White Paper

Bitdefender®

RDP Abuse and Swiss Army Knife Tool Used to Pillage, Encrypt and Manipulate Data

Contents

Authors:

Eduard BUDACA - Security Researcher

Victor VRABIE - Security Researcher

Co-Authors:

Cristina VATAMANU - Senior Team Lead, Cyber Threat Intelligence Lab

Alexandru MAXIMCIUC - Team Lead, Cyber Threat Intelligence Lab

Executive Summary

Bitdefender researchers recently found threat actors abusing a legitimate feature in the RDP service to act as a fileless attack technique, dropping a multi-purpose off-the-shelf tool for device fingerprinting and for planting malware payloads ranging from ransomware and cryptocurrency miners to information and clipboard stealers.

The attack vector involves the Windows Remote Desktop Server. The RDP client has the ability to share a drive letter on their machine, which acts as a resource on the local virtual network. Attackers were able to use the shared directory as a very simple data exfiltration mechanism over the RDP protocol. By using an off-the-shelf component placed on the "tsclient[1]" (Terminal Server Client) network location, attackers could execute it using either "explorer.exe" or "cmd.exe" and use it to download additional malware.

The "worker.exe" component provides a vast array of capabilities, mainly for data gathering. It features capabilities ranging from collecting system information (e.g. architecture, CPU model and core count, RAM size, Windows version etc.) to taking screenshots, collecting the victim's IP address and domain name, pulling information about default browsers and specific open ports, and even anti-forensic and detection evasion commands.

The tool has been found in various campaigns – analyzed below in this paper - potentially associated with one or more cybercriminals or cybercriminal groups, delivering at least three distinct generations of clipboard stealers, ransomware (Rapid Ransomware and Nemty ransomware), Monero cryptocurrency miners, and a highly popular Trojan stealer (AZORult).

The campaigns do not seem to target specific industries or companies; instead, threat actors have used a shotgun approach, focusing on reaching as many victims as possible. In terms of financial impact, estimated cryptocurrency earnings based on the cryptocurrency wallets found indicate attackers have netted at least $150,000 through some of their campaigns.

Key Findings

- RDP abuse to exfiltrate data through network shares
- Off-the-shelf multi-purpose tool used to screen victims and drop malicious payloads (ransomware, clipboard stealers, cryptocurrency miners and info-stealer Trojans)
- Ready-made ransomware families used as payload (Rapid Ransomware and Nemty)
- Clipboard stealers replace cryptocurrency addresses with one that belongs to attackers
- More than $150,000 in cryptocurrency earnings (22.604 BTC, 25.098 ETH, 13.846 DASH and 1.329 LTC), excluding Monero.

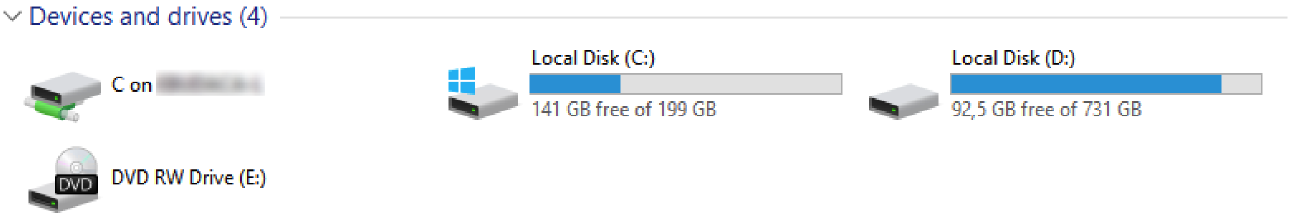
[1] **Wikipedia, "tsclient", https://en.wikipedia.org/wiki/Tsclient**

Infection Vector

The infection mechanism is an interesting one, as the attackers used a technique considered fileless from a disk forensics standpoint. They used a feature of the Windows Remote Desktop Server in which the RDP client can share a drive letter on their machine, which appears on the server as a share on a virtual network location named “tsclient”. For example, if the client shares their C: drive (Fig. 1), the files inside will be available in a directory at \\tsclient\c (the single letter directory name is the drive letter shared). These shares are both readable and writable, making for a very simple data exfiltration mechanism over the RDP protocol, a common use case for most threat actors.

Additionally, since Remote Desktop is a built-in feature of most Windows installations (except for limited “Home” Windows versions), its file-sharing functionality is built in as well, both contributing to its appeal as an infection vector, while being encrypted by default and evading behavior detection.

Interestingly, the limitation of sharing just drive letters is only present in the Remote Desktop client, and not the server. Custom RDP clients can share any directory, which will appear as a share on the “tsclient” virtual network location that has a name not limited to a single letter. From what we have seen in our telemetry, these attackers prefer using these kinds of paths, with multiple-letter directory names.



Devices and drives (4)

C on [redacted]

Local Disk (C:)
141 GB free of 199 GB

Local Disk (D:)
92,5 GB free of 731 GB

DVD RW Drive (E:)

Fig 1. Example of a drive shared over RDP, as seen in this PC under Windows 10 (named [drive letter] on [client machine name])

Analyzing our telemetry, we saw that the main malicious component of this attack, named ***worker.exe***, was located on what appears to be a network share on the “tsclient” network location and was executed by either explorer.exe or cmd.exe. That allows us to say with a high degree of confidence that this attacker controls the victims’ machines using the Remote Desktop Protocol. However, this usually requires the attackers to possess valid credentials to access these machines, and we can only speculate as to how the credentials were acquired. Existing cyber security literature seems to link both worker.exe infections and some exploitation components used by this attacker (for example, Rapid Ransomware) with RDP bruteforcing, but we have found no evidence of this.

It is important to note that the use of worker.exe does not necessarily indicate an attack by this specific threat actor, as others have used it as well in campaigns distributing SamSam and Dharma ransomware, according to security researchers.

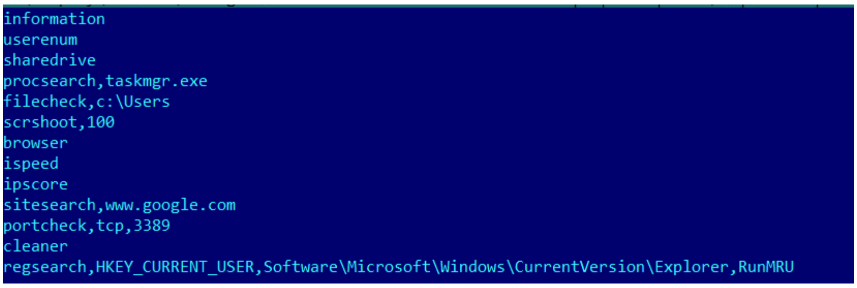
Worker.exe Component – A Swiss Army Knife for Threat Actors

This component is usually the first executed once a machine is infected. It is located in a “tsclient” share next to a text file named “config.ins” (Fig. 2). These samples do not use C&C servers; instead, communication with the threat actors occurs through the config.ins file, from which a list of commands is read and executed. The output of these commands is written to a .nfo file (Fig. 3), with a filename taken from the first command line argument (in our telemetry, this was the victim’s IP address, used as an identifier). In addition, commands can write files with a name derived from the same identifier. At the end of execution, this module, worker.exe, writes an empty file, its name being the same identifier without extensions, to signal that it has finished processing the commands and writing the output. These files are written next to the worker.exe sample. This is very convenient, as writing files to the tsclient share is the precise data exfiltration mechanism of choice for this attack, which makes it fileless (from a forensic point of view) as well. This might indicate that the worker.exe component was purposely built for Remote Desktop attacks.

