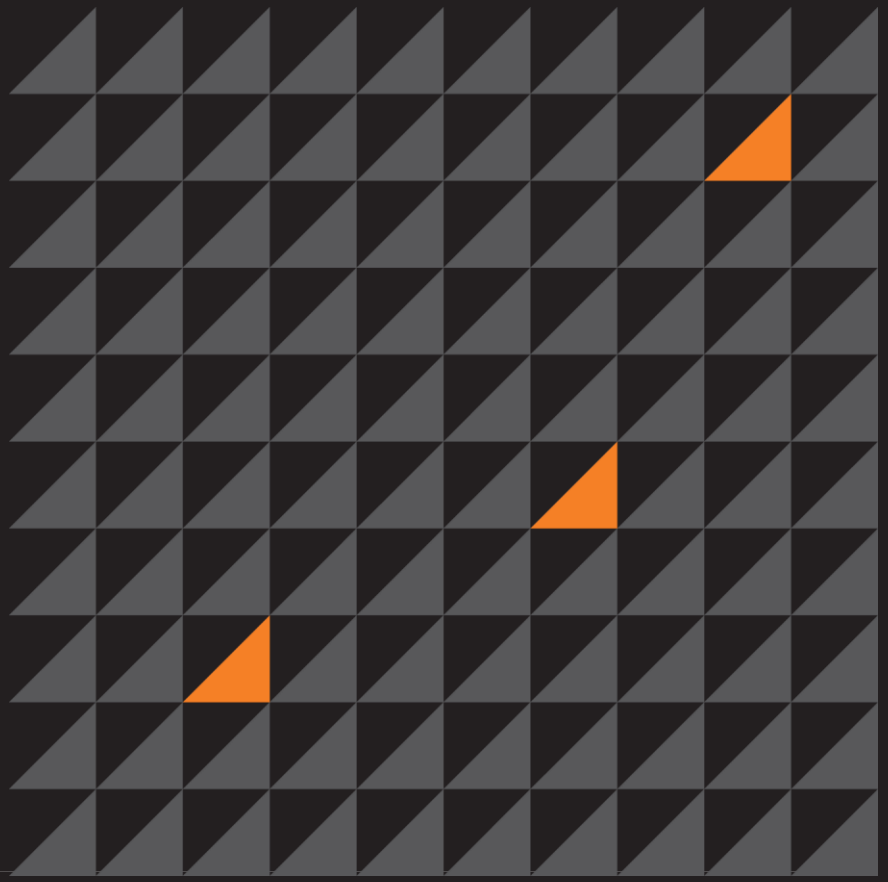
The Pageantry of Lateral Movement

Stuart Morgan

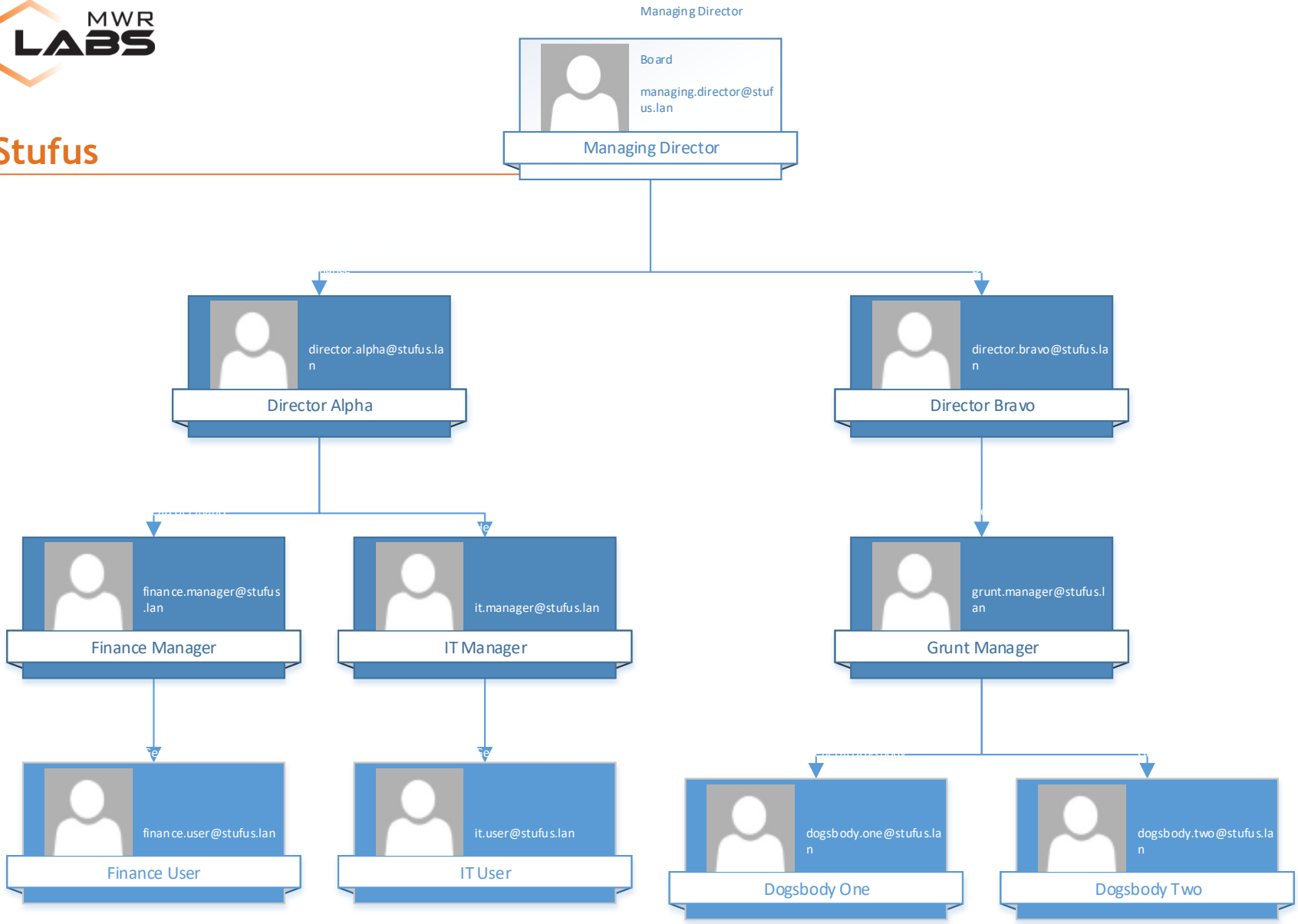
14th January 2016



