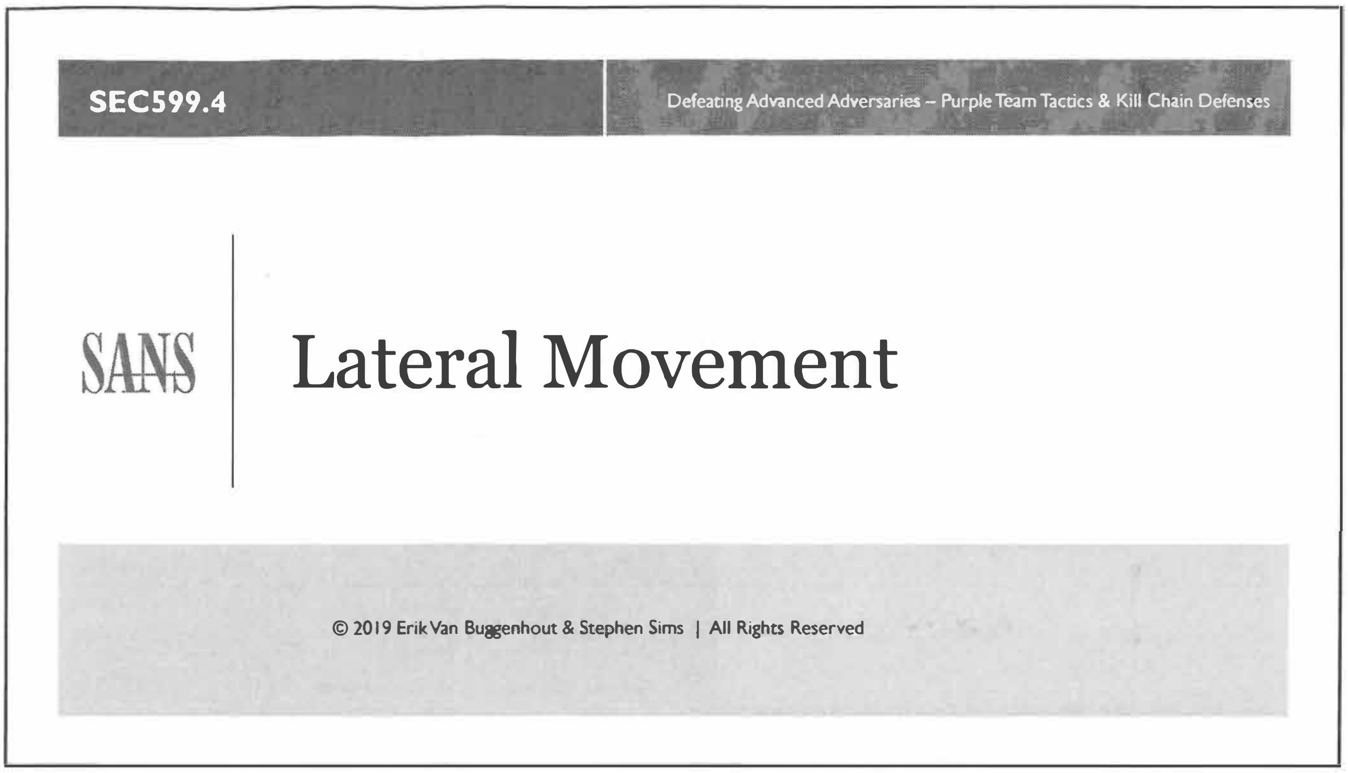
SEC599 | DEFEATING ADVANCED ADVERSARIES – PURPLE TEAM TACTICS & KILL CHAIN DEFENSES