This component provides a vast array of commands that can be contained in the config.ins file, mainly for data gathering, but also providing one command to execute batch or JavaScript files. These commands are in plain text, one per line, with arguments separated by commas. Information that worker.exe can gather includes:

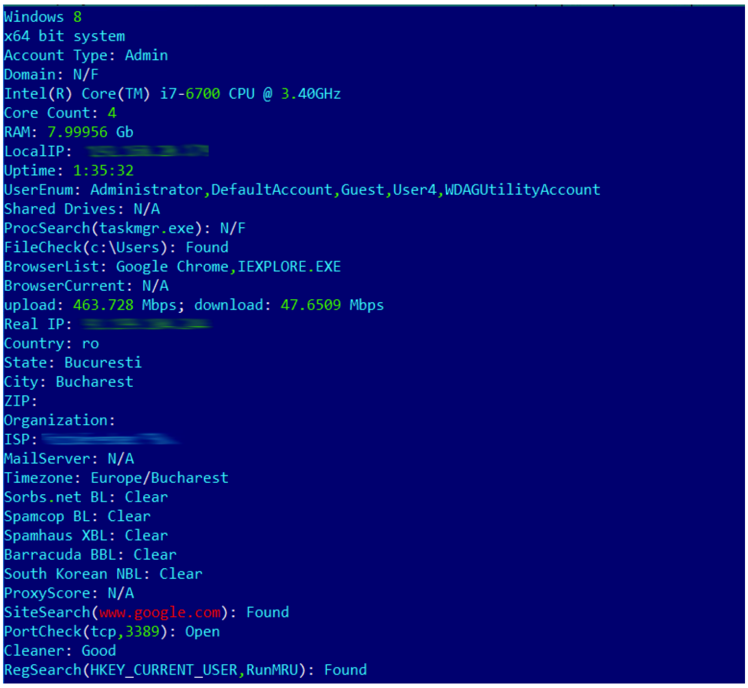
- System information: architecture, CPU model and core count, RAM size, Windows version
- Domain name, whether the user is a local administrator or not, a list of all users on the machine
- Local IP address, upload and download speed, public IP information as returned by the http://www.ip-score.com/ service
- Default browser, whether specific ports are open on this machine (whether a server is running, listening on that port), whether specific entries exist in the machine's DNS cache (whether the machine has recently tried to connect to a specific domain, regardless of whether it was found or not)
- Screenshots
- Whether a certain process is running, whether specific keys and values in the registry exist, whether specific files or directories exist, listing all remote drives (connected network shares with a drive letter assigned)

A final command has an anti-forensic and detection-evasion purpose, allowing the attackers to clear the MRUList entry in the `HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU` registry key, which contains the Run dialog's history. This indicates that the attackers may use this dialog during Remote Desktop sessions and wish to hide the executed commands.



```
information
userenum
sharedrive
procsearch,taskmgr.exe
filecheck,c:\Users
scrshoot,100
browser
ispeed
ipscore
sitesearch,www.google.com
portcheck,tcp,3389
cleaner
regsearch,HKEY_CURRENT_USER,Software\Microsoft\Windows\CurrentVersion\Explorer,RunMRU
```

Fig 2. An example of a valid config.ins file (based on reverse engineering process)



```
Windows 8
x64 bit system
Account Type: Admin
Domain: N/F
Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz
Core Count: 4
RAM: 7.99956 Gb
LocalIP: [redacted]
Uptime: 1:35:32
UserEnum: Administrator,DefaultAccount,Guest,User4,WDAGUtilityAccount
Shared Drives: N/A
ProcSearch(taskmgr.exe): N/F
FileCheck(c:\Users): Found
BrowserList: Google Chrome,IEXPLORE.EXE
BrowserCurrent: N/A
upload: 463.728 Mbps; download: 47.6509 Mbps
Real IP: [redacted]
Country: ro
State: Bucuresti
City: Bucharest
ZIP:
Organization:
ISP: [redacted]
MailServer: N/A
Timezone: Europe/Bucharest
Sorbs.net BL: Clear
Spamcop BL: Clear
Spamhaus XBL: Clear
Barracuda BBL: Clear
South Korean NBL: Clear
ProxyScore: N/A
SiteSearch(www.google.com): Found
PortCheck(tcp,3389): Open
Cleaner: Good
RegSearch(HKEY_CURRENT_USER,RunMRU): Found
```

Fig 3. A *.nfo output file for the config.ins input above

Some of the commands supported by this component are specific to a fingerprinting operation. Screening the victims is useful for the threat actor to decide what further actions to take.

For instance, if the victim belongs to a certain domain, the attacker can conclude that the infected system is a part of an enterprise network. This means that delivering a ransom payload would benefit them more. However, if the account has admin privileges, they can perform certain actions such as privilege escalation, which can lead to service installation, task creation and so on.

Since this component is an off-the-shelf tool, an actor can choose which functionalities from the whole package to actually use in their attack.

Our telemetry shows that the worker.exe component has been used since February 2018. From the beginning, this tool was employed, presumably after some data gathering, to run at least three types of malicious programs: Monero miners, ransomware, and clipboard stealers (replacing cryptocurrency addresses in the Windows clipboard with one of the attacker's). Starting in July of 2018, we have also seen the AZORult information stealer deployed by the worker.exe component.

Almost all files we saw during this investigation, both worker.exe and additional components, used the same packer. Unpacking them reveals many repeated payloads, which indicates that the attackers sometimes re-packed payloads to escape detection. In some cases, a component had two or more layers of packing applied.

Clipboard stealer payloads

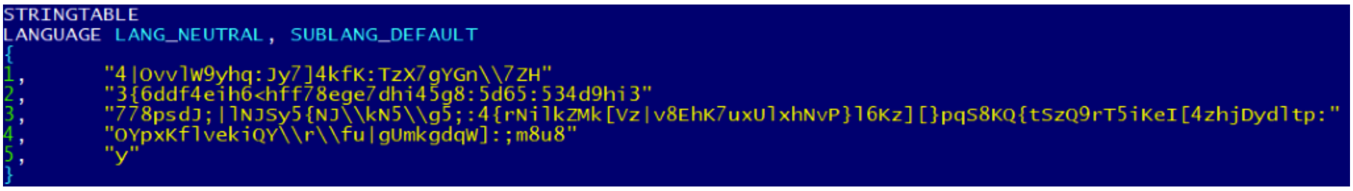
The purpose of these payloads is to exploit the way in which cryptocurrency payments are made, in order to divert funds to the attackers. Since the payment addresses for these cryptocurrency schemes are difficult to type, they are copied to the relevant fields in transaction forms through the clipboard. A malicious program can monitor the clipboard for strings of text that resemble a cryptocurrency address, and, on a match, replace that string with another address that belongs to the attacker.

Three distinct generations of clipboard stealers have been identified:

MicroClip

This component seems to have been written in the C++ programming language. It can identify and replace Bitcoin, Ethereum, Monero and Litecoin addresses in the clipboard, using regular expressions.

The addresses stored in place are contained in a stringtable resource, encrypted with a SUB 3 operation, one address per cryptocurrency scheme (Fig. 4). It achieves persistence only if a fifth entry in the stringtable has the value "y". If so, it copies itself to the user's Application Data directory, in a directory with a name derived from the serial number of the volume where Windows is installed, as "svchost.exe", adds that file to the user's Run key, with the same name as the directory mentioned above, executes its copy, and shuts down. As a measure against one of Windows' threat protection systems, this component deletes the "Zone.Identifier" alternate data stream of the "svchost.exe" copy before executing it.



```
STRINGTABLE
LANGUAGE LANG_NEUTRAL, SUBLANG_DEFAULT
{
1,      "4|Ovvlw9yhq:Jy7]4kfK:TzX7gYGn\\7ZH"
2,      "3{6ddf4eih6<hff78ege7dhi45g8:5d65:534d9hi3"
3,      "778psdJ;|lNJSy5{NJ\\kN5\\g5;:4{rNilkZMk[Vz|v8EhK7uxUlxhNvP}16Kz][]}pqS8KQ{tSzQ9rT5iKeI[4zhjDydltp:"
4,      "OYpxKflvekiQY\\r\\fu|gUmkgdqW]:;m8u8"
5,      "y"
}
```

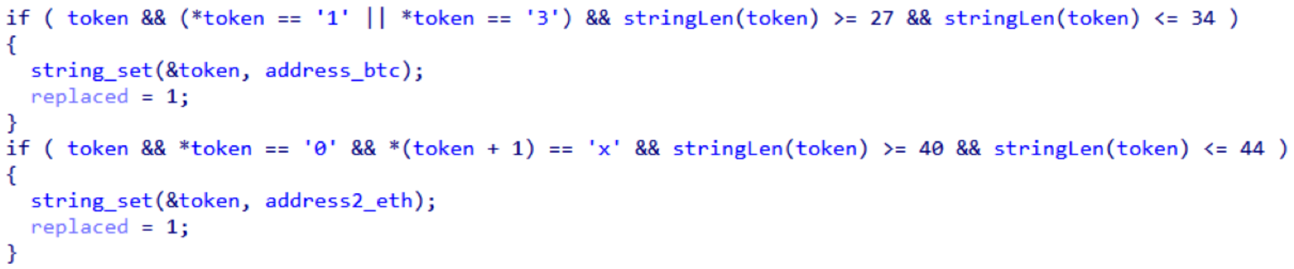
Fig 4. An example of a MicroClip stringtable resource

The name comes from two in-the-wild filenames, "MicroClipV.exe" and "MicroClipV3.exe". Most filenames, however, read like keyboard mashing and were presumably intended to be understood only by the attackers (such as "cllipp998565.exe", "05gold445.exe", "taxi699.exe"). This component was actively spread by worker.exe between June and October 2018.