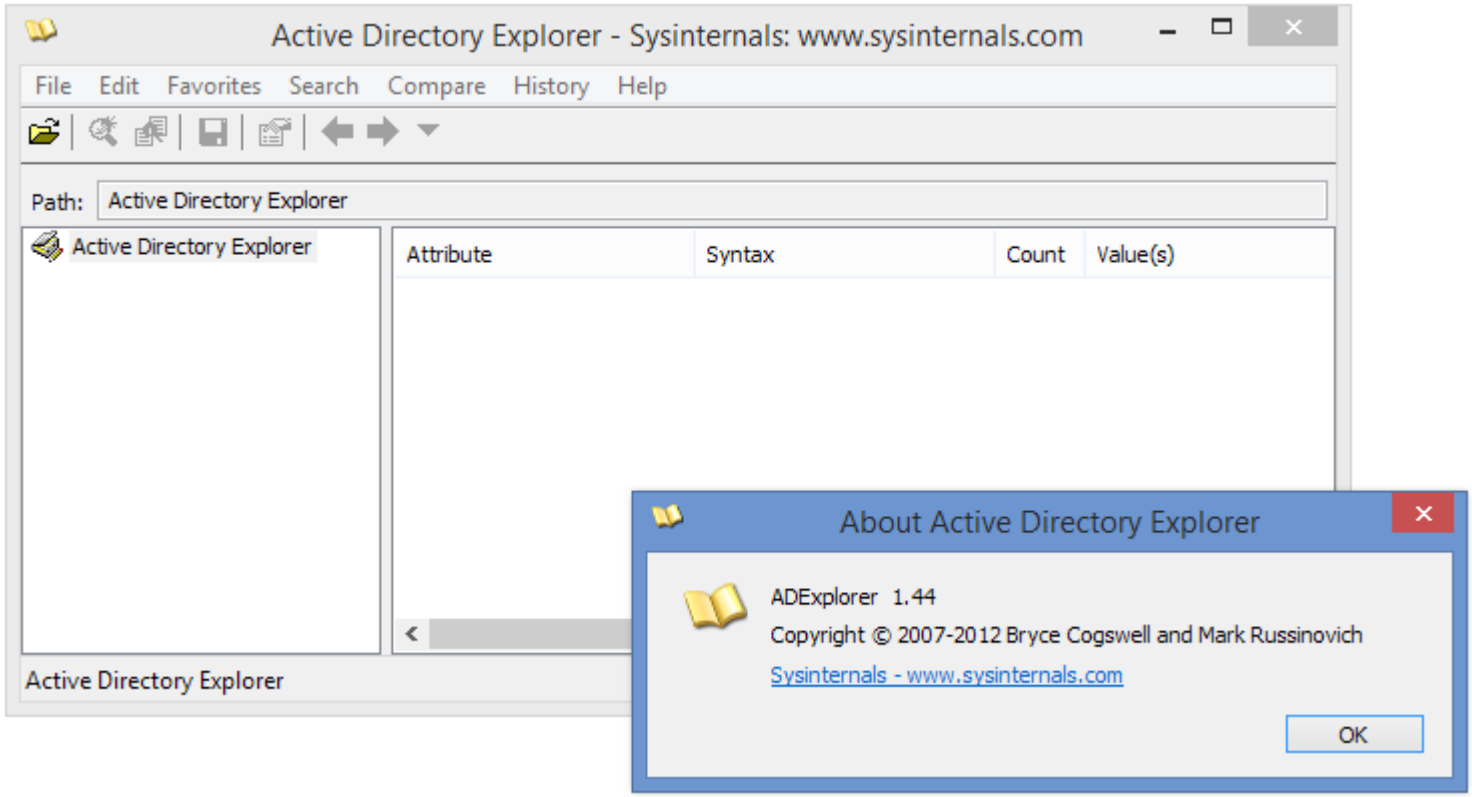
Introduction

- Gained a foothold on 'STUFUS'
  - Phishing e-mail → Meterpreter RAT
  - Username: STUFUS\it.user
  - Password: Pa$$w0rd1

