Return of the Darkside: Analysis of a Large-Scale Data Theft Campaign

By Snir Ben Shimol

Published: 2021-03-18 · Archived: 2026-04-05 18:41:11 UTC

Our team has recently led several high-profile investigations of attacks attributed to an up-and-coming cybercrime group, Darkside. These highly targeted campaigns were conducted in several phases over weeks or months, ultimately targeting theft and encryption of sensitive data, including backups. In this technical blog post, we will review the tactics, techniques, and procedures (TTPs) we've observed.

About Darkside, inc.

The Darkside ransomware group announced their RaaS (Ransomware-as-a-Service) in August of 2020 via a "press release." Since then, they have become known for their professional operations and large ransoms. They provide web chat support to victims, build intricate data leak storage systems with redundancy, and perform financial analysis of victims prior to attacking.

The group's name, Darkside, evokes the image of a good guy (or gal) that has turned from the light. While we can't conclude that the group is comprised of former IT security professionals, their attacks reveal a deep knowledge of their victims' infrastructure, security technologies, and weaknesses.

They have publicly stated that they prefer not to attack hospitals, schools, non-profits, and governments, but rather big organizations that can afford to pay large ransoms.

Our reverse engineering revealed that Darkside's malware will check device language settings to ensure they don't attack Russia-based organizations. They have also answered questions on Q&A forums in Russian and are actively recruiting Russian-speaking partners.

The group has both Windows and Linux toolsets. Much like NetWalker and REvil, Darkside has an affiliate program that offers anyone who helps spread their malware 10-25% of the payout.

Anatomy of an Attack

The Darkside ransomware attack campaigns stood out for their use of stealthy techniques, especially in the early stages. The group performed careful reconnaissance and took steps to ensure that their attack tools and techniques would evade detection on monitored devices and endpoints.

While their initial entry vectors vary, their techniques are more standardized once inside, and their endgame is coldly efficient.

Stealth tactics include:

- Command and control over TOR
- Avoiding nodes where EDR is running
- Waiting periods & saving noisier actions for later stages
- Customized code and connection hosts for each victim
- Obfuscation techniques like encoding and dynamic library loading
- Anti-forensics techniques like deleting log files

During the later stages of their attack sequence, they:

- Harvest credentials stored in files, in memory, and on domain controllers
- Utilize file shares to distribute attack tools and store file archives
- Relax permissions on file shares for easy harvesting
- Delete backups, including shadow copies
- Deploy customized ransomware

Initial Access: Finding the Weak Link

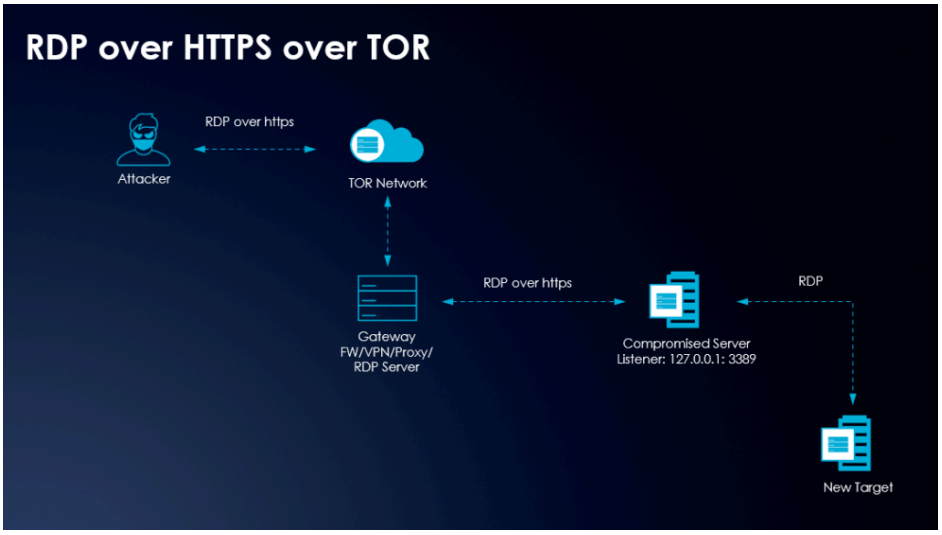
Darkside ransomware gained initial entry through weak links – remotely exploitable accounts and systems.

We observed Darkside use compromised contractor accounts to access Virtual Desktop Infrastructure (VDI) that had been put in place to facilitate remote access during the pandemic. Though, contractor accounts did not.