During our investigation, we were able to find a sample of the builder for this generation of clipboard stealers. It simply reads four addresses (in order: Bitcoin, Monero, Ether and Litecoin) from standard input, encrypts them, and produces two MicroClip binaries, one named “build_yes_startup.exe” and one named “build_no_startup.exe”. Then, the builder configures them by updating their stringtable resource with the supplied addresses, as well as the value “y” or “n”, depending on whether the MicroClip build is configured to install persistence or not. Finally, the builder contains the e-mail address of its developer, and it can itself be found for sale on dedicated forums, which makes MicroClip an off-the-shelf tool.

DelphiStealer

This generation of clipboard stealers is written in the Delphi programming language. It does not use regular expressions, instead manually splitting the keyboard contents by spaces and checking the length of split tokens and a scheme-specific prefix. It only supports Bitcoin (prefix “1” or “3”) or Ethereum (prefix “0x”) stealing (Fig. 5). The replacement addresses, one for each scheme, are hardcoded strings in the binary, encrypted with a SUB 0x6A operation. If a third fill-in-the-blanks string in the binary starts with a “C” character (as was the case in around two-thirds of the samples we saw), this component installs its persistence: the binary sets itself to the current user’s Run key, named “12365435”.



```
if ( token && (*token == '1' || *token == '3') && stringLen(token) >= 27 && stringLen(token) <= 34 )
{
  string_set(&token, address_btc);
  replaced = 1;
}
if ( token && *token == '0' && *(token + 1) == 'x' && stringLen(token) >= 40 && stringLen(token) <= 44 )
{
  string_set(&token, address2_eth);
  replaced = 1;
}
```

Fig 5. DelphiStealer’s heuristics for detecting cryptocurrency addresses

In our telemetry, the filenames the attackers used for this component were variations of “winlogon.exe” (such as “winlodin.exe”, “winlogon64.exe”, “winlogin.exe”) in the Application Data directory, which is a form of masquerading (ATT&CK Technique T1036). This component was actively spread by worker.exe between November 2018 and May 2019, with a brief resurgence in September 2019.

In some respects, this generation is a downgrade from MicroClip, which supported Monero and Litecoin in addition to Bitcoin and Ethereum and used a more robust method of identifying target addresses in the clipboard (the Delphi component can only find addresses separated by spaces). This change may have meant to evade detection, at the cost of lower earnings.

IntelRapid

This third and final generation of cryptocurrency stealers seems to be written again in the C++ programming language but appears to have a different code-writing style from the previously mentioned stealers. Some details are much more carefully designed, leading us to believe that this component has a different author than the previous two. The code is also heavily obfuscated using numerous layers of virtual calls, which makes reverse engineering difficult. This generation supports a much wider range of cryptocurrency schemes: Bitcoin, Litecoin, Ethereum, Monero, Bitcoin Cash, Dash, Ripple, Dogecoin, Neo, and ZCash. The encryption and storage of addresses has been overhauled: each scheme has a PE resource of a custom type (“DATABASE”), encrypted with a hardcoded translation table. However, this makes the resources easy to decrypt, since the translation table consists of 256 distinct, continuous bytes in the binary, which are easy to find.

This component creates a tray window (named “IntelRapidTray”) to be able to intercept the clipboard. When the user copies some text, IntelRapid splits it on a predefined list of separators (all kinds of brackets, whitespace, colon, or equals characters) then tries to match each token to a cryptocurrency scheme, using a set of rules based on token length, character sets and prefixes.

On a match, this component tries to find the best of many addresses in that scheme (sometimes exceeding 1,300, as is the case for a sample’s Bitcoin set) as a replacement. It does this by using a complex scoring mechanism that looks for similarities between the original address and the new one, such as a common prefix, a common suffix or similar-looking characters in close positions.

To show an example of this algorithm's output, we created Table 1 based on the replacements of one of the IntelRapid samples.

Original address	Coin	Replaced address	Matching rule
3Cd4t2kyPBt4LNk7YXJSQBHriH2v3rXN2E	BTC	**3Cd4**vNgjWvukvwgxJ7C8n8mKoBiFoXXucq	Prefix
113TsCxFV9JejtMNfdvVpatQSfvMAn**3HFZ**	BTC	3GbzsfYZ5vkY44uGyr4EpVjfpiJJH9**3HFZ**	Suffix
1256t2kyPBt4LXk7YXJSQkHriH2v3rXN**2E**	BTC	3JQkgwbKnNDBzcvDpt8W6bxKioqNXQQh**2E**	Suffix
0x58F1812D85E8396E48592A0E175CEf8a8e89Bf5C	ETH	**0x5**0d9B28510D2Eb442b1716067f9ebBAb1732A055	Prefix
0x22A4414492071173Dcd9163701a8Edd3Ae673D2A	ETH	**0x2**FA984ed7762942a34695Cc0B1713670D3A9cB57	Prefix
0x99b500f473b321ea892617f6b3e9c0880078adaf	ETH	**0x9**ab12bf9c341c6ba690c8416ceb4f2f8f92afb6f	Prefix
4Ad8uTQvA7ihvoJv3H7AccUSYpjE9743C5RsjCjsn9cXdKTBt4YkRL4Wrvo5BZ57f3jRjsoGZ8QN59NfjQJrmoWA5HiCqGe	XMR	**4A**xuhPR6PBJeZcmfxyxyy8SNkSHbF72qiJCQSBxjaf9gDk9idv6bQv5ifJkhM98qPSKGmgJnbfrGXCYHvmF2cBX8693EEQ1	Prefix
48zNHjHYcBq3tL5mFTouT4W85gqZVoCXTNheWFNPrvmv7ThdV9qb115fWs3WrKjQ62WRN2ekWcsYAc8TpmmKC1Ag7Sn9USb	XMR	**48**tdP3WtHzW5KL3C36iUHXNbQ6CoLYUfyGHyz4eH39BeTof135GXmQDf1q1g5UwtuWPeRWPb5pZmLLTkyQ1saz1BArFZtKB	Prefix
48xKLsuYBUYbAEwyLKfY5cBWSmJ3L72PUH5YaGSZQbKmLVYmKdHYBvxGW2kFQBVHMrFgN4MPVqPUoRizWwAWJSv5VRCvrWG	XMR	**48**tdP3WtHzW5KL3C36iUHXNbQ6CoLYUfyGHyz4eH39BeTof135GXmQDf1q1g5UwtuWPeRWPb5pZmLLTkyQ1saz1BArFZtKB	Prefix

Table 1. Example of IntelRapid address replacement algorithm.
Note how different addresses in the same cryptocurrency scheme are replaced with different addresses, depending on matching characters.

This generation achieves persistence by copying itself to "Application Data\Roaming\Intel Rapid\IntelRapid.exe" in the current user's directory and adding itself both as an "Intel Rapid" scheduled task and as a "IntelRapid.lnk" shortcut in the user's Startup directory of the Start menu. After this component installs itself (on the first run), it runs the new copy and exits.

IntelRapid uses a "{37529D08-A67E-40B3-B0F2-EB87331B47F5}" mutex to ensure only one instance of itself is running. Interestingly, the cryptocurrency stealing routines will not run if a Russian or Ukrainian keyboard is installed on the machine, which seems to be common practice for threats developed by Russian-speaking actors.

Throughout the infection campaign, this component has repeatedly changed its filenames and other names used for persistence. "IntelRapid.exe" is not the first filename chronologically, but it is the first recognizable identity that is has assumed. Other filenames used include (in chronological order) "wmphostk.exe", "wmpl.exe", "windowsmediaplayer32.exe", "IntelRapid32.exe", "WindowsLogonApplication64.exe", and more recently (since mid-August 2019) "RealtekDSb.exe", "RealSoundHelp.exe" and "UtilityService.exe". Other user-visible names (directories, scheduled task names) are changed accordingly, for example to "Windows Logon Application 64", "Realtek Sound Blaster Driver", "Realtek Sound Help" or "Utility Service".

This component was actively spread in April and May 2019, with a small number of infections after August the same year.

Attributing off the shelf tools to the same actor

One important step in our investigation was to expand our sample set and to find similar files for the tools spotted in the first stage of the investigation. Since we saw that these attackers use off-the-shelf tools, like the worker.exe component, we wanted to make sure that we consider only similar files that are linked to this threat actor in our understanding of the campaigns.

A near-certain indicator of attribution for cryptocurrency stealers is the set of replacement addresses, since it would make little sense for an attacker to direct stolen cryptocurrency to an account different than their own.

To do that, we gathered as many samples as we could of MicroClip, the Delphi stealer, and IntelRapid, then extracted their addresses and clustered them: if two samples share an address, they belong to the same cluster. This produced interesting results: Most MicroClip and all Delphi stealer samples ended up in the same cluster ("the main cluster" from now on), along with several IntelRapid binaries.