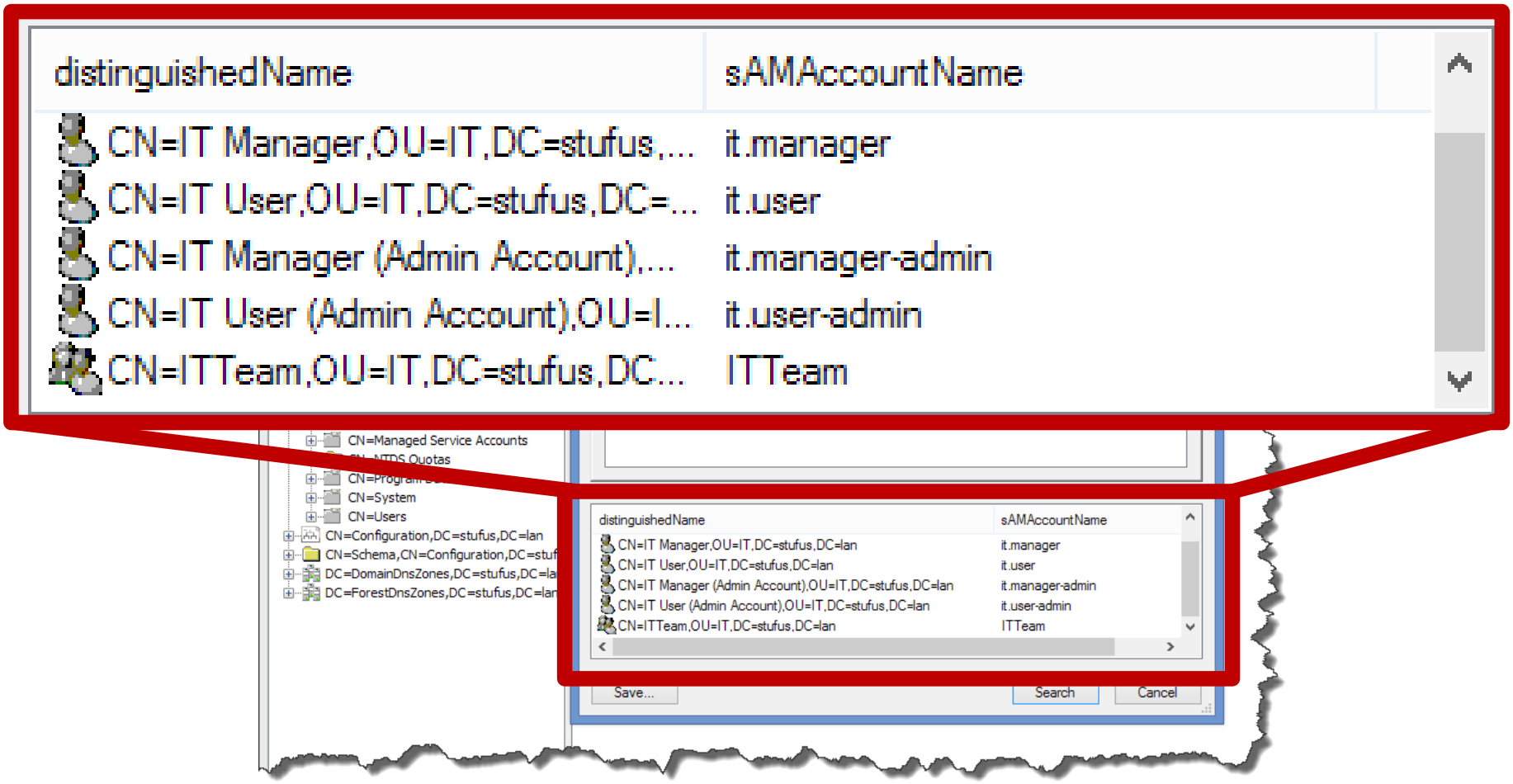
Stufus


Managing Director
Board
managing.director@stufus.lan
Managing Director
director.alpha@stufus.lan
Director Alpha
director.bravo@stufus.lan
Director Bravo
finance.manager@stufus.lan
Finance Manager
it.manager@stufus.lan
IT Manager
grunt.manager@stufus.lan
Grunt Manager
finance.user@stufus.lan
Finance User
it.user@stufus.lan
IT User
dogsbody.one@stufus.lan
Dogsbody One
dogsbody.two@stufus.lan
Dogsbody Two