We also observed them exploit servers, and then quickly deploy an additional RDP that would preserve access should the vulnerable server be patched.

While neither of these vectors is novel, they should serve as a warning that sophisticated threat actors are easily bypassing perimeter defenses. They illustrate the need for multi-factor authentication on all internet-facing accounts and rapid patching of internet-facing systems.

Command and Control



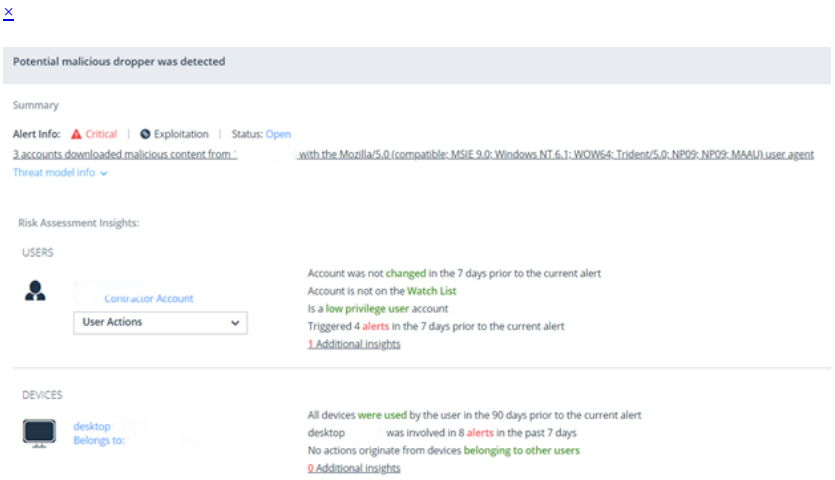
The Darkside ransomware attackers established command and control primarily with an RDP client running over port 443, routed through TOR. After installing a Tor browser, they modified its configuration to run as a persistent service, redirecting traffic sent to a local (dynamic) port through TOR via HTTPS over port 443, so it would be indistinguishable from normal web traffic. These connections were persistent, so the attackers could establish RDP sessions to and through the compromised hosts, facilitating lateral movement.

We found traces of TOR clients across many servers and observed dozens of active TOR connections.

The attackers used Cobalt Strike as a secondary command and control mechanism. We observed dozens of customized stagers that downloaded customized beacons that connected to specific servers. The stagers (named file.exe) were deployed remotely on specific targeted devices using WinRM, each one configured differently. Cobalt-Strike stagers established connections to a dedicated C2 server to download the Cobalt Strike Beacon.

Threat actors commonly use only a few C2 servers per victim, but Darkside configured each beacon to connect to a different C2 server with a different user agent. This would indicate that Darkside operates a large, well-established attack infrastructure.

The stagers and TOR executables were stored in network shares for easy distribution. The actors avoided installing backdoors on systems monitored by EDR solutions.



Detection of the beacon being downloaded into a compromised server

We observed the threat actors log into the Virtual Desktop environment with many accounts, sometimes concurrently. Each time the threat actor logged on, .lnk files were created in the compromised user's home folders. The .lnk file activity helped determine which accounts and VDI environments had been compromised and when each account was used in the attack.

Recon and Credential Harvesting

Darkside ransomware is known for living off the land (LOtL), but we observed them to scan networks, run commands, dump processes, and steal credentials. Like the command and control code, the attack tools were also executed on hosts that had minimal detection and blocking capabilities. Well-known tools included **advanced_ip_scanner.exe, psexec, Mimikatz, and more**.

From the initial set of compromised hosts, ticket requests, and NTLM connections to gain access to additional systems and accounts. After a waiting period, the actor used an Active Directory reconnaissance tool (ADRecon.ps1) to gather additional information about users, groups, and privilege, storing results in a file called, DC.txt. Each of their attack tools was deleted after use. The attacker temporarily stored the recon results and credential information on a very active windows server. Interesting file names written and deleted on the server included: Typed_history.zip, Appdata.zip, IE_Passwords.zip, AD_intel, and ProcessExplorer.zip.

In addition to credential harvesting, the attacker mined credentials from User profile folders, including:

- Users\<user name>\Appdata\[Roaming\Local]\Microsoft [Credentials\Vault]
- Users\<user name>\Appdata\Roaming\Mozilla\Firefox\Profiles
- Users\<user name>\\Appdata\Local\Google\Chrome

The threat actor used Invoke-mimikatXz.ps1 to extract credentials from unmonitored servers and stored them in a file called “dump.txt.” This operation was performed on a high-value target with minimal detective capabilities.

