By: Aaron LeMasters & Michael Murphy

RETRI: Rapid Enterprise Triaging
What is RETRI?

- RETRI is a new, agile approach to the Incident Response process, consisting of 4 phases with clear entry and exit criteria
- Using special network segmentation and isolation technologies, RETRI allows network operators to run a compromised network without risk to the data and minimal impact on its users.
- It saves you time and money
The first part of this presentation presents a new paradigm for the Incident Response process called Rapid Enterprise Triaging (RETRI), where the primary objective is to isolate the infected network segment for analysis without disrupting its availability.

Part two of this presentation will introduce a new Enterprise Incident Response tool named Codeword that complements the RETRI paradigm. The tool is a free, agent-based tool that is deployed to the compromised segment to perform the traditional incident response tasks (detect, diagnose, collect evidence, mitigate, prevent and report back).
Assumptions

- Mid to large sized network (1,000+ users)
- Distributed, domain/forest type of network infrastructure (ie, “Government style”)
- Full Enterprise Compromise
  - This is a lot of work if only one or two machine are compromised
  - Compelling evidence will be required by CEO’s
- The compromised network segment contains critical servers/services that must remain online throughout response effort
- Forensics per se is not crucial for a successful recovery
Current Recovery Options

- Network shut down and rebuilt from trusted media (1-4 months)
  - Pros: 100% assurance, data exfil cut off ASAP
  - Cons: people can’t work
- Rebuild while online
  - Pros: People keep working (for the most part)
  - Cons: Data exfil continues, bad guys keep a foothold, potential recompromise
A New Method is Required

- The RETRI method attempts to solve the shortcomings of each of the existing methods.
  - RETRI Option:
    - Pros: Data exfil stopped, high confidence in network hygiene, people keep working
    - Cons: Costly - lots of work to setup (but still cheaper in the long run)
Case Study 1
(Rebuild while online)

- Survey Data for 2006
  - On average hacked companies spent 4.7 million on cleanup
    - Cost based on lost revenue, cleanup, and brand damage
    - $182 per record lost

- Survey Data for 2008
  - Average cost rose to 6.6 million (up to 32 Million)
  - $202 per record lost

- Lessons learned from the survey
  - Employee down time cost 3 times as much as the actual cleanup
    - Even with rebuilding the network while online, there is significant downtime for employees
    - If only there was a way to eliminate employee down time
  - Record cleanup was how cost was determined, not number of host / infected machines
  - “First Time” Intrusions cost more
    - 84% of 2008 Survey respondents had previous intrusions
    - 2008 numbers would be much higher if they didn’t have “practice” cleaning up intrusions

Case Study 2
(Rebuilding Offline)

Based on a 2007 incident we worked

- Approximate Total Cost: $7 Million
  - IR Tools / IT Support Overtime / User Downtime
  - An extreme effort was made to minimize down time (24/7 shifts with extensive outside resources being brought in)

- Users were offline for 2.5-3 weeks
  - User base: 1500 users
  - User down time cost approximately $4.5 million
    - 1,500 users * 15 days * 40 hours a day * $50 an hour (average)

- Numbers based on network rebuild, not lost sales or record clean up
  - No PII or User data stolen
  - 100% of network host were rebuilt
    - $2.5 Million in IR tools and Labor
Case Study 3
(RETRI: Estimated Cost)

- 10,000 users / clients
  - Projected Cost (~$2.9 Million)
  - Best Case Scenario:
    - Decision to implement made on Thursday evening
    - RETRI Phase 3 finished by COB Monday
      - Limited user down time (1-2 business days)
      - Start on Tuesday, response proceeds at a casual pace
      - Cost breakdown
        - ~ $576,000 for Phase 3 Labor (Network / Server Admins)
        - ~ $1,000,000 in Software Licenses (list price, without discounts)
        - ~ $650,000 in New Hardware
        - ~ $288,000 in IR
        - ~$384,000 in Re-imaging Labor (deploying and desk side support)
  - Keep in mind, this is a large network which is being 100% rebuilt
  - On average it is 2-3 times cheaper than any other method
- So what is RETRI..
RETIRI’s Phased Approach