599.4
Lateral Movement

SANS

THE MOST TRUSTED SOURCE FOR INFORMATION SECURITY TRAINING, CERTIFICATION, AND RESEARCH | sans.org



SEC599.4

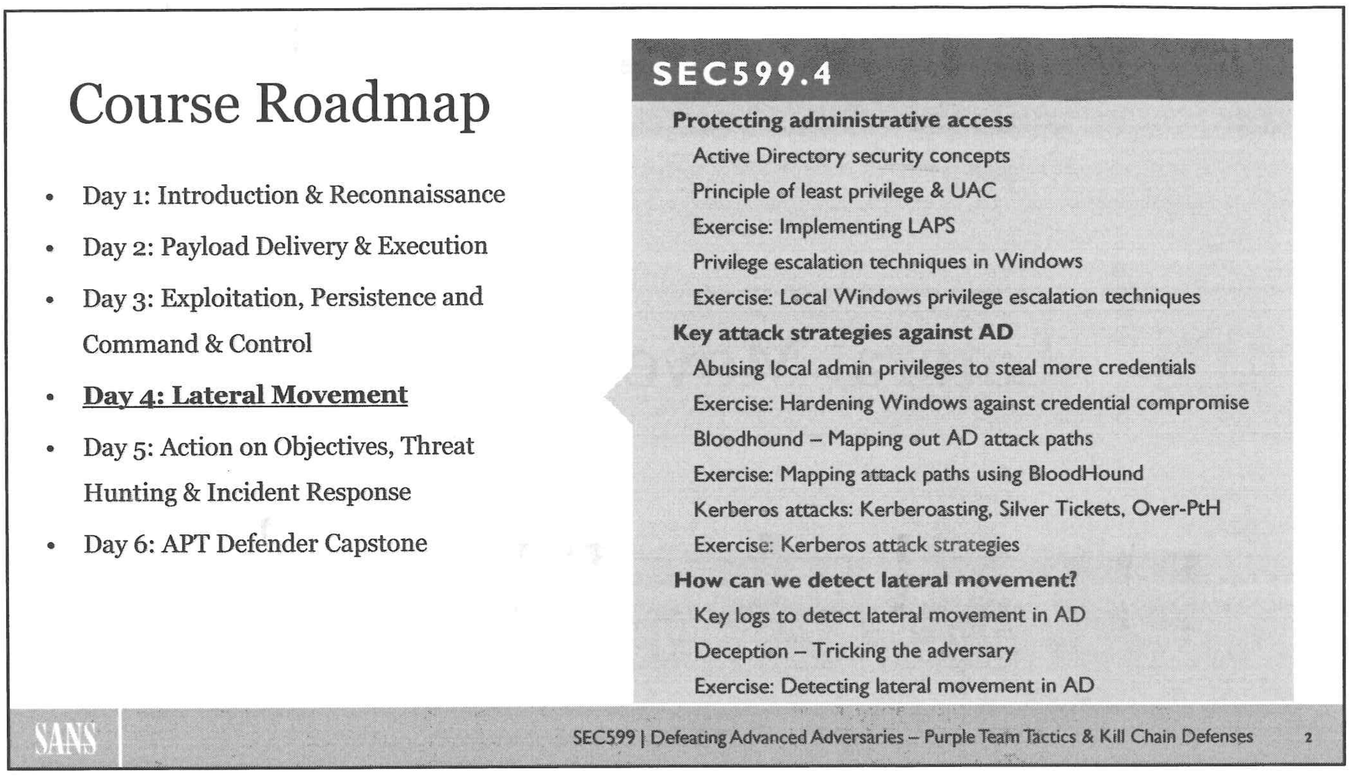
Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses

SANS

Lateral Movement

© 2019 Erik Van Buggenhout & Stephen Sims | All Rights Reserved

This page intentionally left blank.