Once the attacker obtained domain admin credentials, accessed domain controllers. In later stages they performed the well-known DCSync attack, where the attacker pretends to be a legitimate domain controller and utilizes the Directory Replication Service to replicate AD information, gaining access to password data for the entire domain, including the KRBTGT HASH.

Data Collection and Staging

The active Windows server also served as a hub to store data before exfiltration. Data was mined from hundreds of servers with a batch routine (dump.bat) located in \Desktop\Dump, writing files to the same location, compressing them into 7zip archives with a simple naming convention, *.7z.[001]-[999].

Though they had accumulated elevated privileges, we observed the attacker relax the permissions on file systems, opening them up so that they could access the files with any domain user account. The batch file, target data, and the archives were deleted by the attackers within hours of collection

Encryption

Darkside doesn’t deploy ransomware until they’ve mapped the environment, exfiltrated interesting data, gained control of privileged accounts, and identified all backup systems, servers, and applications. We observed several connections to primary backup repositories using compromised services accounts shortly before encryption. By holding off on the encryption phase of the attack, they put themselves in a position to maximize damage and profit.

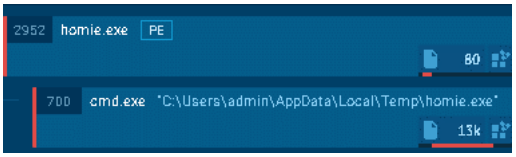
The ransomware code is delivered through established backdoors (TOR-RDP or Cobalt Strike) and is customized for each victim. The payload includes the executable, a unique extension, and a unique victim ID that allows the victim to access Darkside’s website and make payment.

By using unique executables and extensions, the ransomware easily evades signature-based detection mechanisms. Darkside also provides customized ransomware to other threat actors (Ransomware as a Service) and takes a part of the profit in successful attacks.

One version of the customized code was named, “Homie.exe.” In addition to being customized, we found it also uses anti-forensics and anti-debugging techniques, such as self-injection, virtual machine detection, and dynamic library loading. It also deletes shadow copies on victim devices.

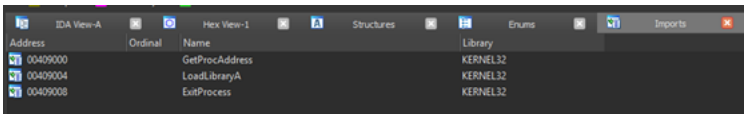
Darkside Ransomware Stage 1 – Self Injection

On execution, the malware copies itself to the path “C:\Users\admin\AppData\Local\Temp\” and injects its code into the existing process with a CMD command:



If the malware finds indications that it is being debugged or run in a VM, it immediately stops.

To avoid detection by AV and EDR solutions, the ransomware dynamically loads its libraries, without registering them in its imports section:



Only 3 libraries are imported, which indicates that other libraries' names resolved dynamically during the malware's run instead of being explicitly imported.

Ransomware Stage 2 – Deletion of Shadow Copies

Using an obfuscated PowerShell command, the malware attempts to delete the shadow copies on the victim device. The obfuscated command:

```
powershell -ep bypass -c "(0..61)|%{$s+=[char][byte]
('0x'+'4765742D576D694F626A65637420576965E33325F536861646F77636F7079207C20466F72456163682D4
F626A656374207B245F2E44656C657465728293B7D20'.Substring(2*$_,2))};iex $s"
```

The de-obfuscated command:

```
PS C:\Windows\system32> (0..61)|%{$s+=[char][byte]('0x'+'4765742D576D694F626A65637420576965E33325F536861646F77636F7079207
C20466F72456163682D4F626A656374207B245F2E44656C657465728293B7D20'.Substring(2*$_,2))};
PS C:\Windows\system32> $s
Get-WmiObject Win32_Shadowcopy | ForEach-Object {$_.Delete();}
```

Ransomware Stage 3 – Encryption of Files

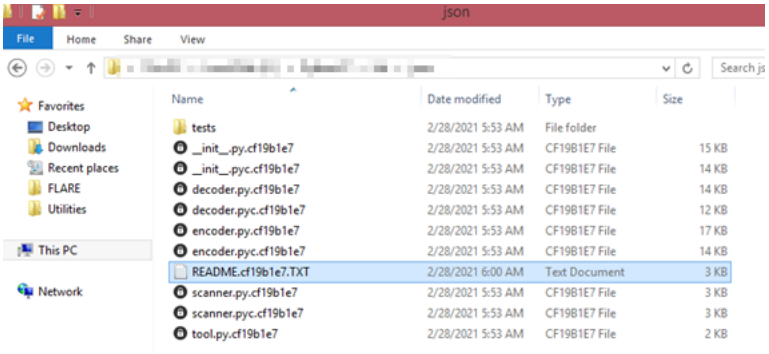
After the deletion of the shadow copies, the malware first closes specific processes to avoid locked files that can delay encryption, and then begins its encryption routine.