When trying to establish a connection in our telemetry between these samples and the main component, worker.exe, most of those in the main cluster were seen shortly after - or even executed by - a "worker.exe" sample. None of the samples belonging to other clusters had any such connection. So, connecting the worker.exe component to the spreading of these cryptocurrency stealers is enforced by these observations.

After we produced a set of samples known to be linked to the same attacker, we could connect other malicious components to it: the ransomware, Monero miner and AZORult attacks linked with worker.exe often happened very soon before or after a cryptocurrency stealer sample was deployed. Miners are even more connected; the mining of Monero occurs for many of the wallets found in the main cluster.

Cryptocurrency stealer campaign earnings

After building a set of samples in the main cluster, we gathered their replacement addresses and scanned the ledgers for transactions involving them. Most addresses never received any cryptocurrency, and many, if not all, were routinely emptied of all funds and had a balance of zero. Out of the ten cryptocurrency schemes, only four had received funds (and Monero is by design untraceable): as of the time of writing, (rounded to three decimal places) 22.604 BTC, 25.098 ETH, 13.846 DASH and 1.329 LTC. To estimate a dollar value for this currency, we used the value of these currencies on the median day of infection in our telemetry: the 29th of November 2018. This would put the earnings around $152,500. (Note that Monero could not be included in this estimate.)

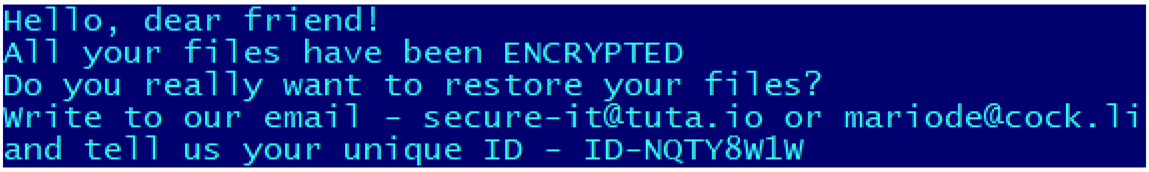
Ransomware payloads

Starting around April 2018 (according to our telemetry), these attackers started deploying ransomware to their victims. We've seen two families being used: Rapid Ransomware and Nemty, both of which seem to be off-the-shelf tools for threat actors. Most of these samples we saw are packed in the same way as worker.exe and MicroClip, some under up to three layers.

Rapid Ransomware encrypts files using a different AES-256 key for each file, in Cipher Block Chaining (CBC) mode. This key is backed up in the file, encrypted with a per-machine RSA key, the private key of which is also backed up in the file, encrypted with the attacker's public RSA key. Before running, Rapid Ransomware deletes shadow copies and disables Windows' Automatic Repair startup utility. For persistence, these samples use both the Run registry key and scheduled tasks. No keys, or any other information, are transmitted to the attacker's servers, (except for, possibly, an encrypted file sent by the victims) so a decryptor most likely contains the per-machine private RSA key, obtained from the files sent by the victims.

This ransomware comes in two versions; one known as "Rapid" and the other as "Rapid 2.0". They share many similarities, but the second brings some improvements, such as the file extension being random (the first version had hardcoded file extensions, ".rapid" or ".no_more_ransom" depending on sample). In addition, the encryption algorithm was changed to AES-256 in Cipher Feedback (CFB) mode, the file reading and writing was tweaked to use memory mappings for improved performance, the builds can be configured with different e-mail addresses or RSA keys by changing a ".cfg" PE section, and more. However, only the first version is configured to exit in case the user's preferred date, time and number format is configured to Russian.

Unfortunately, the ransom notes (Fig. 6) produced by these samples only contain two email addresses that a victim should contact, so there is no way for us to track ransom payments or connect these samples with the clipboard stealers or miners through cryptocurrency addresses. However, we can connect ransomware samples between themselves using overlap in the e-mail addresses contained in the ransom notes, and we could establish their connection with worker.exe and other payloads using our telemetry.



```
Hello, dear friend!
All your files have been ENCRYPTED
Do you really want to restore your files?
Write to our email - secure-it@tuta.io or mariode@cock.li
and tell us your unique ID - ID-NQTY8W1W
```

Fig 6. Ransom note from one of the Rapid Ransomware samples

Another ransomware that these attackers deployed on the victims' computers was Nemty. Existing cybersecurity literature already covers it extensively, so we will not dive into its inner workings. As with Rapid Ransomware, we cannot connect it with "worker.exe" and other payloads using cryptocurrency addresses, since the ransom notes only mention a ".onion" URL and an identifying key. However, we are able to make a firm connection based on filenames found in our telemetry.

In an interesting case we found on one of the victims' computers, the attackers deployed clipboard cryptocurrency stealers and a ransomware sample at the same time. In this situation, if the victim pays the ransom, the address to which the ransom is wired is replaced by the clipboard stealer. There are some theories of why the attacker acts in this manner. If no cryptocurrency could be retrieved through the clipboard stealer, by deploying the ransomware the attacker would force the victim to buy virtual coins.

Now, if the same attacker deploys the ransomware, the money will be received either through the ransomware or through the clipboard stealer. But, if the attacker made use of ransomware from the wild, when the ransom is wired, the addresses are changed, and the money will be received by this attacker and not the author of the ransomware. There is also the possibility of trying to extort more money from the victim. If the transfer is redirected, the attacker can claim (and in some cases demonstrate) that they did not receive the money, forcing the victim to send more cryptocurrency.

Miner payloads

One of the first means of extracting some financial gain used by these attackers is the deployment of cryptocurrency miners, almost exclusively for Monero. Using the addresses obtained from the clustering mentioned earlier, we can also attribute miner samples we've seen in our telemetry to the same actor.

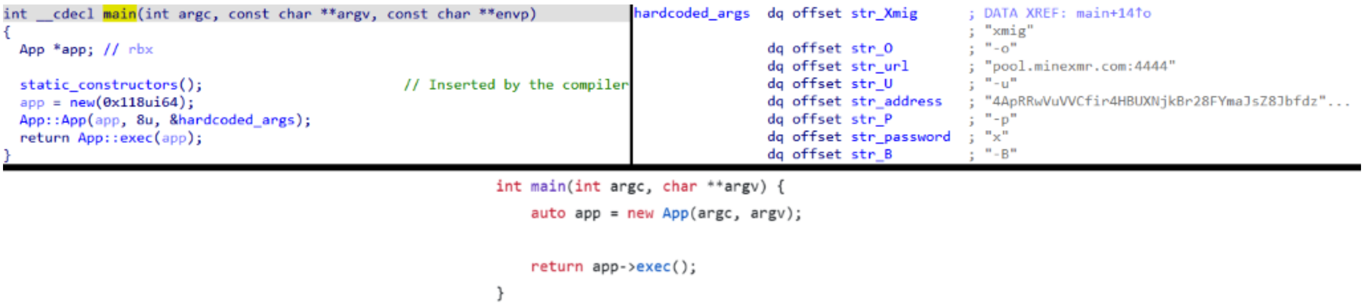
Using this information, we were able to track this attacker to a point before they started using the worker.exe tool. The oldest sample we could find was first seen in our telemetry in July 2017, and distribution tapered off at the start of 2019.

All miner samples are based on the often-abused XMRig suite. However, comparing different builds among themselves, we can see significant variation in the way these samples masquerade, or not, as different legitimate tools and different techniques that are used:

- Most samples connect to a public Monero mining pool, where they provide a hardcoded address to work for (Fig. 7), and, in a few cases, log in to an account with an e-mail address. However, some connect to an IP address (46[.]21.147.75), and others to a domain resolving to this address, (workpc[.]biz). In March 2018, a different sample appeared that mined for a SumoCoin address. Both Monero and SumoCoin are untraceable by design, which unfortunately leaves us unable to estimate earnings for the mining operation.