Situational Awareness: AD Explorer



Situational Awareness: AD Explorer



Windows Messages

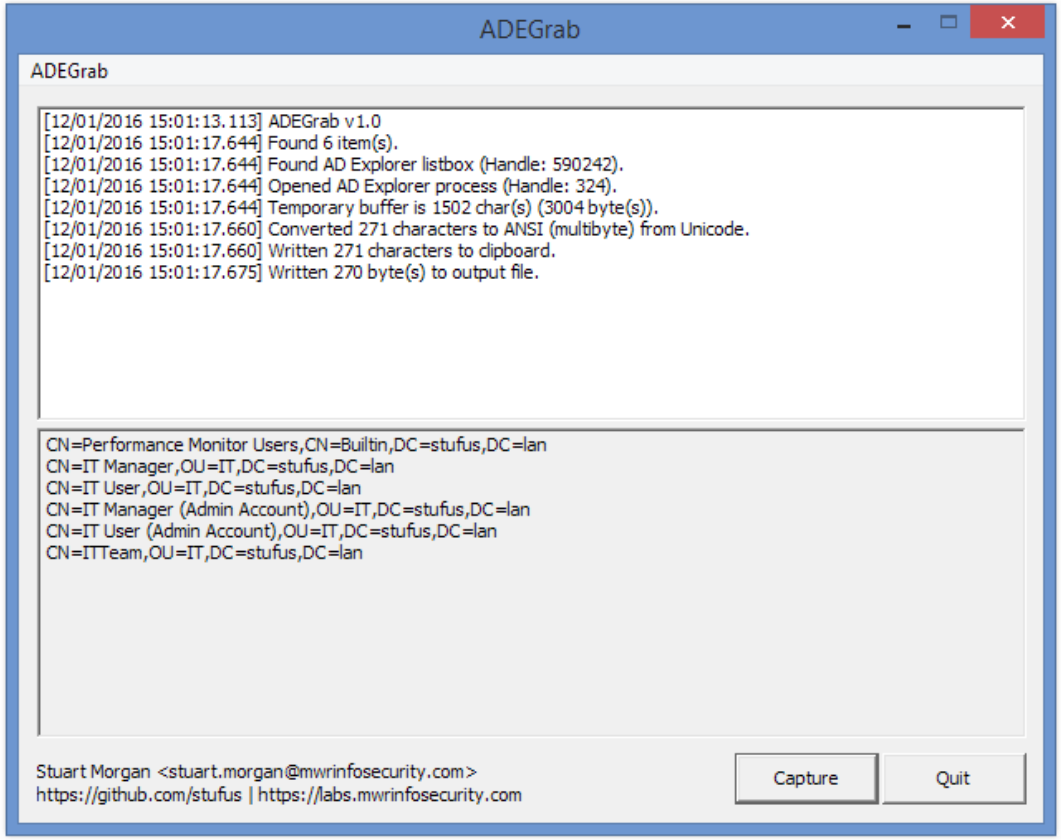
#1: Hook AD Explorer Messages

#2: Interact with AD Explorer

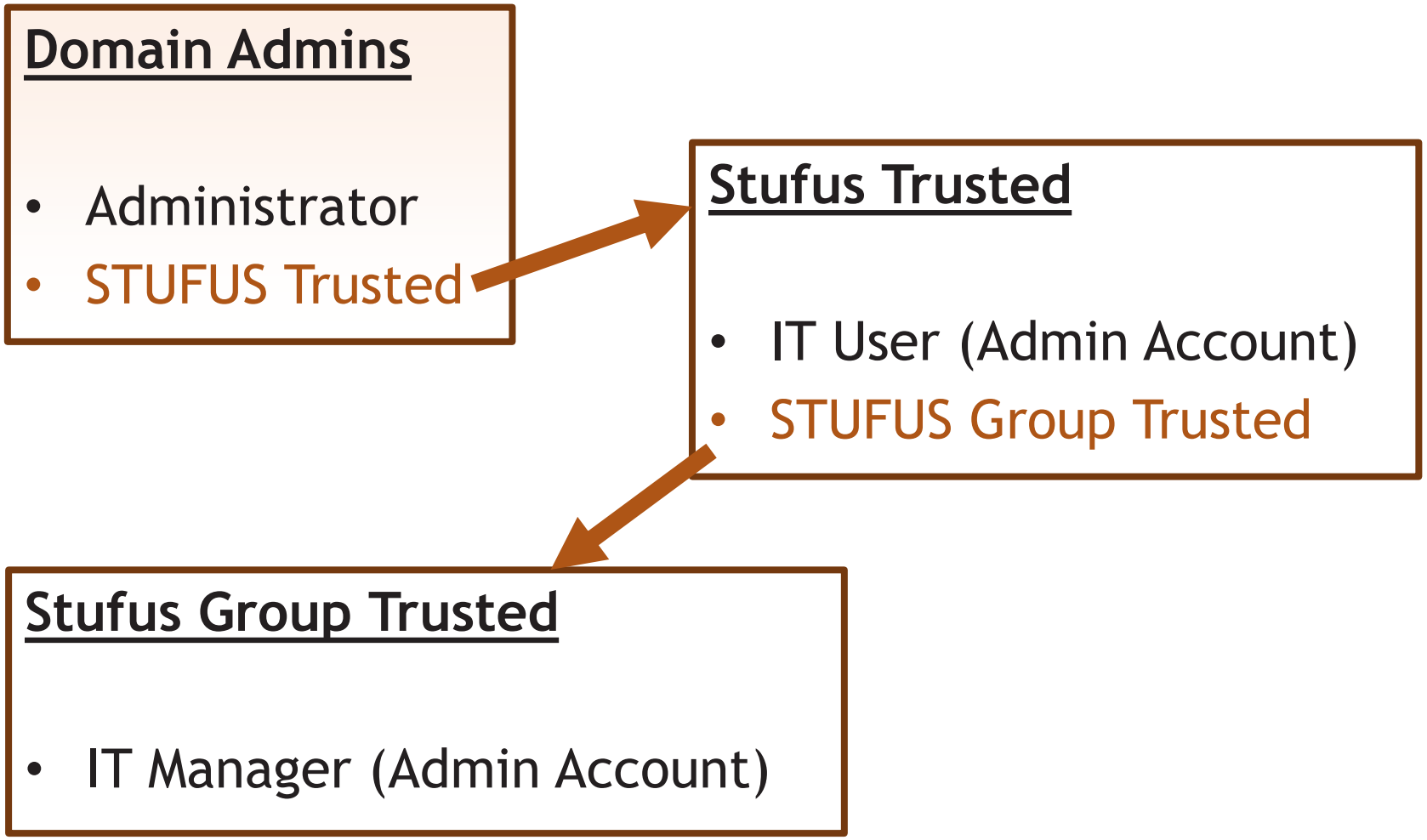
1. Find the process handle of AD Explorer and the handle of the Search Results box.
2. Send LVM_GETITEMCOUNT to the Search Results box to get the number of items.
3. For each item (0 to LVM_GETITEMCOUNT-1)
   a. Allocate a block of memory inside AD Explorer's memory space.
   b. Send LVM_GETITEM to the Search Results box, pointing at the above block.
   c. Copy the block of memory back to our process.
   d. Convert from Unicode, copy to clipboard etc…

ADEGrab

https://github.com/stufus/adegrab



Nested Active Directory Groups



Nested Active Directory Groups

- ‘net group’
- ADSI/LDAP queries
- GUI (e.g. AD Explorer)

Nested Active Directory Groups

1.2.840.113556.1.4.1941	**LDAP_MATCHING_RULE_IN_CHAIN**	This rule is limited to filters that apply to the DN. This is a special "extended match operator that walks the chain of ancestry in objects all the way to the root until it finds a match.

		equivalent to a bitwise **OR** operator.
Searching Binary Data	1.2.840.113556.1.4.1941	**LDAP_MATCHING_RULE_IN_CHAIN**	This rule is limited to filters that apply to the DN. This is a special "extended match operator that walks the chain of ancestry in objects all the way to the root until it finds a match.
Distributed Query			

The following example query string searches for group objects that have the **ADS_GROUP_TYPE_SECURITY_ENABLED** flag set. Be aware that the decimal value of

Nested Active Directory Groups

```
(&
 (objectClass=user)
 (memberof:1.2.840.113556.1.4.1941:=CN=Domain Admins,CN=Users,DC=stufus,DC=lan)
)
```

Nested Active Directory Groups

Meterpreter

- adsi_nested_group_user_enum
- adsi_group_enum

POST Modules

- post/windows/gather/enum_ad_groups
- post/windows/gather/enum_ad_users

https://github.com/rapid7/metasploit-framework/pull/5895

https://labs.mwrinfosecurity.com/blog/2015/09/30/active-directory-users-in-nested-groups-reconnaissance/

Active Directory to Local SQLite DB

1. List all of the groups in Active Directory and store in a SQLite database.
2. For each group, list the users and specify LDAP_MATCHING_RULE_IN_CHAIN. Store the users in the database and inject into a table linking users to groups.
3. List all of the computers in Active Directory and store in the database.