List of processes:

- sql
- oracle
- ocssd
- dbsnmp
- synctime
- agntsvc
- isqlplussvc
- xfssvccon
- mydesktopservice
- ocautoupds
- encsvc
- firefox
- tbirdconfig
- mydesktopqos
- ocomm
- dbeng50
- sqbcoreservice
- excel
- infopath
- msaccess
- mspub
- onenote
- outlook
- powerpnt
- steam
- thebat
- thunderbird

- visio
- winword
- wordpad
- notepad

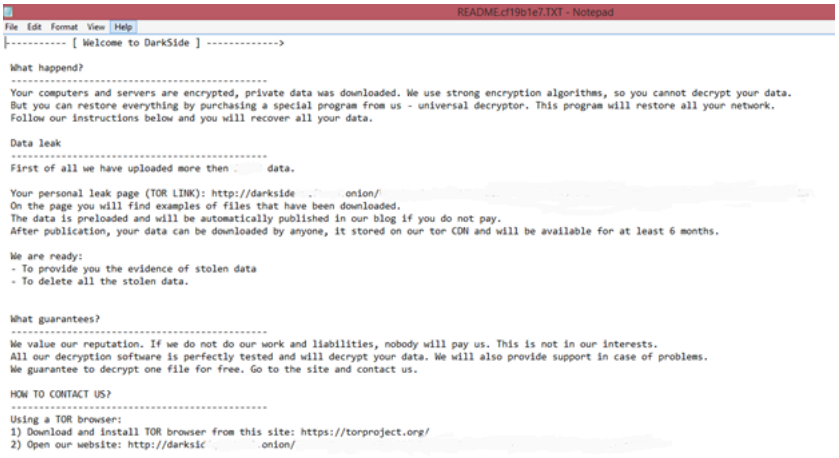
During encryption, the malware appends an 8-character string to the end of the encrypted file names.



- Dark side ransomware avoids encrypting files with the following extensions:

386,adv,ani,bat,bin,cab,cmd,com,cpl,cur,deskthemepack,diagcab,diagcfg,diagpkg,dll,drv,exe,hlp,icl,icns,ico,ics,idx,ldf,lnk,mod,mpa,msc,msp,msstyles,ms

- It creates a ransom instructions (“README…txt”) to contact the ransomware creator for decryption:



How to Prepare for Threat Actors in 2021

Find and fix the weak links before attackers do

Any internet-facing account that doesn’t require MFA is a brute-force attack away from a compromise. Any unpatched internet-facing server is an exploit away from script-kiddie payday.

Assume breach and fix weak links inside

Threat actors look for quick ways to obtain domain admin credentials. Service or admin accounts with SPNs that also have weak encryption, or worse still, privileged accounts with weak or no password requirements are too-easy targets.

In too many organizations, attackers don’t even need elevated credentials to harvest data – the average employee has access to far more data than they require. Lockdown sensitive data so that only the right accounts have access, and then monitor file systems for unusual access and change events.

More lights, please, especially on stuff that matters

Organizations with comprehensive monitoring solutions detect and investigate attacks like these more quickly. If you have blind spots on core data stores, in Active Directory, DNS, remote access systems, or in web connections, you’ll struggle to determine which systems were compromised and whether sensitive data was stolen.

If you detect a breach, let Active Directory triangulate the blast radius

Active Directory events can help you quickly identify compromised accounts and devices. Instead of focusing on one endpoint at a time, once one compromised account or system has been identified, query Active Directory for signs of lateral movement by that account or accounts used on that system.

If you wait for a breach to occur, it's too late. Strengthen your cloud security today and stay ahead of emerging threats with Varonis. Learn more about our comprehensive cloud security solutions and take advantage of our free Data Risk Assessment to help you safeguard your digital assets.

A special thanks to Rotem Tzadok for leading our Darkside investigations and analysis.