- **Phase 1: Preparation**
  - Weeks to months
- **Phase 2: Damage Assessment**
  - 24 hours or less
- **Phase 3: Network Segmentation and Service Restoration**
  - 3-6 days
- **Phase 4: Investigation and Recovery**
  - Whatever is required (users are not affected)
Phase 1 – Preparation

Weeks to months out...
Traditional COOP
- Generally ensures you have backups at an offsite, but....
  - Real-time replicated backups shouldn’t be trusted
- Identify highly critical services and business processes which require Internet connectivity to function

Cyber COOP
- Create a backup plan and identify hardware and software for cyber attack recovery scenario
- Physical media (e.g., tape) backups
- Cloud computing provides no benefit
Resource Considerations

- People:
  - Network Admins, Server and Desktop Support staff, Incident Response Specialists, IDS / IPS Analysts
  - Switch and Router specialists

- Hardware
  - Need servers to restore backups to

- Software
  - Application Streaming Infrastructure (ASI)
    - Citrix $350 per user
    - ThinWorx $199 per user (open to “renting” the software)
    - Quest vWorkspace Enterprise $100 per user
  - IR tools
Don’t forget…

- Scripts / SMS packages
  - Prep to install / remove apps
  - Scripts to change default home page
- User Notifications
  - What will you tell your users
  - What are they allowed to say to outsiders
- Training packages
  - Emails
  - Posters
  - Web CBTs
Virtualization technology enables rapid response and minimizes resource consumption
- Saves on number of physical servers necessary for RETRI network segmentation
- Known good VM images can be restored in moments from backups

This architecture streamlines the use of response tools
- Many tools and applications can be loaded on VMs
- Distributed analysis among analyst teams with common data sets

Leverage software inventory / deployment systems in place
- SMS, Patchlink, Hercules, etc
Know Your Network!

- Where do your assets live?
- What platforms exist?
- Network entry points
- Trust relationships
- “Dark segments”
- Are there any unique dependencies which will need to be addressed?
- Inventory / asset management
  - How will you gauge coverage?
  - If you can’t count your assets...
Phase 2 – Damage Assessment

Within 24 hours of compromise discovery....
Intrusion is detected

- Perform basic incident response to identify the attack vector
- Identify date of infection so backups can be restored from known good sources
- Identify Command and Control method
- Attempt to identify basic malware capabilities
  - Submit samples to AV vendor for rapid signature creation
- Determine the scope of the infection / intrusion
Does RETRI Fit?

- This is a major decision before proceeding...
  - Are critical backups available for RETRI?
    - Domain Controllers, Exchange servers, DNS, File servers, Print servers, Web servers
  - Does the evidence support the decision to begin a network wide rebuild...?
    - Rebuilds are very costly and time intensive
      - RETRI affords you the time to do the rebuild without taking your users offline
    - Some data may be lost
- ...If not, use traditional methods!
- If so... Convince your Boss
Stop the bleeding

- Cut off network access
  - Deny the hackers access to your network and the data you are charged with protecting
    - Implement Firewall or IPS blocks for known backdoors
- Inform management and users
  - Tell them what they can and can’t say...
  - Tell them when services will be restored
- Implement disaster recovery plan
  - Prepare to go to 24/7 operations in all critical IT departments
Phase 3 – Network Segmentation and Service Restoration

3-6 days
**Virtual Routing and Forwarding (VRF)** is a technology that allows multiple instances of a routing table to co-exist within the same router at the same time.
- Because the routing instances are independent, the same or overlapping IP addresses can be used without conflicting with each other.
- Packets get a VRF tag added to them so that routers can distinguish which network they operate on.

**Multi-Protocol Label Switching (MPLS)** is commonly used for Enterprise VRF deployments.
- MPLS allows you to label packets so that the routers can pass packets very quickly based on its label (VRF).