Active Directory to Local SQLite DB

ad_groups

RID	Name
1000	Group 1
1001	Group 2
1002	Group 3

ad_users

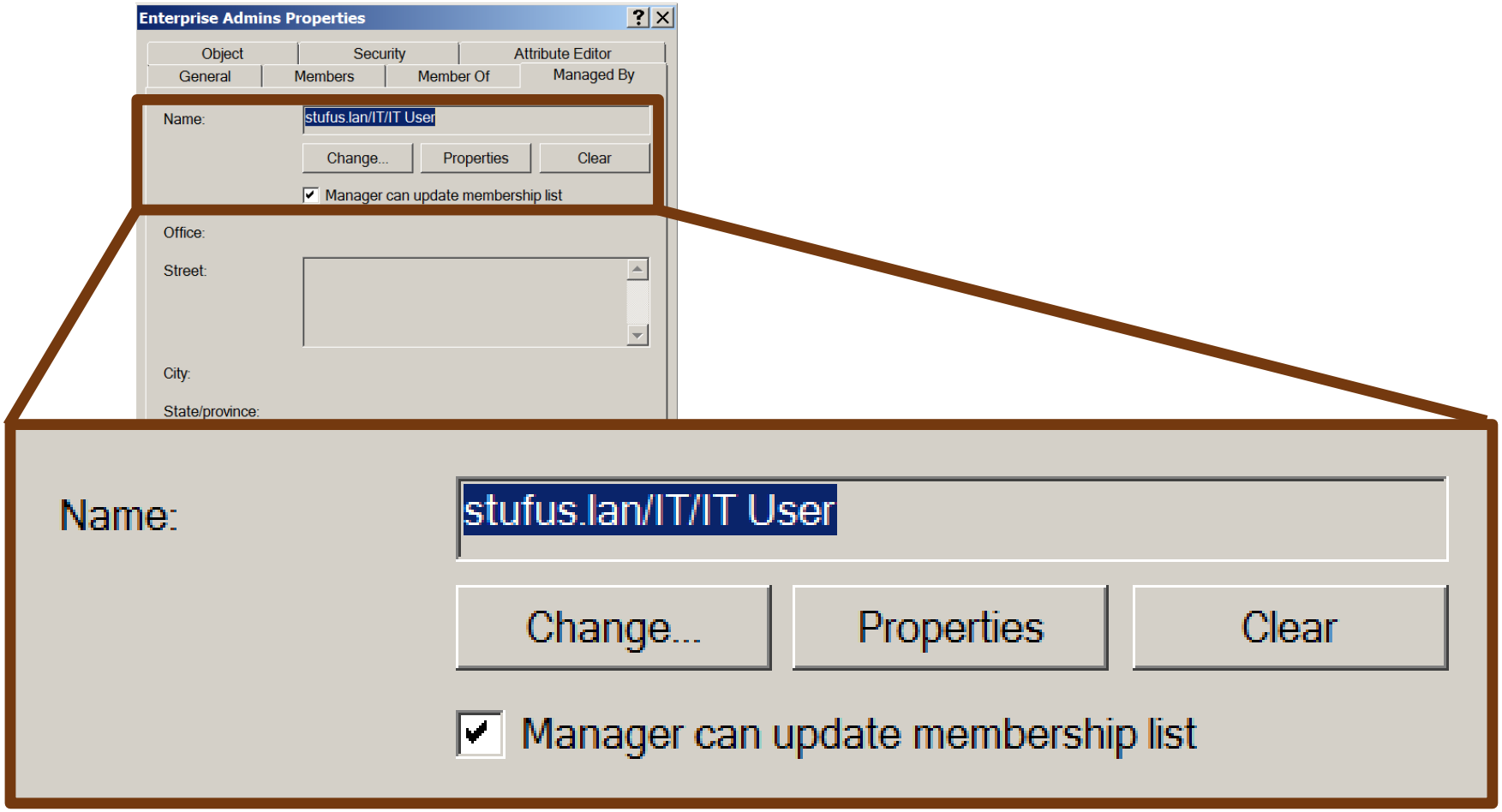
RID	Name
2000	User 1
2001	User 2
2002	User 3

ad_mapping

UserID	GroupID
2000	1000
2001	1002
2002	1002

https://github.com/rapid7/metasploit-framework/pull/6378

But why is *this* possible?



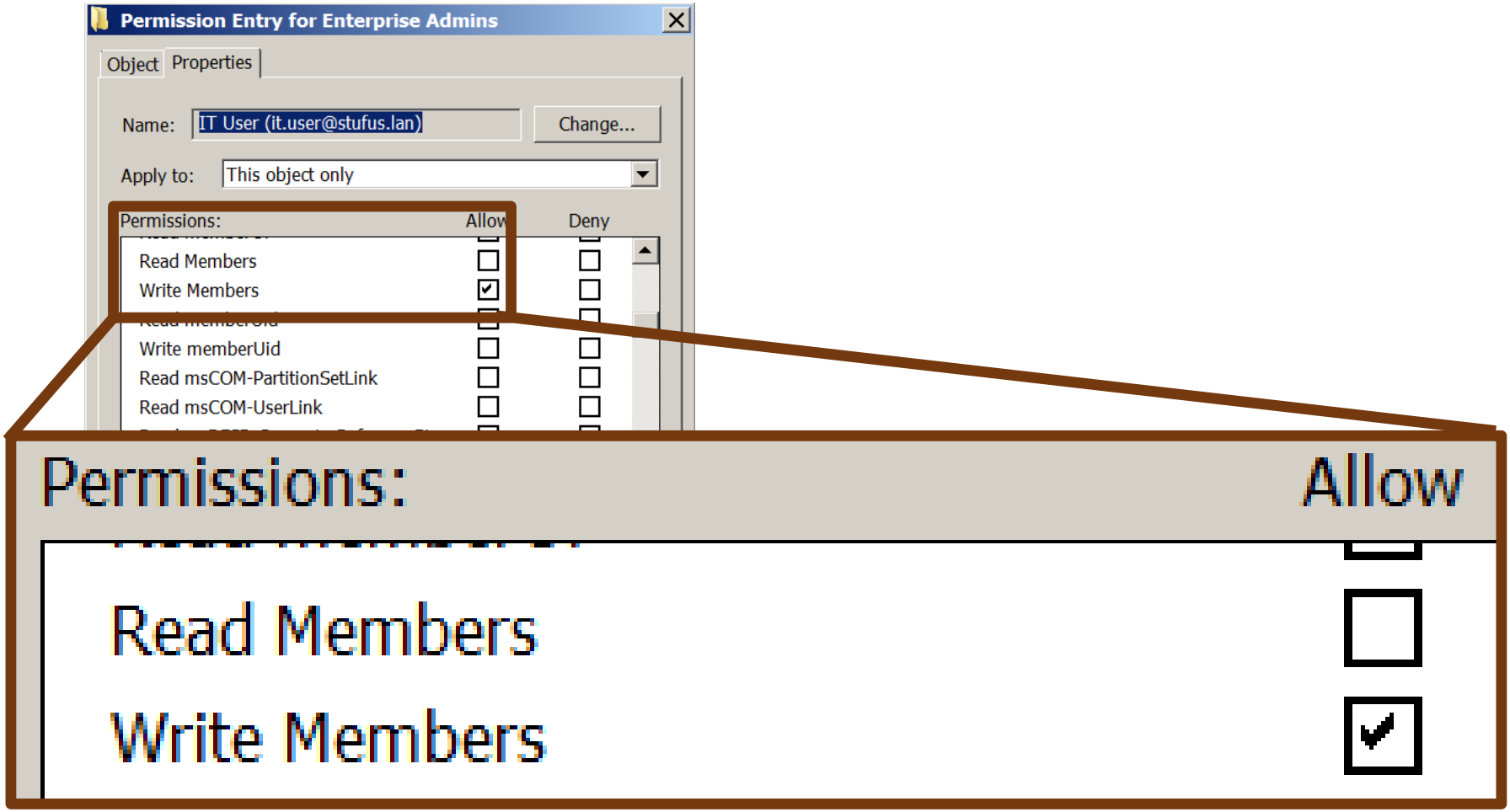
But why is *this* possible?

