Inside Hidden Cobra

A Look At a Nation States' Cyber Offensive Programs

Ryan Sherstobitoff and Thomas Roccia
McAfee Advanced Threat Research
About the Presenters

Ryan Sherstobitoff  
Sr. Analyst Major Campaigns - Advanced Threat Research  
https://www.mcafee.com/blogs/author/ryan-sherstobitoff/  
@R_Sherstobitoff

Thomas Roccia  
Security Researcher - Advanced Threat Research  
https://securingtomorrow.mcafee.com/author/thomas-roccia/  
@fr0gger_
Agenda

▪ The Goal of a Nation State & Geopolitical context
▪ Background on nation state cyber offensive programs
▪ Who is / what is Hidden Cobra
▪ Known TTPs
▪ Arsenal Involved
▪ Code DNA
▪ Conclusion
The Goal of a Nation State & Geo-Political Context

What are the goals of a nation state in the cyber domain?

- Political
- Foreign Policy
- Military
- Financial
- Influence Campaigns

How does the geo-political situation influence cyber offensive programs related to Hidden Cobra?

- Adversary often reacts to sanctions
- Targeting opposition and state enemies
- Seeking foreign military technologies
- Targeting humanitarian aid groups reporting on Human Rights issues in North Korea
Background on Nation State Cyber Offensive Programs

• Most nations have some form of cyber offensive program

• These programs are often designed to accomplish state goals

• Attribution of these cyber attacks are challenging
Who is/What is Hidden Cobra?

• Hidden Cobra refers to the U.S Government’s umbrella classification of North Korean cyber offensive programs

• The activity set maps across multiple groups the private sector has different names for

ID: 0032  
Associated Groups: HIDDEN COBRA, Guardians of Peace, ZINC, NICKEL ACADEMY  
Version: 1.2  
Created: 31 May 2017  
Last Modified: 04 October 2019

https://www.us-cert.gov/northkorea
Group Naming Conventions

• The private sector has identified the Hidden Cobra activity set by various names

• The target objectives of these groups are different when compared to each other
A brief Statistical Review

COMMAND AND CONTROL SERVERS

File Types Used in Q4 2019

- PE
- DLL
- DOC
- DMG
- XLS
- MachO
- ELF
- HWP

© GeoNames, HERE, MSFT, Microsoft, NavInfo, Thinkware, Wikipedia

Powered by Bing

McAfee

HITB2020
A brief Statistical Review

### MITRE ATT&CK Mapping

<table>
<thead>
<tr>
<th>Initial Access</th>
<th>Execution</th>
<th>Persistence</th>
<th>Privilege Escalation</th>
<th>Defense Evasion</th>
<th>Credential Access</th>
<th>Discovery</th>
<th>Lateral Movement</th>
<th>Collection</th>
<th>Command And Control</th>
<th>Exfiltration</th>
</tr>
</thead>
<tbody>
<tr>
<td>11 items</td>
<td>34 items</td>
<td>62 items</td>
<td>32 items</td>
<td>69 items</td>
<td>21 items</td>
<td>23 items</td>
<td>18 items</td>
<td>13 items</td>
<td>22 items</td>
<td>9 items</td>
</tr>
<tr>
<td>Spearphishing Attachment</td>
<td>Command-Line Interface</td>
<td>Hidden Files and Directories</td>
<td>Hooking</td>
<td>Connection Proxy</td>
<td>Hooking</td>
<td>Network Service Scanning</td>
<td>Exfiltration</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drive-by Compromise</td>
<td>PowerShell</td>
<td>Hooking</td>
<td>LaunchDaemon</td>
<td>Deobfuscate/Decode Files of Information</td>
<td>Account Manipulation</td>
<td>Process Discovery</td>
<td>Connection Proxy</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exploit Public-Facing Application</td>
<td>Rundll32</td>
<td>Process Injection</td>
<td>Disabled Security Tools</td>
<td>Bash History</td>
<td>Query Registry</td>
<td>Query Registry</td>
<td>Data Encrypted</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>External Remote Services</td>
<td>Scripting</td>
<td>Access Token Manipulation</td>
<td>Hidden Files and Directories</td>
<td>Brute Force</td>
<td>System Information Discovery</td>
<td>System Information Discovery</td>
<td>Data Staged</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hardware Additions</td>
<td>User Execution</td>
<td>Accessibility Features</td>
<td>Modify Registry</td>
<td>Credential Dumping</td>
<td>System Network Configuration Discovery</td>
<td>System Network Configuration Discovery</td>
<td>Data Obfuscation</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Replication Through Removable Media</td>
<td>AppleScript</td>
<td>AppCerts DLLs</td>
<td>Obfuscated Files or Information</td>
<td>Credentials from Web Browsers</td>
<td>System Time Discovery</td>
<td>System Time Discovery</td>
<td>Multi-Stage Channels</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Spearphishing Link</td>
<td>CMSTP</td>
<td>AppInit DLLs</td>
<td>Process Injection</td>
<td>Credentials in Files</td>
<td>Account Discovery</td>
<td>Account Discovery</td>
<td>Automated Collection</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Compiled HTML File</td>
<td>Accessibility Features</td>
<td>Application Shimming</td>
<td>Rundll32</td>
<td>Credentials in Registry</td>
<td>Application Window Discovery</td>
<td>Logon Scripts</td>
<td>Clipboard Data</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Component Object Model and Distributed COM</td>
<td>Account Manipulation</td>
<td>Bypass User Account Control</td>
<td>Scripting</td>
<td>Exploitation for</td>
<td>Browser Bookmark Discovery</td>
<td>Pass the Hash</td>
<td>Data from Local System</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Compressed</td>
<td>Data Transfer Size Limits</td>
<td>Exploitation Over Alternative Protocol</td>
<td>Exploitation Over Other Network Medium</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Hidden Cobra Threat Profile