**In Summary:**
- Switch Ports get mapped to VLANs
- VLANs get mapped to VRFs
- VRFs get MPLS labels
- MPLS labels logically separate data as it traverse shared network hardware

Creating the two networks

- The Quarantine Network (Qnet)
  - Using VLAN/VRF technology, place your old network into a new VRF
    - All packets get tagged for your new VRF and are restricted to the new zone based on routing / firewall rules
      - No external connectivity

- The Clean Network (CleanNet)
  - Create an empty VRF which mirrors the other network’s IP space and layout
    - The difference is the CleanNet has connectivity to the Internet
    - Initially this network will be totally empty
What is the Qnet?

- All devices on the infected network must be placed in the Qnet
- The Qnet will require basic network infrastructure
  - DHCP, DNS, Active Directory / Auth Services
  - SMS, Software Deployment Services, Remote Imaging
  - AV, Forensic / IR Tools, Network Scanners
What is the CleanNet?

- A network that will become your new enterprise
  - Email Servers, File Servers, Print Servers, Web servers, Domain Controllers, Authentication Systems, DNS, DHCP
  - Printers can be in the CleanNet VLAN while physically remaining where they are
    - Printers should be verified before being placed in CleanNet
    - This way printers can be mapped from the ASI cluster

- A network that has standard internet connectivity
  - Servers moved over or restored here take the IPs they used to have
  - Firewall, IDS and IPS rules should not need to be modified as you restore services in the CleanNet

- ASI Cluster and App Server Farm
How do you provide access to the CleanNet from the Qnet without risking the security of the CleanNet and the data still residing in the Qnet?

- Very restrictive firewall rules
  - Only Port 443 allowed to specific IPs in the CleanNet
  - All communications with the CleanNet must be authenticated by some 2 factor method (Smart Card, RSA, biometrics)
  - All communications with the CleanNet must be encrypted

- Qnet DNS
  - Option 1: All DNS points to the ASI cluster so users always get to a login screen
  - Option 2: (recommended)
    - ASI.company.com points to the ASI
      - Becomes default homepage in browser
    - All other entries (*.com, *.net, etc) point to a tarpit / IDS for analysis
The ASI Cluster

What is available
- Email
- Office Apps
- Web (IE/FireFox)
- Other critical applications which your users/organization rely on

What isn’t
- Multimedia intensive applications
  - Streaming Video
- Locally installed user applications which require direct access to the internet
  - Anything that requires access to the internet must be installed on the cluster or it won’t work
Securing the Cluster

- No Copy/Paste between Qnet
- No Device mapping
- Only 2 factor sessions, encrypted
- Applications locked down
  - Consider disabling Javascript on browsers (or use noscript) and office products
- DEP enforced on all running process
- User permissions extremely limited
- ASI Clients become “Dumb-Terminals”
Moving The File Server...

Before moving it to the CleanNet
- What do you do with a multi-terabyte file server?
  - Scan with multiple AV solutions
  - Scan with IR tool for known bad hashes

After the Move
- On the ASI
  - Enforce MOICE (Microsoft Office Isolated Conversion Environment) on all Office files
  - Disable JavaScript in Adobe Acrobat
  - No untrusted executables
Neutralizing file format threats

- **What is MOICE**
  - Converts 2003 and previous Office files (binary formats) to xml
  - Conversion is done in a sandbox of sorts
  - Exploits in files cause a safe crash in conversion without exploiting user

- **What is DEP**
  - *Data Execution Prevention (DEP) is a set of hardware and software technologies that perform additional checks on memory to help prevent malicious code from running on a system.* (microsoft.com)
  - Software protected by DEP is much harder to exploit

- **PDF Viewer**
  - How many of you use Adobe Acrobat on your network?
    - Adobe Acrobat == Massive Vulnerability / Backdoor
    - Ditch it and get Foxit, etc
Restoring User Services

- Enforce 2 factor and reset any accounts which are not 2 factor
- Install ASI client on all Qnet host
  - Make ASI the default home page on all client machines
- Remove / hide all office applications (in Qnet) with SMS
- Train users
  - Email
  - Handouts, Posters
  - hands/virtual training
  - memos, TPS reports, etc
What’s next?

