New reverse engineering technique using API hooking and sysenter hooking, and capturing of cash card access

NetAgent Co., Ltd.
http://www.netagent.co.jp

Kenji Aiko
What is API (function) hook?

• A technique in which you temporarily alter jmp command or call command when an application program calls a function (instruction code) in an external library (.dll or .so files,) to divert the process to an alternative function.

• There are libraries for API hook for Linux (UNIX) and Windows each.
The method of function intercepts

• An intercept that change the head address of functions (Detours).
• An intercept that change the IAT (Import Address Table) which is on the process.
• An intercept that replace DLL.
• Native API intercepts by SSDT alteration.

There are some other methods …
Windows CryptoAPI（1/2）

- Decoding API provided by ADVAPI32.dll.
- Available in Windows 2000 and later.
- You can use many crypt algorithm without professional knowledge.
- Related libraries like Hash, Signature, Confirmation as well.
- SSL communications in Windows environment often uses CryptoAPI internally.
Windows Crypt APIs (2/2)

- Cryptographic train is exported with function names Crypt***.

Crypt functions exported by ADVAPI32.dll

<table>
<thead>
<tr>
<th>Offset</th>
<th>Type</th>
<th>Export</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>77D97F96</td>
<td>.text</td>
<td>Export</td>
<td>CryptAcquireContextA</td>
</tr>
<tr>
<td>77D985F1</td>
<td>.text</td>
<td>Export</td>
<td>CryptAcquireContextW</td>
</tr>
<tr>
<td>77DC0CDA</td>
<td>.text</td>
<td>Export</td>
<td>CryptContextAddRef</td>
</tr>
<tr>
<td>77D9A2F9</td>
<td>.text</td>
<td>Export</td>
<td>CryptCreateHash</td>
</tr>
<tr>
<td>77D9A7B1</td>
<td>.text</td>
<td>Export</td>
<td>CryptDecrypt</td>
</tr>
<tr>
<td>77D9A685</td>
<td>.text</td>
<td>Export</td>
<td>CryptDeriveKey</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>77DC1C49</td>
<td>.text</td>
<td>Export</td>
<td>CryptSignHashA</td>
</tr>
<tr>
<td>77DC1C39</td>
<td>.text</td>
<td>Export</td>
<td>CryptSignHashW</td>
</tr>
<tr>
<td>77D9AB80</td>
<td>.text</td>
<td>Export</td>
<td>CryptVerifySignatureA</td>
</tr>
<tr>
<td>77D9B462</td>
<td>.text</td>
<td>Export</td>
<td>CryptVerifySignatureW</td>
</tr>
</tbody>
</table>
A demonstration (1/4)

• As the data in SSL communication go through CryptoAPI, you can capture them by intercepting cryptographic functions in the process.

Demo 1

Capturing InternetExplorer’s SSL communication
Encrypted Data via SSL

Wireshark can capture SSL (https) communications running on IE as illustrated below: Confirm that data has been encrypted by SSL.

We can watch encrypted data

Encrypted Application Data: 35204A95F1183D47C673ACAF929FBEC68E0844055911D3D...
Data gone through CryptoAPI

- Data gone through Crypt Encrypt/Decrypt can be seen in plain text.

<table>
<thead>
<tr>
<th>send data</th>
<th>recv data</th>
</tr>
</thead>
<tbody>
<tr>
<td>-- CryptEncrypt --↓</td>
<td>-- CryptoDecrypt --↓</td>
</tr>
<tr>
<td>GET / HTTP/1.1↓</td>
<td>HTTP/1.0 200 OK↓</td>
</tr>
<tr>
<td>wave-flash, application/vnd.ms-powerPoint</td>
<td>ETag: &quot;3941-13-42b6325d&quot;↓</td>
</tr>
<tr>
<td>Accept-Language: ja,en-us;q=0.5↓</td>
<td>Accept-Ranges: bytes↓</td>
</tr>
<tr>
<td>Accept-Encoding: gzip, deflate↓</td>
<td>Content-Length: 19↓</td>
</tr>
<tr>
<td>User-Agent: Mozilla/4.0 (compatible .4322; .NET CLR 2.0.50727)↓</td>
<td>Connection: close↓</td>
</tr>
<tr>
<td>Host: <a href="http://www.netagent.co.jp%E2%86%93">www.netagent.co.jp↓</a></td>
<td>Content-Type: text/html↓</td>
</tr>
<tr>
<td>Connection: Keep-Alive↓</td>
<td>ssl.netagent.co.jp↓</td>
</tr>
<tr>
<td>↓</td>
<td>↓</td>
</tr>
<tr>
<td>箏/JA~ X蔵・ケטע↓</td>
<td>鈷/!鉤&lt;メモVI・↓</td>
</tr>
</tbody>
</table>
Security in SSL communications

• An encrypted, simply tapping the traffic will not show the contents.

• While eavesdropping with MITM (Man In The Middle) is possible, reliability and security of the communication is guaranteed by using legitimate security certificate.
Multiple purposes of API hooking

• By intercepting at the very moment of decoding in an application program, the contents of SSL traffic are visible.
• Even the contents of traffic can be altered.
• The contents can be altered no matter whether the security certificate is valid or not.
API hooking is easy

• We can intercept some functions,
  – by using LD_PRELOAD on Linux (UNIX).
  – by installing Detours library which is released from Microsoft Research Team on Windows.

Detours libraries
http://research.microsoft.com/sn/detours/
LD_PRELOAD

• Available on Linux (UNIX).

• Only have to register corresponding .so file in LD_PRELOAD environmental variable.