- Hidden Cobra is using cyber operations as a means of accomplishing state military goals in place of conventional warfare. Hidden Cobra has had some form of cyber-offensive dating back to 2007.

- Objectives of cyber offensive programs

  - More cost effective than conducting conventional war (for a nation state that has heavy imposed by economic sanctions)
  
  - Creates a level of deniability for whom is responsible (often placing blame on false groups)
  
  - Can be used to disrupt or deceive enemies anywhere in the world
Timeline of Events
Modus Operandi of Known Attacks

- Circumventing sanctions by engaging in crypto currency and bank heists.
- Targeting North Korean defectors and opposition groups.
- Seeking access to foreign technologies in the Defense Industrial Base (DIB)

Operation Sharpshooter

- Sharpshooter was a global campaign that appeared in 2018
- New activity appeared in 2019 with additional targets in the Middle East
- A new implant known as Rising Sun was used against targets
- ATR discovered linkage to other Hidden Cobra attributed campaigns
- With this insight we could effectively map back activity to 2017
Operation Sharpshooter

• Actor used compromised servers to host command and control code

• Chinese webshells were used to maintain persistence to the asset

• Actor connected via Express VPN service to manage the hacked assets
Operation Sharpshooter

- Some malicious TLS certificates were identified and associated with C2 infrastructure
- Based on the TLS certificates we identified more C2s using the same certificate
- In these operations we often find shared TLS certificates use for C2 protocol, this enables hunting for more infrastructure
Operation Sharpshooter

Connections to other operations
Operation Sharpshooter

• Backend was based on Python code, other iterations were found written in ASP language

• Backend used a multi-layered approach to relay commands to a master server

• Backend was custom coding written by the adversary

• We can date the usage of this server to 2017

• ATR discovered additional C2s with more implants from previous campaigns that used the Sharpshooter backend framework
Operation Sharpshooter

- **Free**: write infected end-point’s IP to a log file called jquery2017.js
- **Query**: Write the data gathered from Rising Sun implant
- **Suggestion**: read the data from the name file and present it to intermediate C2
- **Result**: send the results of command execution to actual C2
- **Set**: obtain a new C2 IP address of the actual C2 (master)

Command handler and data acceptor (mainmenu.php)

```
$var1_enum=<random_number>&page=<request_type>&wr_id=<encoded_time_stamp>&session_id=4K4+base64 encoded data
request_type="
  "free" //indicates initial recon data - first connect to CnC
  "query" //indicates request to fetch the command if from the CnC
  "suggestion" // indicates request to fetch additional data from CnC
  "result" //indicates data obtained from the command's execution on the endpoint by RisingSun
  "set" //indicates command for the CnC to set the IP of the actual CnC server in its config file
```

Data Format

Obfuscation of Commands
(random names with no meaning)
Operation Sharpshooter

- Additional functionality custom coded

Connection opened to the actual command and control server by the intermediate command and control server.

```
if ($file_exists($ConFile)) {
    $fp = @open($ConFile, "r");
    $config = @read($fp, filesize($ConFile));
    @close($fp);
    $configary = @explode(';', $config);
    $ip = $configary[0];
    $port = $configary[1];
    $fp = @fsockopen($ip, $port, $errno, $errstr, 30);
```