- While initial miner samples relied on the infection vector or deployment mechanism (such as droppers or downloaders, although these were rarely used) to ensure persistence, many samples were programmed with different techniques starting in October 2017. For instance, they add themselves to the user's Run or RunOnce keys (with a wide range of possible names), one file copies itself to the Startup directory in the user's Start Menu, and one used Scheduled Tasks. Depending on the infection vector, the samples are usually copied to a new directory inside Application Data\Roaming (again, with a wide range of possible names).
- One method the attackers used to conceal the purpose of these samples is altering their version information resource, which can be checked by inquisitive victims in the file's properties. As expected, the replacement information refers to Microsoft services, drivers, legitimate Windows binaries like rdpclip.exe (the RDP Clipboard Monitor), utilities like Microsoft Office Word, and others. However, sometimes not all version information is cleared: some files masquerading as rdpclip.exe still have the OriginalFilename field set to "xmrig.exe".
- Some changes made by the threat actors to disguise miner samples as seemingly legitimate tools include things that would not normally be observed by victims. For instance, the output of some XMRig command-line options, like the usage information (accessible by running `xmrig.exe --help`) or version information (accessible by running `xmrig.exe --version`) is altered, changing the name of the tool with strings like "Microsoft server", "Microsoft", "FileSys", "RdpClip", "Microsoft Office Word", or "Microsoft Office".
- Another difference between the samples we have seen is a side effect of the way most samples hardcode the mining pool URLs and addresses: the XMRig source code is edited to pass a different list of arguments to its command-line parsing routine (Fig. 7). By convention, the first argument (at index zero) is the name of the executable (more exactly, the name that was used to find the executable). As far as we have been able to determine, XMRig does not meaningfully use it in any way, which leaves the choice of this argument purely meaningless. However, these attackers have set it, across different samples, to values like "xmmon", "xmig", "WmiPrvSE", "roop", "maseee", "visit", or simply "xmrig".
- Finally, some samples have some special functionality: one miner we saw between December 2017 and June 2018 also replaces Monero and Bitcoin addresses found in the clipboard.

At the height of the mining operation, around March 2018, this attacker used at least 13 different combinations of the features above, at the same time.



```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  App *app; // rbx

  static_constructors();                    // Inserted by the compiler
  app = new(0x118ui64);
  App::App(app, 8u, &hardcoded_args);
  return App::exec(app);
}
```

```
hardcoded_args  dq offset str_Xmig      ; DATA XREF: main+14↑o
                                        ; "xmig"
                dq offset str_O         ; "-o"
                dq offset str_url       ; "pool.minexmr.com:4444"
                dq offset str_U         ; "-u"
                dq offset str_address   ; "4ApRRwVuVVCfir4HBUXNjkBr28FYmaJsZ8Jbfdz"...
                dq offset str_P         ; "-p"
                dq offset str_password  ; "x"
                dq offset str_B         ; "-B"
```

```
int main(int argc, char **argv) {
    auto app = new App(argc, argv);

    return app->exec();
}
```

Fig 7. Comparison of main function in one of the modified XMRig samples (top left) and official XMRig source code (bottom). Note how, instead of passing the command-line arguments ("argc" for argument count and "argv" for argument values), the malicious build passes a constant length (8) and a hardcoded list of values (pictured top right, with string contents in grey).

Some miner samples we saw were in fact droppers that contain a build of XMRig without a hardcoded destination address, receiving it instead via the command line, a separate runner binary embedded in the dropper, or the dropper itself. Apart from deploying these executables, the dropper also installs their persistence mechanisms (copies itself and the embedded binaries to the Application Data directory and adds itself either as a scheduled task or a Run entry in the registry). Also, one miner sample we saw was brought to the system by a binary similar to other droppers used by this attacker. In this particular case, the binary downloads the miner, instead of dropping it from a resource.

An additional feature of some of the dropper samples we saw was a mechanism of privilege escalation by exploiting the Windows Event Viewer. On startup, this legitimate tool needs to open a .msc file, whose default handler application is defined in `HKEY_LOCAL_MACHINE\SOFTWARE\Classes\mscfile\shell\open\command`, which cannot be edited without Administrator privileges. However, that handler can also be configured in the user-specific `HKEY_CURRENT_USER\SOFTWARE\Classes\mscfile\shell\open\command`, which can in fact be edited without any special rights. Since under default settings the Windows Event Viewer bypasses User Account Control without any user prompt, and the .msc file handler will be run with its privileges, the dropper simply needs to overwrite the per-user registry key above with a path that they can control (either its own filename, the miner's, or the runner's) and run eventwvr.exe (the Event Viewer). That utility will, in turn, start the malicious handler with administrator rights and then silently crash, thus stealthily achieving privilege escalation. While this technique does not succeed in the newest versions of Windows, it can be used to great effect on unpatched systems.

AZORult payloads

In July 2019, the attackers started deploying AZORult for easier, in-the-loop control of victims' computers. As extensively mentioned in the existing cybersecurity literature, this is a stealer trojan that can also be used for basic remote control. The difference from the way worker.exe is used is that, while the latter runs a static script-like file, AZORult gets its commands from a C&C server.

In our telemetry, we've only seen AZORult as a parent process to MicroClip and ransomware samples. We can assume it was used for system discovery, as we have seen these attackers deploy different payloads on different machines, but we can only speculate on whether it was used for purposes like credential stealing. It is well connected to the main cluster, as we have seen it being executed by "worker.exe", located next to cryptocurrency stealer samples with timestamps close to each other, and also used to start known ransomware samples, and possibly more. In addition, the C&C server of one of the AZORult samples has been seen hosting a MicroClip sample belonging to the main cluster. As with most tools used by this attacker, the samples are usually packed in a similar way.

Payload Evolution & Victims

While victims are relatively distributed across the world (Fig. 8), most of them are located in Brazil, the United States and Romania. From our telemetry, these campaigns do not seem to target specific industries, instead trying to reach as many victims as possible.

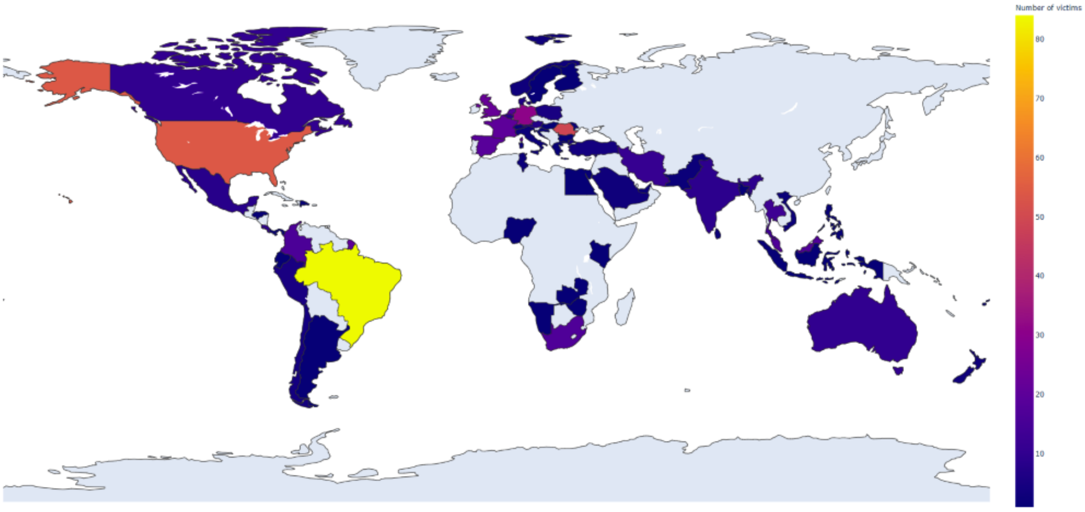


Fig 8. Global distribution of payloads

Since this actor stands out through the wide-ranging use of different tools and payloads, we were curious to see if we can distinguish a trend in their usage, or if the actor prefers one instead of others. As a result, we decided to generate a chart illustrating the evolution for all worker.exe payloads (Fig. 9).

The data was obtained from reports in our telemetry of the samples we can confidently link to this attacker, mostly through connections to the main cluster. The top plot contains the number of infection reports (a different sample type first seen on a victim's machine) per month, and the bottom plot (the rug plot) represents each individual report as a tick, regardless of the monthly distribution.

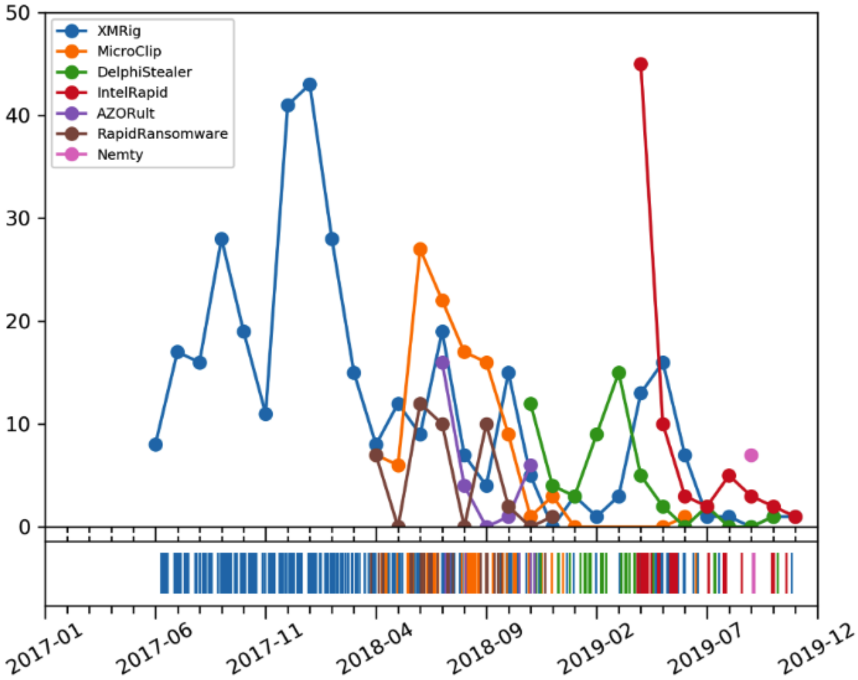


Fig 9. Evolution of payload types

From this chart, we can see that only miners were used before approximately April 2018, but since then a wide variety of tools were employed, especially until the early months of 2019.

