Understanding and preventing common attacks

Romain Wartel

7th March 2010, ISGC Security Workshop, Taiwan.
Overview
Underground market

• The main motive behind most security attacks: money.

<table>
<thead>
<tr>
<th>2008 Rank</th>
<th>2007 Rank</th>
<th>Item</th>
<th>2008 Percentage</th>
<th>2007 Percentage</th>
<th>Range of Prices</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>Credit card information</td>
<td>32%</td>
<td>21%</td>
<td>$0.06–$30</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>Bank account credentials</td>
<td>19%</td>
<td>17%</td>
<td>$10–$1000</td>
</tr>
<tr>
<td>3</td>
<td>9</td>
<td>Email accounts</td>
<td>5%</td>
<td>4%</td>
<td>$0.10–$100</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>Email addresses</td>
<td>5%</td>
<td>6%</td>
<td>$0.33/MB–$100/MB</td>
</tr>
<tr>
<td>5</td>
<td>12</td>
<td>Proxies</td>
<td>4%</td>
<td>3%</td>
<td>$0.16–$20</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>Full identities</td>
<td>4%</td>
<td>6%</td>
<td>$0.70–$60</td>
</tr>
<tr>
<td>7</td>
<td>6</td>
<td>Mailers</td>
<td>3%</td>
<td>5%</td>
<td>$2–$40</td>
</tr>
<tr>
<td>8</td>
<td>5</td>
<td>Cash out services</td>
<td>3%</td>
<td>5%</td>
<td>8%–50% or flat rate of $200–$2000 per item</td>
</tr>
<tr>
<td>9</td>
<td>17</td>
<td>Shell scripts</td>
<td>3%</td>
<td>2%</td>
<td>$2–$20</td>
</tr>
<tr>
<td>10</td>
<td>8</td>
<td>Scams</td>
<td>3%</td>
<td>5%</td>
<td>$3–$40/week for hosting, $2–$20 design</td>
</tr>
</tbody>
</table>

Goods and services available for sale on underground economy servers

Source: Symantec
Exposure and motivation

• Grids are valuable to attackers
  – Large numbers of distributed hosts
  – High availability
  – High throughput network

• Grids are also particularly exposed
  – Transparent access/attack propagation from one site to another
  – Large number of identical hosts
  – Heterogeneous skills, staffing and security standards
Incidents in the grid

• Security incidents affecting WLCG sites:
  - All these incidents are standard site security issues
    - Several could have been avoided
How do attacks happen?

What can be done to prevent them?
Typical incident

- Gain local access
  - Stolen account
  - Web application vulnerability
  - Design errors (incorrect AFS ACL, password on CVS)
  - Poor password choice
  - etc.

- Attempt to obtain root privileges
  - Use a (or wait for a new) “local root exploit vulnerability”
  - Exploit the system, gain root access
  - Game over.

- Maintain access with a rootkit, harvest credentials
- Use new harvested credentials against further hosts
- Go to step 1
Most common cause of incidents

• Security incidents are often caused by:
  – Failure to apply security patches provided by vendors
  – Security vulnerabilities on exposed services
  – Poor access control management (ex: root password)
  – Incidents at other sites
  – Unresolved past security incidents (lack of traceability)
  – Incorrect risk assessment (threats were not correctly identified)
Vulnerabilities

• Well documented topic
  – Often fatal on a publicly available service
  – Web servers particularly at risks
    • Immediate impact to the organisation (e.g. defacement)
    • Exposed service may contain user code
    • Difficult to scan for vulnerable Web applications

• Pointers
  – http://cern.ch/security/codetools/
Demo

• Based on CVE-2006-0475
Security patching

• Limit your exposure
  – Remove/disable unnecessary services
  – Implement a local/site firewall
  – Segregate your user communities

• Subscribe to the relevant vendor announcement list

• Ensure all hosts are up-to-date with security patches
  – Do not forget to restart your hosts after a kernel update

• Ensure all the kernel modules can be prepared quickly
  – Sometimes vendors need several months to make a release
  – Important point when purchasing hardware

• Monitor your patching status
  – Many sites think they are fully patched, when they are not
Pakiti

### Pakiti - Patching Status System

<table>
<thead>
<tr>
<th>Show:</th>
<th>vulnerable</th>
<th>unpatched</th>
<th>all</th>
<th>not reporting</th>
</tr>
</thead>
<tbody>
<tr>
<td>Order by:</td>
<td>tag</td>
<td>host</td>
<td>time</td>
<td>kernel</td>
</tr>
<tr>
<td>Select tag:</td>
<td>all</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Tag:

<table>
<thead>
<tr>
<th>Security</th>
<th>Other</th>
<th>CVEs</th>
<th>Hostname</th>
<th>OS</th>
<th>Current kernel</th>
<th>Last report</th>
<th>Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td>Scientific Linux 4.8</td>
<td>2.6.9-89.0.16.EL</td>
<td>28.11.09 04:09</td>
</tr>
</tbody>
</table>