Check IP against hashed IPs

Delete Log Files Function
Operation Sharpshooter

- Designed to target Middle East aerospace companies
- First stage implant used by the actor to collect basic data and install further implants
- Retrieved by Framework.php hosted on the command and control server
- Capabilities
  - Gets HTTP user agent
  - Collects and sends file path with running processes
- As a response to HTTP POST, Vendor.php sends apple.png (Rising Sunv2) to Mypng.png
- Once the contents of apple.png file are downloaded from CNC, decrypts Rising Sun v2 into memory
Operation Sharpshooter

- Tracking additional C2s was possible by knowing the HTTP request format associated with command interpreter.
- Command interpreter accepts a specific format, C2 backend provided insight.
- We discovered additional C2s hosting ASP code instead of PHP.
- This indicates the backend was adapted into two code formats to be able to be run on any kind of platform.
- In the request header ‘Accept-Language’ we identified North Korean language set.
Operation Sharpshooter

- Vendor PHP file is used to
  - Log remote IP and identifier to a log called jquery2018.js with timestamp
  - Whitelist checking of client IP against specific MD5s
  - Checks HTTP User Agent
  - Checks to see if the POST request contains the parameter alive=verify_session
  - Script will serve the file apple.png to the infected client

```php
if(!empty($_POST['alive'], "verify_session") && file_exists("apple.png"))
{
    @readfile("apple.png");
}
```

Vendor.php serving apple.png to downloader
Variations of Rising Sun can be traced back to as early as 2015.

Another indication that the backend framework has been used for years to support operations.

ATR can trace a lineage of samples originating in the public domain going back to 2017.

Operation Sharpshooter

The Evolution of Rising Sun Implant

- Execute commands
- Get drive information
- Launch process from Windows binary
- Get processes information
- Terminate process
- Get file times
- Read a file
- Write a file to disk
- Delete file
- Get additional file information
- Connect to an IP address
- Change file attributes
- Variant of change file attributes

Variation Similarities

C2 configuration information: Contains URLs and IP addresses of the C2C servers

Local configuration data: Contains Privilege names, VM related strings, Registry keys etc.

Differences

<table>
<thead>
<tr>
<th>C2 Configuration Location Data</th>
<th>Communication Mechanisms</th>
<th>Deployment Techniques</th>
</tr>
</thead>
<tbody>
<tr>
<td>V1.0 Hardcoded configuration data blobs in the implants itself.</td>
<td>V1.0 HTTP POST requests with optional HTTP data in a specific format.</td>
<td>V1.0 Distributed via malicious document that inject shellcode into Word process.</td>
</tr>
<tr>
<td>V1.1 Uses a file on disk in the currently logged in user’s profile folder to read the C2 configuration data from.</td>
<td>V1.1 Different set of HTTP headers to transmit the data to its C2C.</td>
<td>V1.1 Distribution techniques currently unknown.</td>
</tr>
<tr>
<td>V2.0 Embedded resources in the binary containing the RC4 encrypted C2C data.</td>
<td>V2.0 Uses SSL to connect to its C2 IP addresses with hardcoded certificates.</td>
<td>V2.0 Distributed via downloader binaries.</td>
</tr>
</tbody>
</table>
• Additional activity was observed in 2019 targeting an Israeli defense contractor

• Within the Accept-Language parameter in the email header, Korean language was present

• Attached file exploited CVE-2018-20250 involving a WinRAR vulnerability

• Masquerading as SysAid product documentation that actually contains a Rising Sun downloader
Tools and Implants

**US CERT Classified Implants**
- BANKSHOT
- BADCALL
- HOPLIGHT
- TYPEFRAME
- KEYMARBLE
- SLICKSHOES
- BUFFETLINE
- ELECTRICPHISH
- ARTFULPIE
- CROWDEDFLOUNDER
- BISTROMATH
- HOTCROISSANT

**Industry Classified Implants**
- GOLDRAGON
- RISING-SUN
- HAOBAN
- HONEYBEE
- BACKDOOR ESCAD
- BACKDOOR AKDOOR
- BACKDOOR NUKESPED
- BACKDOOR DESTOVER
- TROJAN AKDOOR
- TROJAN HWDOOR
- BRAMBUL
- JOANAP
Implant Development – the past, the present and the future