The AZORult component was only used between July and November of 2018, right until DelphiStealer started being spread. For zoomed-in versions of this graph for ransomware and clipboard stealer payloads, see figures 10 and 11.

On this graph, worker.exe came into use (February 2018) at a time of marked decline in miner distribution, possibly as a trial run, before its usage with serious delivery of other crimeware payloads starting in April.

The fact that the actor used multiple types of clipboard cryptocurrency stealers always intrigued us. If we zoom the previous chart and consider only this type of payload (Fig. 10), some interesting observations emerge.

While there is some small overlap, we can see that the attackers first distributed MicroClip, until November 2018, then DelphiStealer, and, since April 2019, IntelRapid with rare sightings of the other generations.

We can also see that IntelRapid was distributed much more aggressively, even If, over time, MicroClip seems to have been spread to more victims. In this case, the rug plot proves more useful, as we can more clearly approximate when in the month the campaigns occurred (when reading the rug plot, note that the ticks and labels on the bottom of the charts correspond to the middle of the month).

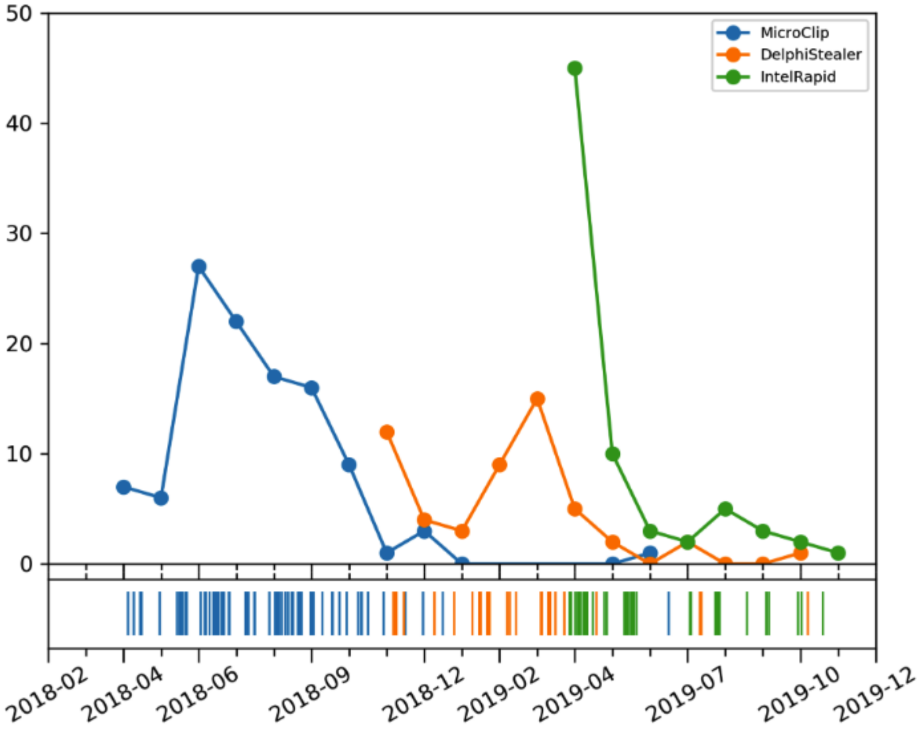


Fig 10. Evolution of cryptocurrency stealer generations

Finally, we zoomed into part of the graph above to study the usage of ransomware by this author. Apart from a month of Nemty usage in September 2019, these kinds of payloads were only distributed until December 2018. In addition, the relatively low number of infections (at least according to our telemetry) show that deploying ransomware was not seen as the main purpose of these attacks.

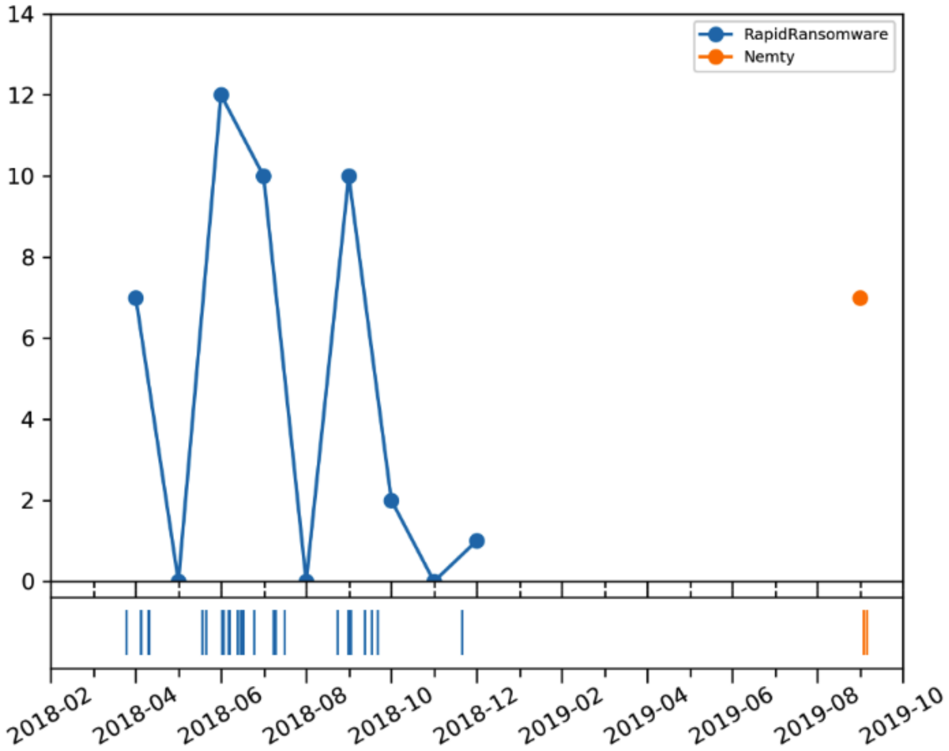


Fig 11. Evolution of ransomware usage

Recommendations

This final purpose of this complex campaign is to extract virtual currency from victims using multiple types of payloads. According to our telemetry, throughout the attacks, we observed that the campaign does not seem targeted. The threat actors seem interested in compromising as many victims as possible since the financial gain is proportional with this number.

The threat actors have used ingenious techniques and tactics to follow through with the attacks, some of them taking advantage of misconfigurations or oversights. With the complexity of today's environments and because managing such environments can often be overwhelming, tracing each aspect that might represent a risk is very difficult, and omission can happen.

There are lots of aspects to take into consideration when we talk about cyber security, and reviewing these aspects sometimes can prevent the success of an attack. For example, if the policy "Do not allow drive redirection", located in "Computer Configuration > Administrative Templates > Windows Components > Remote Desktop Services > Remote Desktop Session Host > Device and Resource Redirection" is enabled, the infection mechanism for this attack can be prevented.

Bitdefender has a dedicated project meant to help identify these types of oversights and misconfigurations and to offer suggestions for their remediation so that certain types of TTPs can be prevented and to make risk management more manageable. For more information: https://businessresources.bitdefender.com/endpoint-risk-analytics-endpoint-security-whitepaper

Tactics, Techniques and Procedures (TTP)

The MITRE ATT&CK matrix provides a mechanism to share TTP indicators used by threat actors from a public knowledge base, where techniques (general ways to achieve goals) are each assigned a unique identifier and attributed to one or more tactics (high-level goals). The procedure is the specific description of a way to apply a technique. Below, we have an overview of how the abilities and the in-the-wild behavior of these campaigns map onto the ATT&CK matrix:

Tactic: Initial Access

Technique: *T1078 "Valid Accounts"*: Initial access is through Remote Desktop, and existing cybersecurity literature points to the use of RDP bruteforcing.

Tactic: Execution

Technique: *T1059 "Command-Line Interface"*: We saw worker.exe and payloads being started by cmd.exe, likely during a Remote Desktop session.
Technique: *T1061 "Graphical User Interface"*: We saw worker.exe and payloads being started by explorer.exe, likely during a Remote Desktop session.
Technique: *T1064 "Scripting"*: worker.exe's "config.ins" file is essentially a script (a list of instructions to be executed).