----- terminal
% gcc –shared –fPIC –o intercept.so intercept.c –ldl
% LD_PRELOAD=./intercept.so target_prog
-----
Detours library (1/3)

• This is function intercept library which is released by Microsoft Research Team.

• This can intercept by changing the first few bytes of target function.

• It’s simple and easy to use.
Detours library (2/3)

- I indicate head few byte of CryptEncrypt function blow.

<table>
<thead>
<tr>
<th>77DA1558</th>
<th>6A 24</th>
<th>PUSH 24</th>
</tr>
</thead>
<tbody>
<tr>
<td>77DA155A</td>
<td>68 1016DA77</td>
<td>PUSH ADVAPI32.77DA1610</td>
</tr>
<tr>
<td>77DA155F</td>
<td>E8 B553FEFF</td>
<td>CALL ADVAPI32.77D86919</td>
</tr>
<tr>
<td>77DA1564</td>
<td>33FF</td>
<td>XOR EDI,EDI</td>
</tr>
</tbody>
</table>

- This is a trivial assembler code, but if we intercept function by using Detours, assembler code will be changed as seen in the picture next page.
Detours library (3/3)

• The first 5 bytes of CryptEncrypt function was changed to “jmp” by detours.dll.
Intercept by changing IAT

• The way of jumping another function by changing IAT in process.

You can see more detail in “Advanced Windows” by Jeffrey Richter.
• We can intercept a function by making fake DLL based on legitimate DLL with identical export function.

- normal

- call or jmp

prog.exe \[\rightarrow\] call or jmp \[\rightarrow\] kernel32.dll
We can intercept a function by making fake DLL between prog.exe and kernel32.dll.
SystemServices hooking (1/2)

• System service (synonymous with system call on Linux) intercept by altering SSDT (System Service Descriptor Table).

• Processing takes place in the kernel land.

Details found at
Hooking Windows NT System Services
http://www.windowsitlibrary.com/Content/356/06/2.html
SystemServices hooking (2/2)

SSDT

<table>
<thead>
<tr>
<th>NtCreateFile</th>
<th>hook_createfile</th>
</tr>
</thead>
<tbody>
<tr>
<td>NtDeleteFile</td>
<td>real_deletefile</td>
</tr>
</tbody>
</table>

.........

SDT

Real code

<table>
<thead>
<tr>
<th>real_deletefile</th>
</tr>
</thead>
<tbody>
<tr>
<td>real_createfile</td>
</tr>
</tbody>
</table>

Fake code

| hook_createfile |

change to hook_createfile
sysenter hooking (1/4)

• In WindowsXP/2003 (x86) environment, processes are handed over to the kernel by sysenter command.

• sysenter is called in ntdll.dll.

• sysenter will jump to the value assigned in MSR.
sysenter hooking (2/4)

The value in eax register shows system call number.
sysenter hooking (3/4)

sysenter executed

1. Load the value of (MSR-174H) into CS
2. Load the value of (MSR-176H) into EIP
3. Load the value of (MSR-174H) + 8 into SS
4. Load the value of (MSR-175H) into ESP

Therefore, sysenter hooking can be achieved by altering (MSR-176H) corresponding to the CPU.
sysenter hooking (4/4)

CPU

<table>
<thead>
<tr>
<th>MSR-175H</th>
<th>f9e73000</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSR-176H</td>
<td>804e0f6f</td>
</tr>
</tbody>
</table>

...........

nt!KiFastCallEntry:

mov ecx,23h
push 30h
pop fs
.............

nt!hook_KiFastCallEntry:

xor eax, eax
.............

Change to nt!
hook_KiFastCallEntry(804e0f6f)

normal

hooking
A Demonstration (2/4)

- Eavesdropping with MITM by using API hooking.

Demo 2

Capturing the traffic of P2P programs
E-money Edy

• Prepaid e-money (technically identical to suica).

• Can be charged by bank transfer.

• Balance can be confirmed in real-time, also can be recharged, using a devoted software.
FeliCa Port (PaSoRi)

• A device to read the data in IC cards directly into PC’s developed by SONY.

• External ones connected through USB also available in stores.

• There are libraries for FeliCa Port available under BSD license.
EdyViewer.exe

- A software to read and maintain the data stored in Edy.
- Can be charged from registered bank account.
- Operable on Windows.
- Official software for FeliCa Port.
felicalib libraries

• Library to access IC cards using an USB-type device (PaSoRi). Licensed under BSD.
  

• Can be used to access e-money’s like Suica, Edy, nanaco.

• Inofficial libraries for FeliCa Port.
A Demonstration (3/4)

• IC card reading tool can be built with felicalib.

Demo 3
Get the information from the IC card
Security of IC cards (1/2)

• Have readable blocks and unreadable blocks.
• Have encrypted blocks as well in IC card.
• Can not be written with felicalib.
• Can not be accessed to encrypted blocks with felicalib.
Security of IC cards (2/2)

- With the official tool EdyViewer.exe reading from encrypted blocks, writing, all possible.

- Uses SSL (https) to communicate with the admission server.
A demonstration (4/4)

• Examine the SSL communication while charging to an IC card.

Demo 4

Capturing the SSL traffic of the official tool
Perspectives (1/2)

• With API hook, communication between the user land and the kernel land can be captured.

• How can we capture the communication between EdyViewer.exe and a FeliCa Port driver?
Perspectives (2/2)

- With sysenter hook, system call can be observed.

- How can we estimate the function call history using the system call history at hand?
Thank you!

Any questions?