- After restoring operations, the focus shifts to cleanup, recovery, and attribution
- Verify initial assumptions and analysis
- Deeper Malware analysis of collected samples
  - Submit samples to AV vendors
- Network data analysis
- Verify attack vector (root cause)
- What data was taken – regulatory implications (HIPAA, SOX, etc)
- “Deep dive”
Introducing **Codeword**: A tool for rapid detection, recovery, mitigation and cleanup

**Phase 4 – Investigation and Recovery**
Tools of the trade

- **Commercial forensics tools:**
  - Enterprise versions are very costly
  - Complicated
  - Steep learning curve
  - Require expensive full-time resources
  - Heavily forensics-focused, not recovery-focused
  - Mostly bulky, slow and painfully “thorough”

- **Other enterprise “security tools” (e.g., Scanners, AV, HIPS):**
  - Poorly configured, not watched
  - Not widely or consistently deployed
  - Require problematic integration with infrastructure

- **Free/Open source tools:**
  - Mixed capabilities
  - Enterprise design not in mind
Bottom line

You need the 10-day solution, not the 90-day solution
Critical data is easy to get

- There is a limited set of critical data that an analyst must be able to quickly search and retrieve to identify a majority of common infections:
  - Disk indicators: file name, size, hash, PE characteristics
  - Memory indicators: process name, loaded modules, command line arguments, strings in heap
  - Registry indicators: GUIDs and other static values
- Codeword’s main purpose is to quickly expose this information in a meaningful way, so that an analyst can come to a reasonable conclusion about an enterprise-wide, active infection in minutes to hours
- Of course, it also has more advanced features ;-)
Frustration with commercial forensics tools
- Bugs
- Time wasted on service calls
- Licensing headaches
- Inconsistent results (v5.5a != v6.5.1 ??)
- Over-engineered, misses the simple use cases
- Core capabilities aren’t customizable
- Lacking robust rootkit detection

Fruitless search for a comprehensive open-source alternative

The agile, responsive attitude of Codeword fits perfectly with RETRI
Imagine combining these enterprise tools into one simple, easy-to-use tool:

- Vulnerability & AV scanners – Codeword uses signatures to detect and scan host locally
- Enterprise forensic tool – Codeword uses forensic techniques to collect malware evidence in an agent-based framework
  - Rootkit detection – think GMER or Ice Sword
- Extensible – define what you consider to be malicious
- Free...
Current Capabilities

- **Detection** - Uses registry, file and memory “signatures” to detect malware and misconfigurations and heuristics to identify anomalous behavior
- **Evidence collection** – collects any malicious files discovered
- **Reporting** - Results are collected, compressed/encrypted and uploaded to a secure location in the Qnet (Sftp, http, smtp, or network share)
- **Mitigation** – disable devices, uninstall apps, change system policies, etc
- **Cleanup** – kill processes/threads, delete/rename files, delete/clear registry entries, restore boot sector
- **Remote Analysis** – connect to agent from admin interface
Major Features

- Write your own **signatures** to find malware
  - Simple signature logic – use file names, sizes, hashes, etc
- Tweak advanced **heuristics** for better detection
  - User mode, kernel mode, and low-level heuristics
- Isolate, clean and prevent future reoccurrence of infections
- Thorough detection – Codeword searches the computer’s registry, hard drives and removable media, and live system memory for evidence of infection
- Receive **usable** alerts and data – collect all relevant evidence, along with meaningful log files and summary reports, and ships those back to you over a reporting method of your choice.
- Real-time, remote analysis – connect to agents over encrypted tunnel
Benefits and other uses