Tactic: Persistence

Technique: *T1060 "Registry Run Keys / Startup Folder"*: Most samples make themselves persistent by adding themselves to the Run key (miners, ransomware, MicroClip, ClipDelphi) or the Startup directory in the Start Menu (IntelRapid, miners).
Technique: *T1053 "Scheduled Task"*: These are also used when enough rights are available to persist IntelRapid, RapidRansowmare, or XMRig infections.
Technique: *T1158 "Hidden Files and Directories"*: Some samples install themselves or their payload as hidden files.
Technique: *T1078 "Valid Accounts"*: Once an attacker uses valid credentials to infect a machine, it can repeat this at any time, unless security measures are applied (firewall, password change).

Tactic: Privilege Escalation

Technique: *T1088 "Bypass User Account Control"* and *T1068 "Exploitation for Privilege Escalation"*: A privilege escalation exploit abusing the Windows Event Viewer is employed by some XMRig droppers.

Tactic: Defense Evasion

Technique: *T1036 "Masquerading"*: Almost all samples hide their purpose through legitimate-sounding filenames, binary version information, names used for persistence, and more.

Technique: *T1027 "Obfuscated Files or Information"*: Many worker.exe and payload samples are packed to make reverse engineering more difficult.
Technique: *T1158 "Hidden Files and Directories"*: Some samples install themselves or their payload as hidden files.
Technique: *T1107 "File Deletion"*: Many payloads delete the original sample after copying it to its preferred persistent location (for example the Application Data directory).
Technique: *T1088 "Bypass User Account Control"* and *T1211 "Exploitation for Defense Evasion"*: A privilege escalation exploit abusing the Windows Event Viewer is employed by some XMRig droppers.
Technique: *T1064 "Scripting":* Using the «config.ins» script, the worker.exe component performs many actions quickly and, by using the Windows API directly, avoids running command-line tools that may trigger defense mechanisms.
Technique: *T1070 "Indicator Removal on Host"* and *T1112 "Modify Registry"*: worker.exe contains a command that deletes the Run dialog history.

Tactic: Credential Access

Technique: *T1110 "Brute Force"*: Existing cybersecurity literature points to compromise by credential bruteforcing through RDP.

Tactic: Discovery

Technique: *T1082 "System Information Discovery"* and *T1033 "System Owner/User Discovery":* worker.exe›s «information» command lists information like Windows version, architecture, CPU name, the domain of the logged-in user and whether they have administrator rights.
Technique: *T1087 "Account Discovery"*: The worker.exe component has a "userenum" command that lists users on the local system.
Technique: *T1135 "Network Share Discovery"*: The worker.exe component supports a "sharedrive" command that finds drive letters associated with network shares.
Technique: *T1012 "Query Registry"*: Apart from using the Windows API, the worker.exe component obtains some of its information using the Windows registry.

Tactic: Collection

Technique: *T1113 "Screen Capture"*: The worker.exe's can take screenshots of the machine if the "scrshoot" command is executed.

Tactic: Command and Control

Technique: *T1071 "Standard Application Layer Protocol"* and *T1219 "Remote Access Tools"*: Command and Control of worker.exe is done through the Remote Desktop Protocol.
Technique: *T1105 "Remote File Copy"*: Malicious binaries are delivered using RDP's built-in "tsclient" virtual host and network shares.

Tactic: Exfiltration

Technique: *T1041 "Exfiltration Over Command and Control Channel"* and *T1020 "Automated Exfiltration"*: worker.exe's output files are transferred automatically using the "tsclient" virtual network shares over the same RDP connection as payload deployment.

Tactic: Impact

Technique: *T1496 "Resource Hijacking"*: worker.exe also deploys miners that take advantage of the victims' resources to extract cryptocurrency.
Technique: *T1492 "Stored Data Manipulation"*: Clipboard stealers are also employed, changing cryptocurrency addresses in the clipboard with the attackers' in the hopes that they will be unknowingly used as a payment recipient.
Technique: *T1486 "Data Encrypted for Impact"* and *T1490 "Inhibit System Recovery":* Some of the payloads used by these attackers are ransomware, which encrypt the user's data until a ransom is paid. To increase the likelihood of payments, these samples first delete shadow copies and disable built-in recovery modes.

Indicators of Compromise

An up-to-date list of indicators of compromise is available to **Bitdefender Advanced Threat Intelligence** users. More information about the program is available at https://www.bitdefender.com/oem/advanced-threat-intelligence.html.

Hashes

This is a selection of hashes for the files that we encountered over the course of this investigation, categorized by sample type. More hashes are available on our GitHub repository here: https://github.com/bitdefender/malware-ioc/tree/master/rdp_abusers

DelphiStealer (in-the-wild):

05e2f51dddd77c5db48a72a3bf78cbdf865f02105f852fbfd8db35db80dd51fa

08517d1fd0a164526683a8708376abeba63d077705dba54fa6f306163471c9c4

20c942864b74b482b8c7fe07bdb4745388ca6f4e7a90362d271effaa2c21fd19

3cf63caf7f2a3b37699e2e7f254bb5c33a093bda354510ab72daf662e71e66c0

5449db2fc3bbb57a05aff4cb26480ee54f2d7bc9118b1c26b0550680a41fd515

990cfaa8660c8c9118f1e4eebd49ec75844aec93c2385f6cfadac0e94e34799f

b257c4b962c3de4b9d34afbf23d7b09f9f0741bdd66dbc6733225ae2cca909fb

DeplhiStealer (payloads):

05e2f51dddd77c5db48a72a3bf78cbdf865f02105f852fbfd8db35db80dd51fa

20c942864b74b482b8c7fe07bdb4745388ca6f4e7a90362d271effaa2c21fd19

3cf63caf7f2a3b37699e2e7f254bb5c33a093bda354510ab72daf662e71e66c0

5449db2fc3bbb57a05aff4cb26480ee54f2d7bc9118b1c26b0550680a41fd515

990cfaa8660c8c9118f1e4eebd49ec75844aec93c2385f6cfadac0e94e34799f

b257c4b962c3de4b9d34afbf23d7b09f9f0741bdd66dbc6733225ae2cca909fb

MicroClip (in-the-wild):

14873eba6c4513ed5b9fdef6b86b5912cfa7df16422efef7ec490b1445b842cc

30dba80a875553ef8908947bcd37003c328dac366e14c76368361d4f60624cb4

3861e4627f5c1fd769935d6dca6557e6e26a01d130780cd00675ee98c355259b

502ba411b7400f3f3ba531584a71da5140b82bb281d33fc0c4d202d584cf7686

5bc17497a238a15c1a3e31874b418a705c40c13f3c20541cb2504c804c590ab5

5e13c2986360268b54e95e7133ac07e48c72bfcf1be5778c1f1a62b112942925

936729e66021fd42f88c2d4f34520810b2b7d5b4b6b5e2b7b1752433aa0ab937

b53cbaa58d719eadc0bbd127c574f548c016230d31348c9af5f67187b4953398

efeea68805d393bae0fcc8ccdad5dd6fb36136f71d8f3cf9adf1b788fd7eac49

f76b7254ccb8731a6f32979a7f82e40b4c7d88e1f186a3d4477da24ea0a205f6

MicroClip (payloads):

0d1f106eca46b5b4fe9a7e4d5c67405fb12cfe8fd2927f3d40037c03e0fe034e

10c9a324fb345d710fe45b044fd1304b4ad386f761a356a39af3dd12aff7bfc3

2083c01b207b5b7c930ab556490683689cfc9a8118087ae5849fd6416ef1490d

401c5f405220b4911aa695e37918593078cbe130ee1e00f83e820b981e65ce2d

68585f9830f174f4ccd53df5d2396e62c88c3ba07645be50f482a980f8ab03cb

dccc057db24b9178b7851a301885e33c8e7be593deae7bd403446149b5a5e048

fe328a7074d01f2c9080028ee2962fd2f6b6a80de97bb3efe34b04e6aec4bc3c

IntelRapid (in-the-wild):

21808615025f902b62e3f6336ba878451ee9ab63c6588b60537407ded9076424

3e57cf8e94fdd1c2fc41d49aebe87a292add4fd378cb377bbd69b2723343945f

4d020d354ebcb892bce5cf83a4257edd23f1ceaacd8210f1214054d691094c1f

63c8d6ab0e57487a02f63501e1f1b98adc18ef4d7dbf61c2bc977cc5f0a2287b

807a275957f9a0728e6a537d50c97dfd90ae42efa3075e0f531b6a6f353fb007

a36b8cbfe3067f212a3d971d2cb82084d3dac521c58aeb391057ae4d625b9313

d6d1a78c6df058e68a17ecac6ca839d0c40be4b487fb53ba0bd180992eed9c9b

e7aec13a3220183ae0be8266a5c79a957d3fbd6bdae3ef1a698d1b3d85d0d1a6

ecbd14d38dfd754ea68bfc20b61329d11e39c820af370598824c20e8bfebe4c8

f15d91582408c43b39560ab46e290c63fdd7eafa6b21de2a7313968eb8257ccc

IntelRapid (payloads):

