A Longitudinal Study of the Little Endian That Could

Andrew White
Senior Security Researcher
Secureworks
What is 9002?

- Remote Access Tool
- Also known as:
  - NAID, Lynx, McRat, McProxy
- First seen in Operation Aurora during 2009
- Used exclusively by targeted actors
- Delivered through strategic web compromises and phishing
9002 Characteristics

- Packed code, plaintext strings
- Unusual import format
- Plugin based architecture
- “rat_Uninstall” event
- LZO compressed traffic
Expected Malware Development

Version 3 – Response to public disclosure

Version 2 – shiny new feature

Version 1.1 – bug fix

Initial version
9002 Branches - Facts

- 4 unique development branches
- No overlap in improvements
- XOR Traffic Branch
  - Altered traffic format
- Disk Artifact Branch
  - Altered binaries at rest
- RAT Branch
  - Integrated plugins into main binary
XOR Traffic Branch

- Focused on modifying traffic format to evade detection
- Three major revisions
  - Two still use “Lynx” as User-Agent
- Debug strings
  - “Dog create a loop thread”
- Event uses format string
  - __rat_UnInstall__%d
Disk Artifact Branch

• Focused on modifying code at rest to evade detection
• Uses different import format
• No events registered
• No identifying strings in binaries on disk
• Configuration and malware payload stored encoded in registry
RAT Branch

- Includes keylogger plugin by default
- Updated configuration format
- Added debug strings
  - “wo try to get better”
- Uses randomly-named mutex instead of event
9002 Branches - Theories

1. Low confidence - Single source producing different versions as requested
2. Low confidence - Impersonation of existing groups as a false flag
3. High confidence - Source code shared amongst affiliated groups
Defending against 9002

- Standard threat actor playbook applies
  - Every target has a unique sample
  - Every target has a unique C2

- Hash and C2 indicators not applicable for initial detection

- Instrument everything – network, endpoint, email, etc

- Take advantage of all malware idiosyncrasies to develop countermeasures
  - Don’t rely on just one!
9002 Countermeasures

- **File**
  - File contains `rat_UnInstall`

- **Network**
  - Packet ends with “19 FF FF FF FF 00 00 00 00 11 00 00 00”
  - User-Agent is Lynx (assumes http proxy required)

- **Endpoint**
  - Event contains `rat_UnInstall`
  - Allocation contains `%%TEMP%%\%s_p.ax`
## 9002 Countermeasure Review

<table>
<thead>
<tr>
<th>Variant</th>
<th>File String</th>
<th>Packet Sig</th>
<th>User-Agent</th>
<th>Event Name</th>
<th>Memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original-1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Original-2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>XOR-1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>XOR-2</td>
<td>✓</td>
<td>✕</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>XOR-3</td>
<td>✕</td>
<td>✕</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>XOR-4</td>
<td>✕</td>
<td>✕</td>
<td>✕</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Disk-1</td>
<td>✕</td>
<td>✓</td>
<td>✓</td>
<td>✕</td>
<td>✓</td>
</tr>
<tr>
<td>Disk-2</td>
<td>✕</td>
<td>✓</td>
<td>✓</td>
<td>✕</td>
<td>✓</td>
</tr>
<tr>
<td>RAT-1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✕</td>
<td>✓</td>
</tr>
</tbody>
</table>
Summary

• 9002 has been in use for 6+ years
• Different development branches with different goals
  – Traffic / Disk / Features
• Hashes and C2s will not protect you against targeted threats
• Detecting intrusions requires a holistic, layered approach
Questions?