https://github.com/PowerShellMafia/PowerSploit/pull/105
https://github.com/rapid7/metasploit-framework/pull/6375
https://github.com/PowerShellEmpire/Empire/pull/119

```
PS C:\Users\it.user\Documents> Find-ManagedSecurityGroups


GroupDN           : CN=Privileged,CN=Users,DC=stufus,DC=lan
ManagerDN         : CN=IT User,OU=IT,DC=stufus,DC=lan
ManagerCN         : IT User
ManagerType       : User
GroupCN           : Privileged
CanManagerWrite   : True
ManagerSAN        : it.user
```

But why is *this* possible?



Defences

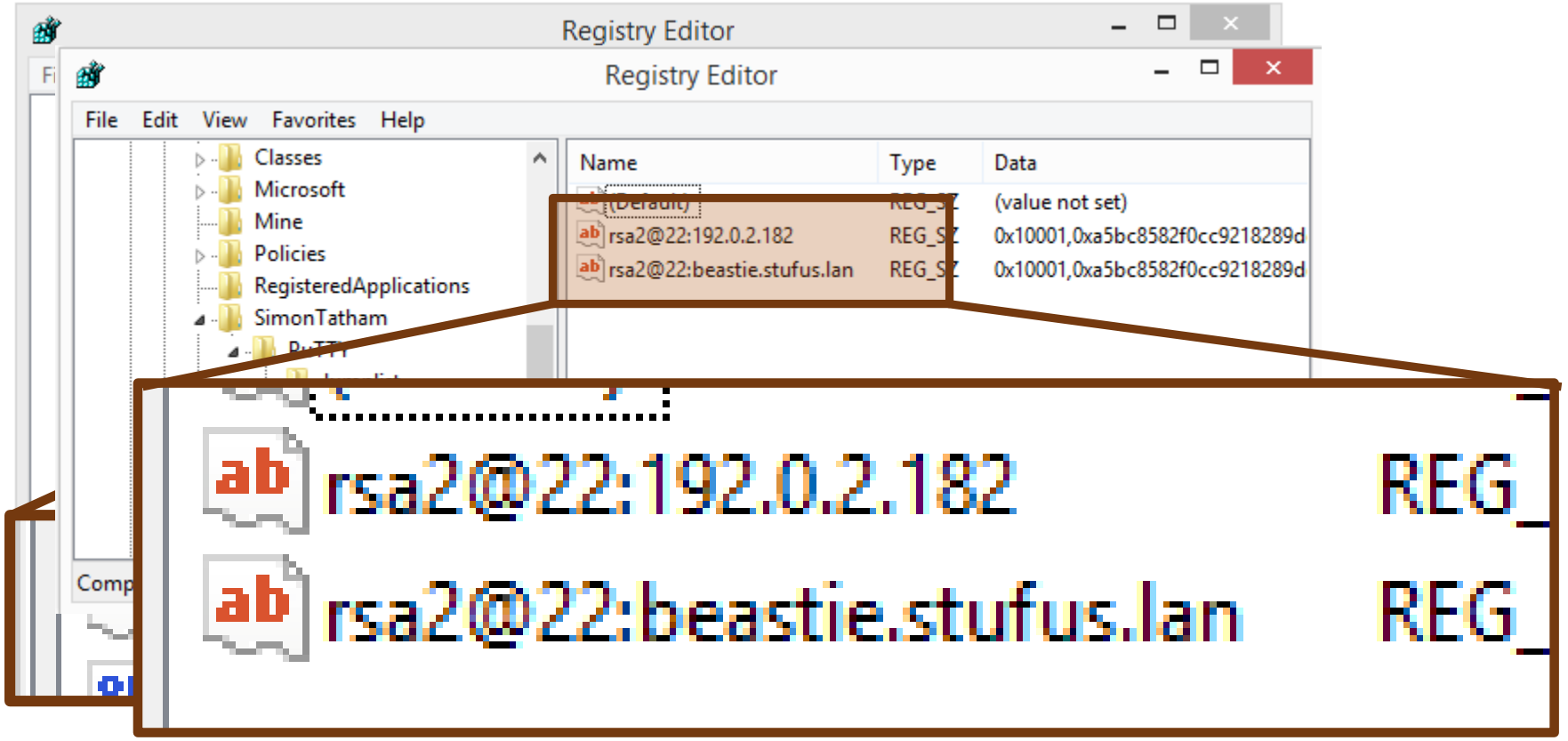
https://technet.microsoft.com/en-us/magazine/2009.09.sdadminholder.aspx

- Each domain has an 'AdminSDHolder' object.
- Each hour, a process runs on a domain controller which checks the ACLs of protected groups.
- If they are different, it overwrites the security ACL.
- This has the effect of removing the ability to delegate privileges on protected groups automatically…..

….but it doesn't apply to 'non-protected' groups….

SSH using PuTTY

PuTTY stores previous connections and saved sessions in the registry.



SSH using PuTTY

https://github.com/rapid7/metasploit-framework/pull/5359

msf> use post/windows/gather/enum_putty_saved_sessions

- Enumerate saved PuTTY sessions.
- Retrieve configured private keys.
- Detect usage of Pageant (an SSH agent).
- Retrieve hosts that PuTTY or Plink have previously connected to.
  - There is no interface to remove this….

SSH using Pageant

1. Load keys into the SSH Agent
2. PuTTY asks the SSH Agent to sign the challenge with key #1
3. PuTTY asks the SSH Agent to sign the challenge with key #2….#3….#4….#n

- PuTTY itself never sees the private key.
- The SSH agent never reveals the private key.

Pageant

- pageant.exe will sign any requests asked of it….
  …and is running in the background.

- putty.exe is running too as a separate process.

- Other tools such as FileZilla and WinSCP can also communicate with Pageant natively.

