was analyzed using the Volatility framework. First, Volatility's *pstree* plugin, which lists running processes in a tree view, was executed. The result immediately revealed signs of a suspicious cmd.exe process running as a child of the ACLIENT.EXE process (see Figure 2).

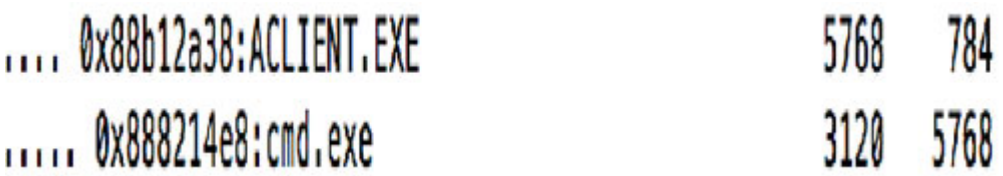

Figure 2. Suspicious cmd.exe process. (Source: Dell SecureWorks)

In an attempt to recover commands that had been executed via this command prompt, Volatility's *cmdscan* plugin was run on the memory dump (see Figure 3).

```
CommandProcess: csrss.exe Pid: 716
CommandHistory: 0x4fa160 Application: cmd.exe Flags: Allocated
CommandCount: 0 LastAdded: -1 LastDisplayed: -1
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0xcf8
Cmd #0 @ 0x4f5c48: cd \recycler
Cmd #1 @ 0x10bca78: ?sexec \\10.65.91.3 -s cmd.exe      ???0?? \\10.101.104.31\c$
Cmd #2 @ 0x4fa760: ?sexec \\10.24.32.14 -s cmd.exe$     ?
```

Figure 3. Suspicious commands recovered from memory. (Source: Dell SecureWorks)

CTU researchers immediately recognized suspicious commands, such as changing the working directory to recycler and executing commands from that location, that were unlikely to have been connected to legitimate system administrator operations. The results also revealed indications that PsExec, a popular system administration tool for executing commands on remote systems, was run against several target hosts to spawn shells on them. To better understand how the adversary was operating and what other actions they had performed, CTU researchers examined cmd.exe and its supporting processes to uncover additional command line artifacts.

While cmd.exe is a console application, it still requires GUI-like functionality and other support to interact with the operating system. On the Windows XP platform, this support is provided by the csrss.exe process. Because commands run from cmd.exe are acted on by csrss.exe, additional evidence of command history and responses sent to the cmd console window are often discoverable by analyzing the csrss.exe process's memory. The output in Figure 3 shows the Process ID (PID) of the csrss.exe process to be 716. Running Volatility's *vaddump* plugin on this process allowed CTU researchers to obtain the Virtual Address Descriptor (VAD) sections (see Figure 4).

```
Pid      Process          Start    End      Result
-------- ---------------- -------- -------- ------
     716 csrss.exe        0x012a0000 0x012a0fff /tmp/csrss.exe.934d830.0x012a0000-0x012a0fff.dmp
     716 csrss.exe        0x01040000 0x0104ffff /tmp/csrss.exe.934d830.0x01040000-0x0104ffff.dmp
     716 csrss.exe        0x00dc0000 0x00dfffff /tmp/csrss.exe.934d830.0x00dc0000-0x00dfffff.dmp
     716 csrss.exe        0x00d00000 0x00d0ffff /tmp/csrss.exe.934d830.0x00d00000-0x00d0ffff.dmp
     716 csrss.exe        0x00830000 0x00835fff /tmp/csrss.exe.934d830.0x00830000-0x00835fff.dmp
     716 csrss.exe        0x00710000 0x0074ffff /tmp/csrss.exe.934d830.0x00710000-0x0074ffff.dmp
     716 csrss.exe        0x00660000 0x0066ffff /tmp/csrss.exe.934d830.0x00660000-0x0066ffff.dmp
     716 csrss.exe        0x00470000 0x004affff /tmp/csrss.exe.934d830.0x00470000-0x004affff.dmp
     716 csrss.exe        0x00270000 0x00285fff /tmp/csrss.exe.934d830.0x00270000-0x00285fff.dmp
     716 csrss.exe        0x00110000 0x00110fff /tmp/csrss.exe.934d830.0x00110000-0x00110fff.dmp
     716 csrss.exe        0x00000000 0x0009ffff /tmp/csrss.exe.934d830.0x00000000-0x0009ffff.dmp
     716 csrss.exe        0x00100000 0x00100fff /tmp/csrss.exe.934d830.0x00100000-0x00100fff.dmp
     716 csrss.exe        0x000d4000 0x000effff /tmp/csrss.exe.934d830.0x000d4000-0x000effff.dmp
     716 csrss.exe        0x000a0000 0x000bffff /tmp/csrss.exe.934d830.0x000a0000-0x000bffff.dmp
     716 csrss.exe        0x000c0000 0x000d3fff /tmp/csrss.exe.934d830.0x000c0000-0x000d3fff.dmp
```

Figure 4. Output from vaddump. (Source: Dell SecureWorks)

The relevant strings inside the VAD sections were UTF-16 encoded and revealed additional insights once extracted. TG-1314 was mapping network drives using a compromised Altiris account to connect to additional systems[2] (see Figure 5).

Figure 5. Net use command. (Source: Dell SecureWorks)

After identifying compromised credentials and executed commands, CTU researchers shifted focus to determine how the threat actors were obtaining the shell and executing their commands on the compromised host. This exploration required a look at the suspect cmd.exe's parent process, shown earlier in the investigation to be ACLIENT.EXE. Volatility's *procdump* command was used to dump the executable from memory (see Figure 6).

```
Process(V) ImageBase  Name                 Result
---------- ---------- -------------------- ------
0x882b2020 0x00400000 ACLIENT.EXE          OK: executable.4212.exe
```

Figure 6. Output from procdump plugin. (Source: Dell SecureWorks)

As shown in Figure 7, running the strings utility against the dumped ACLIENT.EXE binary revealed evidence that the file was the Altiris agent.

```
SOFTWARE\Altiris\Remote Control
Altiris Kernel Driver
SOFTWARE\Altiris\eXpress\NS Client
SOFTWARE\Altiris\eXpress
SOFTWARE\Altiris\Altiris Agent
```

Figure 7. Output from strings plugin. (Source: Dell SecureWorks)

These results indicated that the threat actors leveraged the Altiris management platform installed at the client site, along with compromised domain credentials associated with the Altiris system, to move laterally within the compromised environment.

Conclusion

Threat groups often follow a path of least resistance to achieve their objective. They will leverage legitimate remote access solutions for entry and valid system administrator tools for lateral movement, if possible. To help disrupt this tactic, it is important that organizations implement two-factor authentication for all remote access solutions and consider doing the same for internal, high-value assets like their internal system management consoles. CTU researchers assess with high confidence that threat groups like TG-1314 will continue to live off of the land to avoid detection and conduct their operations.

[1] The Dell SecureWorks Counter Threat Unit™ (CTU) research team tracks threat groups by assigning them four-digit randomized numbers (1314 in this case), and compiles information from external sources and from first-hand incident response observations.

[2] One limitation of collecting strings from the VAD of the csrss.exe process is that there is no temporal information.

Source: https://web.archive.org/web/20150626073312/http://www.secureworks.com/resources/blog/living-off-the-land/