#### Tag: Pakiti client

<table>
<thead>
<tr>
<th>Security</th>
<th>Other</th>
<th>CVEs</th>
<th>Hostname</th>
<th>OS</th>
<th>Current kernel</th>
<th>Last report</th>
<th>Ops</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td>CentOS release 5.4 (Final)</td>
<td>2.6.24-24-generic</td>
<td>25.11.09 21:15</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>2</td>
<td></td>
<td></td>
<td>Scientific Linux SL 5.4</td>
<td>2.6.18-164.6.1.el5</td>
<td>25.11.09 21:21</td>
</tr>
</tbody>
</table>

### http://pakiti.sourceforge.net
Pakiti

http://pakiti.sourceforge.net
How can an attacker maintain its access?
Rootkits

• “Designed to hide or obscure the fact that a system has been compromised.” (Wikipedia)

• Set of software to maintain malicious access to a compromised host

• Rootkit: first generation
  – Change binaries (ps, ls, netstat, lsof, ssh) or libraries (ld.so.preload, etc.)
    – Pros: kernel independent
    – Cons: need to be compiled for the target platform, easy to detect
    – How to detect: check system binaries against trusted instances
      • Tripwire, rpm -V, etc.
Rootkits

- **Rootkit: second generation**
  - Kernel level rootkits
    - Modify kernel structures (syscall table, IDT, etc.)
  - Malicious codes is loaded directly in the kernel
    - Loadable Kernel Modules
    - Direct /dev/kmem access (patch kernel on-the-fly)
    - Direct /dev/mem access (patch kernel on-the-fly)

- **Pros**: difficult to detect, usually includes backdoor features
- **Cons**: LKM may be disabled, access to /dev/{k,}mem may be restricted
- **How to detect**: search for known patterns, or known bugs.
  - rkhunter, chkrootkit, Samhain, etc.
Rootkits

- Rootkit: new trends
  - Filesystem, network stack level rootkits
    - Often used as additional features
  - Hypervisor rootkit
    - The OS runs within the rootkit
  - Debug register based rootkit

- Conclusions
  - Rootkits are getting more and more sophisticated
  - Many rootkits are easily and publicly available
  - Deploying and using rootkits requires little technical skills
  - Root account compromised == “game over”
How to deal with a compromised system?
Incident management

• Procedure to deal with a security incident
  – Contact your local security team
  – Follow your incident response procedure

• Procedure to deal with a compromised system
  – Define roles and responsibilities
    • Communications with other sites, management, users, etc.
    • “Log digger”
    • Forensic coordinator
  – Write down every action taken, including time
  – Evidence collected essential to prevent re-occurrence
    • But unlikely to be usable in court
  – A compromised system has to be reinstalled from scratch
Forensics

• Freeze the scene
• Cut network access and/or switch the host down
  – Remove the network cable, use the power button
• Prevent users/staff to access the system
• Online analysis
  – Kernel, local binaries trust issue
  – Potentially precious volatile memory information
• Offline analysis
  – More time, better tools and working environment
  – Work only on a disk image (e.g. not on the actual hard drive)
  – Impossible to revert to online analysis
Forensics - data collection

• Collect data from the most to the least volatile storage

• Priority order
  – System memory
  – SWAP
  – File system (HDD)
  – Central logs, network logs, firewall logs

• Use appropriate tools
  – dd(1), netcat(1), netstat(8), lsof(8), tcpdump(8), ps(1), stat(1), etc.
  – The TCT (http://www.porcupine.org/forensics/tct.html)
  – Linux forensics live CD (PenguinSleuth, etc. many available)
Forensics - data analysis

• Analyze the data
• Reconstruct the attack scenario from collected data
• Common tools:
  – `strings(1)`, `ldd(1)`, `file(1)`, `objdump(1)`, `grep(1)`, `last(1)`, `kill -STOP`,
  – But don’t trust tools/logs from the compromised system
• The goals are to determine
  – The entry point in the system, vulnerabilities exploited, etc.
  – The actions taken by the attackers
  – What was lost (credentials, etc.)
• Gain expertise
  – `http://old.honeynet.org/scans/scan29/`
  – `http://www.cert.org/forensics/tools/`
Conclusion
When contacted

- When alerted about a security incident

- This is not “for your information only”: action is needed!
- Please check your logs with the information provided
- If you need assistance (or don’t know what to do):
  - Please ask for help! (http://cern.ch/osct for EGEE, etc.)
Soon or later...

- Each site has been or will be affected by a security incident
  - Either by exploiting a local vulnerability
  - Or through a user account from a partner site
- Part of normal operations, just need to ensure
  - It is “cheap” to deal with
  - The overall infrastructure is not affected
- It is essential to prepare for this event to reduce its:
  - Impact (appropriate & timely response, precautionary measures, etc.)
  - Likelihood (prevention, service hardening, etc.)
- Share information, and report incidents
  
  http://cern.ch/osct/incident-reporting.html