- Can be used on a regular basis as part of a network security best practice
- Use as a triage tool (e.g., in support of RETRI)
- Aggregate information on all system infections by site name and location
- Help find original infection point: All malware and system information, including pinpointing USB devices, is reported back
Codeword is not a “Forensically-sound” tool
It will not solve all of your problems
You should use Codeword as part of an overarching response process, not as The Easy Button
Codeword is beta freeware – don’t complain when it crashes
Comes with no warranties or hypno-toads
Components

- Codeword has 3 primary components:
  - **Admin Console (C#)**: A graphical interface used to generate new agents and connect to existing deployed agents; wraps agent binary in an MSI installer file for deployment
  - **Agent (C#)**: A single binary contained inside the generated MSI; a host-level scanner to detect viruses, clean related files and footprints, and to implement remediation actions to prevent further infection
  - **Kernel-mode driver (C)**: A single SYS file that contains rootkit detection logic and other evidence-collecting code
Quick start: using Codeword

1. Create an agent
   - Define signatures specific to malware
   - Choose user mode and kernel mode heuristics
   - Generate agent MSI installer
   - Deploy using psexec, sms, altiris, etc.

2. Connect/scan/analyze
   - Fire-and-forget mode: agent automatically sends an encrypted zip archive with results/evidence
   - Enterprise/Remote Control: use Admin Console

3. Collect/Mitigate
Admin Console
Step 1: Create an agent
Once the agent has unpacked, what would you like it to do?

- **Fire-and-Forget mode**
  The agent will unpack, run the scan, report back, and remove itself.

- **Remote control mode**
  The agent will unpack and open a listening port for commands.

- **Enterprise mode**
  The agent will unpack, run the scan and open a listening port for commands.
Connection

![Connection Settings](image)

**Agent service**
- Listening port: 41014
- Use random port number

**Authentication**
- Agent’s private/public key pair in PFX/PKCS #12 format:
  - Keystore file:
  - Password:
- Force strong authentication (AES-256 only)*
- Authenticate server to client
- Authenticate client to server
- Enforce certificate issuer:

*Note: AES-256 is only supported after WinXP SP3
Persistence/Stealth

**Persistence**
How long should the agent remain on the system?

- **Install as a service**
  - The agent will remain on the system until an administrator removes it.
  - Service name: CwAgent
  - *Installs to system folder

- **Run once**
  - The agent will destroy itself after completing the given tasks.

**Stealth**
How should the agent keep its presence secret?

- Randomize the name of the agent’s process
- Hide the agent’s process
- Do not attempt to install .NET
- Load driver using system load and call image
- Load driver using ZwLoadDriver()
## Reporting

### Configuration Options

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
<th>Example/Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Send results to</td>
<td>Enable automated reporting</td>
<td></td>
</tr>
<tr>
<td>Network share</td>
<td>example: \CorpShare\ScanResults$</td>
<td></td>
</tr>
<tr>
<td>FTP Server</td>
<td></td>
<td></td>
</tr>
<tr>
<td>E-mail</td>
<td>Address:</td>
<td></td>
</tr>
<tr>
<td>SMTP Server</td>
<td>port:</td>
<td></td>
</tr>
<tr>
<td>Web server URI</td>
<td>http(s):/</td>
<td></td>
</tr>
<tr>
<td>Confidentiality and Integrity</td>
<td>Use TLS/SSL [port: ]</td>
<td></td>
</tr>
<tr>
<td>Authentication</td>
<td>Application:</td>
<td></td>
</tr>
<tr>
<td>User name</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Password</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transport</td>
<td>Public Key (server):</td>
<td>Browse</td>
</tr>
<tr>
<td>Archive password</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Defining signatures

[Image of a software interface with options for Agent Settings, Signatures, Registry GUIDs, Registry, File, Memory, and Heuristics. The interface includes a panel for defining actions and keywords related to dynamic GUIDs.]
Selecting Heuristics
Generate it!
Step 2: Connect/Scan/Analyze Enterprise and Remote Control Modes
1. Specify admin console keys

2. Click connect!
..we are connected
The Toolbar

- Start a scan
- Update signature file
- Collect evidence
- Mitigate Findings
- Disconnect
- Uninstall agent
Issue a scan