- Several implants have long development timelines lasting years
- Some implant families have appeared recently with new variants
- Dataset is based on samples observed by McAfee Labs
Implant Development – the past, the present and the future

BACKDOOR DESTOVER

TROJAN NUKESPED

TROJAN.WIN32.NUKESPED
Implant Development – Trojan Hwdoor

• HWDoor is a broad anti-malware detection name for a family of Hidden Cobra backdoors

• HWDoor has been in existence since Operation Troy

• New versions of this backdoor have appeared in 2020
Implant Development – Backdoor Escad

• Escad is an implant that has been associated with Hidden Cobra for years

• Escad is a listening implant installed on victim machines

• Variants of Escad have been tied to numerous high profile intrusions such as the Sony Pictures incident

• Last active development of Escad was April 2019
Using Graph Correlation to identify malware DNA

• Using visualization for:
  - Trends
  - Evidences
  - Similarities

• It can be scalable and can be used on thousand of samples.
• It spots similarities between them.
• It helps to draw hypothesis.
Graph Theory

- A graph is a structure amounting to a set of objects in which some pairs of the objects are in some sense "related".

- The objects correspond to mathematical abstractions called vertices (also called nodes or points).

- Each of the related pairs of vertices is called an edge (also called link or line).

\[ G = (\mathcal{V}, \mathcal{E}) \]
Strings Similarity

- String metrics or string similarity measure how similar two strings are.

- The unit that measures string similarity is the *distance* between strings.

- Malware from the same family or compiled from the same environment can share a significant amount of strings indicating similarities between them.

- For this exercise, we extracted strings for all the samples and compared them with a Jaccard distance to evaluate the similarities.
Code DNA – Hidden Cobra

• Extracting a full set of strings from a smaller sample set of Lazarus / Hidden Cobra samples

• Using data science models we determine relationships between samples

• Individual clusters appear that indicate overlaps between families of Hidden Cobra malware
Code DNA – Breaking out into Clusters

- Extracting a full set of strings from a sample set of Lazarus / Hidden Cobra samples
Clustering by PE Rich Header

- PE Rich header is a useful signature for tracking similar samples, but be aware of false flags
- 324 Samples from 2018/2019 with Rich Header information generated
- Intersections between some malware families indicate shared development environments
Clustering by PE Rich Header

- Breaking out the clusters reveals interesting links
- Several malware families were found to link to each other based on common development environments
- The same developers were responsible for multiple clusters of implants.
Code Similarities

- Code similarities is used to identifies similar functions or part of code of a sample.

- To scale this part we used the Machoc Hash.

- Machoc is a fuzzy hash of the Control Flow Graph (CFG) which is a representation of the function call in binary.

- The Machoc Hash can be used to calculate the similarities between two samples, and it is reliable enough for malware research.
Code DNA – BankShot v.s BadCall Code Sharing

- Clustering with data science models shows that BADCALL and BANKSHOT share a significant amount of strings.
- Further code analysis indicates 65% similar functions.
- Code overlap exists in the functionality to enable host to act as a hop point and through implementation of Fake TLS method.

SSL Proxy Code

BANKSHOT

BADCALL

Strings comparison

Machoke hash comparison

SSL Proxy Code

BANKSHOT

BADCALL

Strings comparison

Machoke hash comparison

HTTPS://www.us-cert.gov/sites/default/files/publications/MAR-10135536-B_WHITE.PDF

https://www.us-cert.gov/sites/default/files/publications/MAR-10135536-B_WHITE.PDF

D1f3b9372a6be9c024306e445262025179b674e494e2028f7212a6be

9e7 2/7/2016 DLL File

2cffc3dcf8ef45f1020c2bc65fb89444e5223325234a3cac8dabeb63f10f171c
Both uses functionality and load the external library SSLEAY32.dll and WS2_32.dll in the same way.
Code Factory – Shared Functions

• Multiple implant families shared code amongst each other – this is also indicative based on sharing of development environments

• Hidden Cobra uses a code factory type approach in building implants
Take away

• Hidden Cobra is a well organized and aggressive attacker.

• They conduct cyberespionage, sabotage and cybercrime campaign.

• They keep updating their tools and arsenal since more than a decade.

• Following their campaigns along with graph correlation allowing us to proactively detect new threat and draw the story behind.

• Analyzing and study reveal that multiple team inside the group are working with same malware DNA but for different goals.
Thank you.