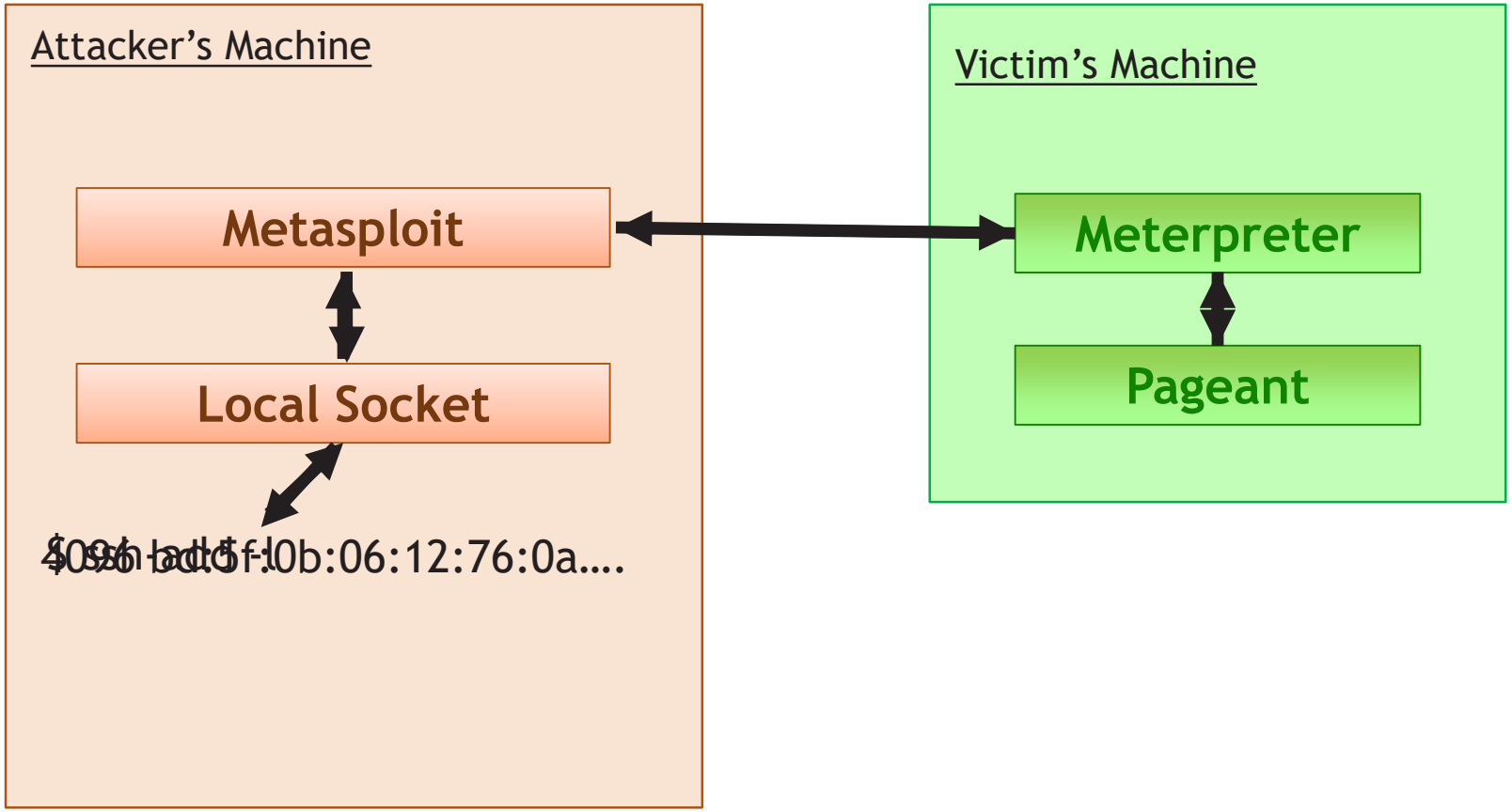
- They are different processes, with a different address space etc.

How does PuTTY 'talk' to the Agent?

1. PuTTY obtains the handle of the Pageant process (by looking for a window of class name 'Pageant').
2. PuTTY allocates a block of shared memory (8KB in size) with name 'PageantRequest<thread id>'.
3. PuTTY copies its request to the shared memory.
4. PuTTY sends the WM_COPYDATA message to Pageant with '0x804e50ba' and 'PageantRequest<thread id>'.
5. When the SendMessage() API call completes, the shared memory will be overwritten with the response to the original request.

Read https://raw.githubusercontent.com/openssh/openssh-portable/master/PROTOCOL.agent