05c0f6cee99f2e438c123ba5902fd64c9e064a9ee1b126b3b1b7fb098ef6ccb7

23a102263fa7b2be5bdeb9d7efcb29be73fb90b2bddce6f086b8d1d36b21b39f

35ca6031688f5af36e896cc8f4a345a0069e7d6581cf8eaa8dfe30a2f5bcd376

4db6da27bde99ba28cb9e5cc185ede30ec355e5e8c881c99717765449b045483

509314c535ccee477b2d88dd0cd8e35e78246962f93a9599a9e33f5d3257fff1

936da874fd0428fffe0636ce35a218ce93f11a75d13e4118c44e75eb708c9df0

d04500d89f97daf71b90766e149c315bee2861c61488d870d1a4eccb755ff370

f0ec3e15c4b6aba9a3290199f6ed14eaa86422675f64c50680e65ccd9c04bb84

XMRig (in-the-wild):

167faab88b40c13e481677c76f6541ce5081e7c248d2f2e4f0c9a467df3e42c1

2d199b069ca58b05185b5286bd0bd3deca688fc30f5e1b21a50f496b7f790a7c

4bea0913904e449def5a94d8e428ec85736fbe42f9086b7a08d085e37e44dd94

86072e0e7f9bc61e22dbc69b70df0b2c95a0badfdf352b7fc6d9eb6182118b05

9e1ce9cee6bb35de538afa1ddee8743680d6662d508167ceaeba828d6b74010d

a647c0a8298da14f1402cd16958b31318892b898cadc829dd21436c5b2dd1cb8

cd2f7f392f02d711b0260dfc080095ea1d2121244b9db9706553bea7c9d29b28

ce9c39ea38114e0bc32161f8ea4b69b4de78991e3a5742763dd20f911ea79855

XMRig (payloads):

43fee0d62ef0f1991a62cf6a195a5f9c2d7176fdd25fdf585d40ec85e8180a52

4bea0913904e449def5a94d8e428ec85736fbe42f9086b7a08d085e37e44dd94

5aea07985114053368601c9e3aa6c6564349cefcba3ef69c986ce8749c9fa99d

5e15b1c7d8b5564d960010331d753d01e9286a45573a749d7836482abe72dc5f

6273a8c3f1d748388ca350d24bd8fda86151253b67c26b03ccb5cee5d3499270

9b934875d44383cdab400506927804809ba665ae80f7d2347db87d583115afbd

a915c72774d31f71143fead1b6fc38e11b283985ae5faefd6c1a90d0de01b3e4

b84f7256caf521d037d869e6994292248372f2d6fbaf9e8fe1d7d76de9d20265

AZORult (in-the-wild):

1de9348612df332f39c07ecfb6680d16d8a8978e790183a28878328766e5d843

383d115201f7ccd9647f4f018c1a40a6625bf0c442558dc00ab83c7dd27621f1

6adfdb690e2932397a1a5e4edff6f4ad6a3a380400707a71b35c36abb52e8553

6b1cce3fbac60e735586002b2768059798fefd315010f8d756f40b7c9110a7b5

88341ac8d35390d70c49f99fd93cd1b139d156fc8bb591c6a1808cc4f83f6218

b27e3d051882f5554de01603a7ababed0ba587f2854e375853becad1527d811f

ed089448d76df8133fcabfe21e6f6687d71edb6329cc8fea63da42db23487abd

AZORult (payloads):

35fc256c67aa816e2820779dc1a7cb605cf5b7a6a2464280853533d95826a93a

bbdf3076132fd5ff62c897e658e65025851d1ddd00e56f71402f14ed10dd271d

RapidRansomware (in-the-wild):

029b286197dc8842a6126004d0111aad3f947770fe269c7ee8d47fa4d8b750ef

39a4cca7be143daba488b864e23ba16d9492b70e9e49cc2e574e8e896c717239

76aa2d373ef3cb101c41808a49054e78747ebc40cf1a5f0993f788609670d07c

d33eb6000b793805d6071c89cea96d224408e39d79b6931dea41f53887b058cc

e89dd5d78b75ddc8ba66e30ae51a2df7c4afe186249b9c58b3938b7b394fb6f0

ec5a057f0171aff28630880bee8a5378e0090d27662f949f28dffa3af99658ba

fa049f7bf36b1287bafc72de17ef11be6f852b6f6f1deebe590d0910aacf563e

fbbcd4665b581d30d52942aa0dbb469ed35d07a3318bb6abb4d3ba7271b6bc3e

RapidRansomware (payloads):

041ae2e080b8e03a3b29ea60a688a2235956d9caac188f16c3463b923a199597

3e975f3d9316f3555750f06facb9aea5eee4d87f269099138c20f283fd2f93c6

7acb4b6472a3a265bd7752cd2691250b39f49e21cf4c83eea2ad1ecb9dbf55b7

bea170429c42ad7902e198612f589a686f45da81324ee87cfff4ed5bb52783dd

d0ad4503386affa3e7de701047ab5a36360483a5161d8498521b5e5432d0113e

e6c5380cc85eefa4725f50b66fcf13732f1052f809f82bf2a54d125e049a8b2f

Nemty (in-the-wild):

399503962b14b9b08824e15c59a33efcd95e7980fa02fd93908697c67632df20

455d509757dcad64cb1652d2b83ea5fb3e7b7f3459eb4d7e730a3b8f151bdb51

481b450a77918de85f17dccf1bba92f366969c6a103bad6053f2a36324f6d8de

dffec313da406f87b9caa9553838e8d94d25d00d6921dc8f986ff624d6851662

Nemty (payloads):

dffec313da406f87b9caa9553838e8d94d25d00d6921dc8f986ff624d6851662

Network indicators

```
workpc.biz
mytele.ga
megabitcoin.life
mandevelopm.org
petrofig.beget.tech
46.21.147.75
```

```
http://mytele.ga/index.php
http://mytele.ga
http://megabitcoin.life/index.php
http://mandevelopm.org/index.php
http://petrofig.beget.tech/xxx.exe
http://petrofig.beget.tech/minexmr.php
```

Windows Scheduled Tasks

```
Encrypter
EncrypterSt
Intel Rapid
Windows Media Video
Windows Logon Application 64
Realtek Sound Blaster Driver
Realtek Sound Help
```

```
Utility Service
WinMediaPlay
WindowsMediaPlayer32
Intel Rapid32
Client Server Runtime Process
AppDataLow
ReadMe
```

Registry locations

Registry values are shown as “`key > value`”.

Persistence registry values:

```
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run > Client Server Runtime Process
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run > RDP Clipboard Monitor
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run > winlogon
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run > Microsoft server
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run > UpgrdWndws
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run > Word Processing Software
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run > Readme
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run > userinfo
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run > Encrypter_074
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run > 12365435
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run > Windows Update Center
HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\RunOnce > Microsoft server
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run > msword
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows\CurrentVersion\Run > RDP Clipboard Monitor
```

Additional registry keys and values used by ransomware:

```
HKEY_CURRENT_USER\Software\EncryptKeys\

HKEY_CURRENT_USER\SOFTWARE\AppDataLow\Software\

HKEY_CURRENT_USER\Software\EncryptKeys > local_enc_private_key

HKEY_CURRENT_USER\Software\EncryptKeys > local_enc_private_key_len

HKEY_CURRENT_USER\Software\EncryptKeys > local_public_key

HKEY_CURRENT_USER\Software\EncryptKeys > local_public_key_len

HKEY_CURRENT_USER\SOFTWARE\AppDataLow\Software > key0

HKEY_CURRENT_USER\SOFTWARE\AppDataLow\Software > key1

HKEY_CURRENT_USER\SOFTWARE\AppDataLow\Software > key2
```

Bitdefender is a global security technology company that delivers solutions in more than 100 countries through a network of value-added alliances, distributors and reseller partners. Since 2001, Bitdefender has consistently produced award-winning business and consumer security technology, and is a leading security provider in virtualization and cloud technologies. Through R&D, alliances and partnership teams, Bitdefender has elevated the highest standards of security excellence in both its number-one-ranked technology and its strategic alliances with the world's leading virtualization and cloud technology providers.
More information is available at http://www.bitdefender.com/.