- Click the big green “PLAY” button
- Issues a command to the agent to begin scanning with whatever signature file it has
- Scan as many times as you like; change signatures by uploading new signatures file
### Storm Worm Results: Registry

<table>
<thead>
<tr>
<th>Key Name</th>
<th>Value Name</th>
<th>Value Data</th>
<th>New Value Data</th>
<th>On Disk?</th>
<th>Act</th>
</tr>
</thead>
<tbody>
<tr>
<td>HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY\WINCOM32</td>
<td>NextInstance</td>
<td>1</td>
<td>False</td>
<td>False</td>
<td>Del</td>
</tr>
<tr>
<td>HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY\WINCOM32</td>
<td>Type</td>
<td>1</td>
<td>False</td>
<td>False</td>
<td>Del</td>
</tr>
<tr>
<td>HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY\WINCOM32</td>
<td>Start</td>
<td>2</td>
<td>False</td>
<td>False</td>
<td>Del</td>
</tr>
<tr>
<td>HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY\WINCOM32</td>
<td>ErrorControl</td>
<td>1</td>
<td>False</td>
<td>False</td>
<td>Del</td>
</tr>
<tr>
<td>HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY\WINCOM32</td>
<td>ImagePath</td>
<td>??\C:\WINDOWS\system32\winco...</td>
<td>False</td>
<td>Del</td>
<td></td>
</tr>
<tr>
<td>HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY\WINCOM32</td>
<td>DisplayName</td>
<td>wincom32</td>
<td>False</td>
<td>False</td>
<td>Del</td>
</tr>
<tr>
<td>HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY\WINCOM32</td>
<td>NextInstance</td>
<td>1</td>
<td>False</td>
<td>False</td>
<td>Del</td>
</tr>
<tr>
<td>HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY\WINCOM32</td>
<td>Type</td>
<td>1</td>
<td>False</td>
<td>False</td>
<td>Del</td>
</tr>
<tr>
<td>HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY\WINCOM32</td>
<td>Start</td>
<td>2</td>
<td>False</td>
<td>False</td>
<td>Del</td>
</tr>
<tr>
<td>HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY\WINCOM32</td>
<td>ErrorControl</td>
<td>1</td>
<td>False</td>
<td>False</td>
<td>Del</td>
</tr>
<tr>
<td>HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY\WINCOM32</td>
<td>ImagePath</td>
<td>??\C:\WINDOWS\system32\winco...</td>
<td>False</td>
<td>Del</td>
<td></td>
</tr>
<tr>
<td>HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY\WINCOM32</td>
<td>DisplayName</td>
<td>wincom32</td>
<td>False</td>
<td>False</td>
<td>Del</td>
</tr>
</tbody>
</table>

**Command History**

1. **initialize**: Loading scan settings...
2. **initialize**: Starting...
3. SCAN: Loading signatures from XML file...
4. **initialize**: Successfully turned OFF .NET security.
5. SCAN: Scan starting on 07/08/2003 21:33:22

---

**Signature Scan**

SCAN: Scanning registry for infections...
SCAN: Loading INI USER DAT files into HKKEY_USERS...
SCAN: Using home: HKLM...
SCAN: Scanning for signature "HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY\WINCOM32"...
SCAN: Signature matched on host!
NextInstance = "1" (0x1)
SCAN: Using home: HKLM...
SCAN: Scanning for signature "HKLM\SYSTEM\CurrentControlSet\Enum\Root\LEGACY\WINCOM32"...
SCAN: Signature matched on host!
## Storm Worm Results: File