PageantJacker

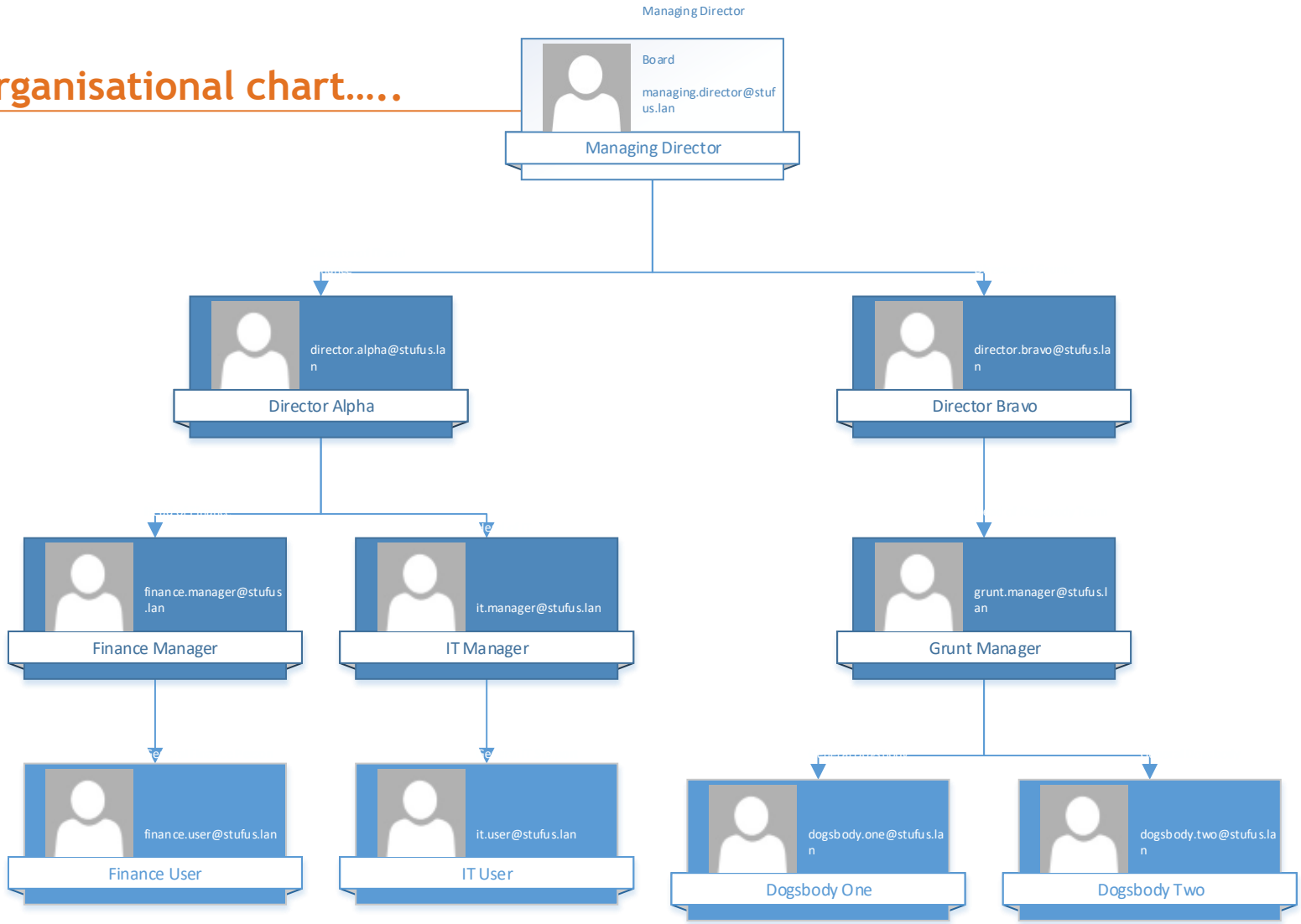


PageantJacker

https://github.com/rapid7/metasploit-framework/pull/5380
https://github.com/rapid7/meterpreter/pull/164
https://github.com/rapid7/metasploit-payloads/pull/29

```
msf> use post/windows/manage/forward_pageant
```

That organisational chart…..



That organisational chart…..

https://github.com/rapid7/metasploit-framework/pull/6377

msf> use post/windows/gather/make_csv_orgchart

---

cn,description,title,phone,department,division,e-mail,company,reports_to

"Director Alpha","","Director of IT and Finance","","","","director.alpha@stufus.lan","","Managing Director"

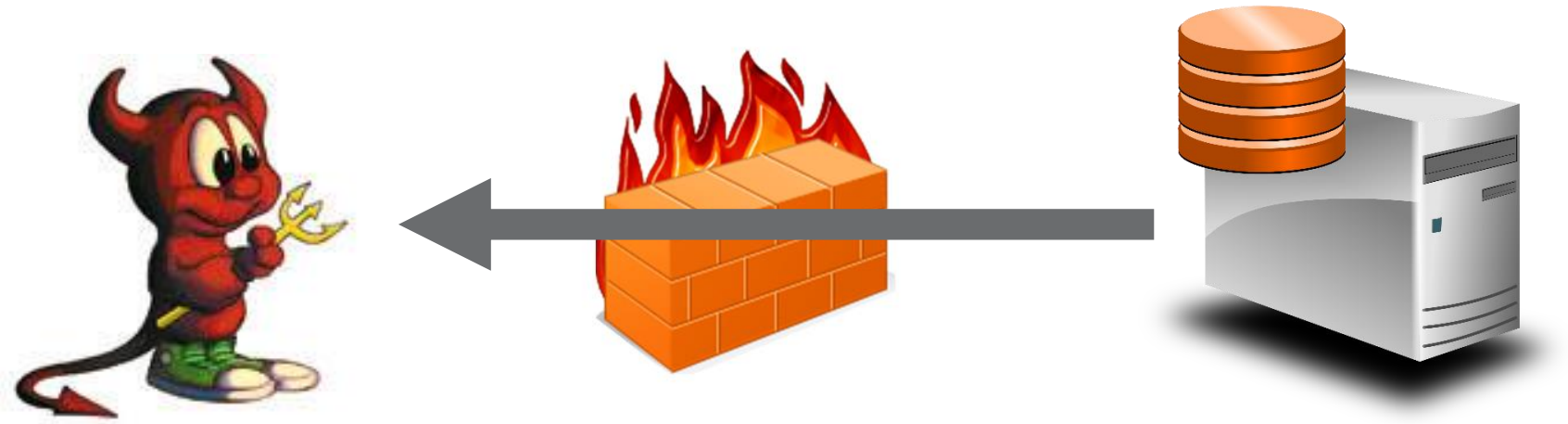
"Finance Manager","","Head of Finance","","","","finance.manager@stufus.lan","","Director Alpha"

"Finance User","","General Finance Person","","","","finance.user@stufus.lan","","Finance Manager"

…continued…

Egress Busting

- You have code execution but no interactive shell on a host….
- ….or you are just looking to check a firewall's effective configuration…..



Attacker | Various firewalls?? | Compromised host

Egress Bruteforcing

- Can handle both TCP and UDP.
- Can specify a range of ports (or all ports).
  - e.g. 22-25, 80, 33434-33534 etc.
- Does not require us to listen on 65535 ports.
- Does not require admin access on the victim's side (does not matter about our side).
- Supports Windows and UNIX-like operating systems.
- Lightweight.
- Ideally does not require separate binaries (could trip AV etc).
- Can be run from the command line or through a RAT.

Egress Bruteforcing

- A ‘framework’
  - You tell it what the destination IP address is, what ports to try, what protocol to use etc.
  - It generates code in the language of your choice.
  - You run that on the victim side….
- You then run **tcpdump** and sniff the incoming packets.
  - ….and then format them accordingly.

Egress Bruteforcing

https://github.com/stufus/egresscheck-framework

https://github.com/PowerShellEmpire/Empire/pull/117
https://github.com/rapid7/metasploit-framework/pull/6296

MWR LABS

Questions

https://labs.mwrinfosecurity.com/

https://github.com/stufus/

@ukstufus

stuart.morgan@mwrinfosecurity.com