**Recent Agents**

<table>
<thead>
<tr>
<th>Name</th>
<th>Path</th>
<th>Size</th>
<th>Hash</th>
<th>PE Signature</th>
<th>Created</th>
<th>Accessed</th>
</tr>
</thead>
<tbody>
<tr>
<td>peers.ini</td>
<td>C:\WINDOWS\system32\peers.ini</td>
<td>5483</td>
<td>4491E53D931605FBA4F5D0605E198EB</td>
<td>None</td>
<td>Wednesday, July 08, 2009</td>
<td>Wednesday, J1</td>
</tr>
<tr>
<td>wincom32.sys</td>
<td>C:\WINDOWS\system32\wincom32.sys</td>
<td>41728</td>
<td>A76A0CD2517A32D4CA5E3D0682E1F3C</td>
<td>None</td>
<td>Wednesday, July 08, 2009</td>
<td>Wednesday, J1</td>
</tr>
</tbody>
</table>

**Command History**

- **INITIALIZE:** Loading scan settings...
- **INITIALIZE:** Success.
- **SCAN:** Loading signatures from XML file.
- **INITIALIZE:** Successfully turned OFF .NET security.
- **SCAN:** Scan starting on 07/08/2009 21:58:50

---

**SIGNATURE SCAN**

- **SCAN:** Scanning for infections.
- **SCAN:** Loading NTUSER.DAT files into HKEY_USERS...
- **SCAN:** Scanning for signature \HKLM\SYSTEM\ControlSet001\Enum\Legacy_Wincom32\...
- **SCAN:** Signature matched on host!
- **SCAN:** Next instance = 11
- **SCAN:** Using hive \HKLM
- **SCAN:** Scanning for signature \HKLM\SYSTEM\ControlSet001\Services\wincom32\...
- **SCAN:** Signature matched on host!
Step 3: Collect and Mitigate Enterprise and Remote Control Modes
Collect
Mitigate
## Mitigate (2)

<table>
<thead>
<tr>
<th>Name</th>
<th>Path</th>
<th>Size</th>
<th>Hash</th>
</tr>
</thead>
<tbody>
<tr>
<td>peers.ini</td>
<td>C:\WINDOWS\system32\peers.ini</td>
<td>5483</td>
<td>44015E530931605F8A4F5DD609E19BE8B</td>
</tr>
<tr>
<td>wincom32.sys</td>
<td>C:\WINDOWS\system32\wincom32.sys</td>
<td>41728</td>
<td>A76A0CD2517A38204CA5E93D0B2E4F3C</td>
</tr>
</tbody>
</table>
Fire-and-forget Mode
What’s reported?

- A password-protected, encrypted (AES 256) Zip archive containing:
  - Infection summary report
  - Mitigation report
  - All collected malware binaries and evidence
  - A detailed run log
Video Demos
Demo #1: Storm Worm

GOAL:
- Understand how to define registry, disk and memory signatures to detect user-mode malware

SCENARIO:
- VM Guest infected with Storm worm

OBJECTIVES:
- Deploy agent using Remote Control mode
- Examine malware footprints
Demo #2: TcpIrpHook

GOAL:
- Understand how Codeword heuristics help catch kernel malware (and anti-virus)

SCENARIO:
- VM Guest infected with kernel-mode rootkit TcpIrpHook

OBJECTIVES:
- Deploy agent using Remote Control mode
- Scan with Driver IRP hook heuristic
Conclusions
Possible Limitations

- Software licensing costs can be prohibitive
  - These costs are outweighed by user productivity
  - “renting” the software may be a cost-effective solution
- Some challenges that plague traditional methods also impact RETRI:
  - Disorganized networks, lack of funding, lack of mgmt-level support, lack of resources, etc.
  - Assumptions made early on have cumulative impact later on:
    - Availability of backups
    - COOP readiness
    - Date and scope of infection
Final Thoughts

- Preparation is key to ensuring services are restored quickly
  - Know your network and critical services
  - Ensure backups exist
  - Have hardware / software ready
- Keeping services up significantly reduces the cost of recovery
- Remember: User downtime costs 3 times as much as the actual cleanup
Thanks for coming!!

Email us
Mike.A.Murphy@gmail.com
AaronLemasters@yahoo.com

Website:
www.hexsec.com
www.code-word.org