Course Roadmap

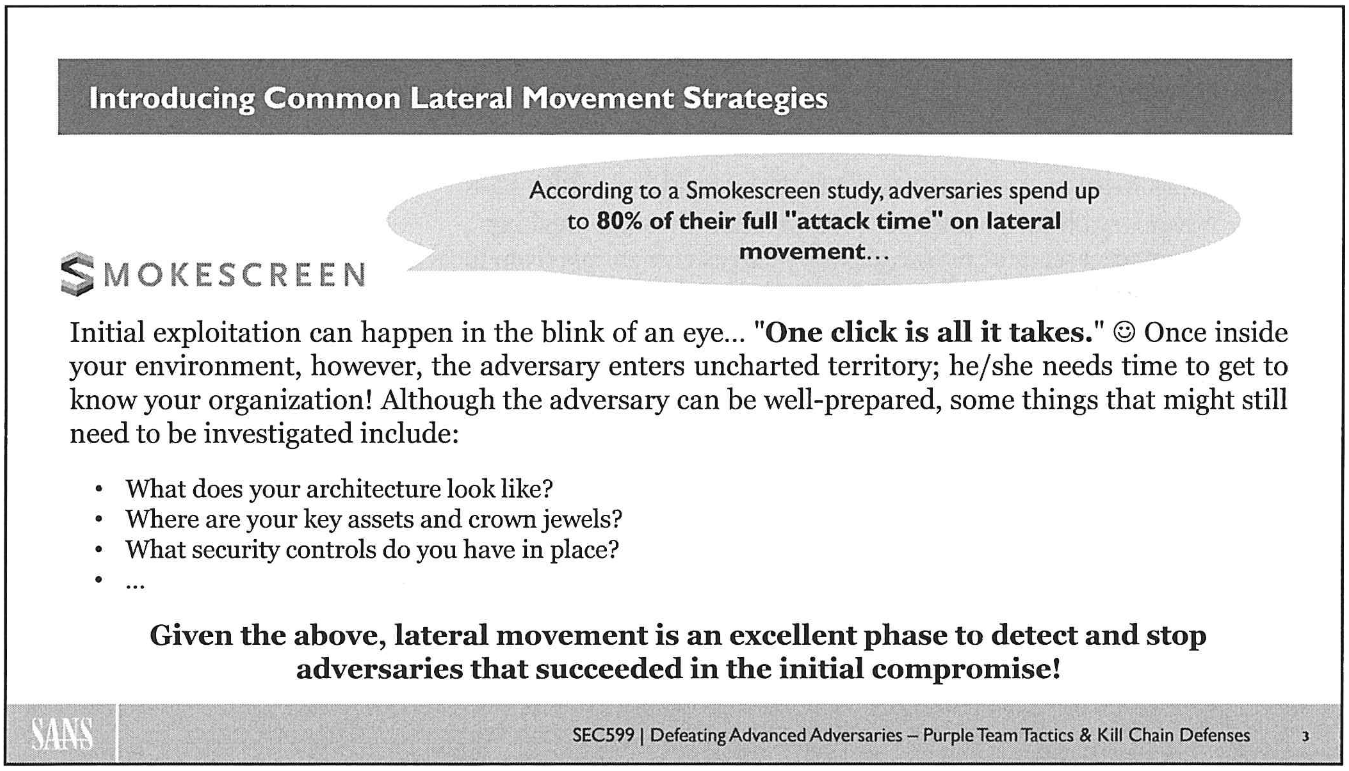
- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- **Day 4: Lateral Movement**
- Day 5: Action on Objectives, Threat Hunting & Incident Response
- Day 6: APT Defender Capstone

SEC599.4

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 2

This page intentionally left blank.

© 2019 Erik Van Buggenhout & Stephen Sims



Introducing Common Lateral Movement Strategies

According to a Smokescreen study, adversaries spend up to **80% of their full "attack time" on lateral movement...**

Initial exploitation can happen in the blink of an eye... "**One click is all it takes.**" ☺ Once inside your environment, however, the adversary enters uncharted territory; he/she needs time to get to know your organization! Although the adversary can be well-prepared, some things that might still need to be investigated include:

- What does your architecture look like?
- Where are your key assets and crown jewels?
- What security controls do you have in place?
- ...

Given the above, lateral movement is an excellent phase to detect and stop adversaries that succeeded in the initial compromise!

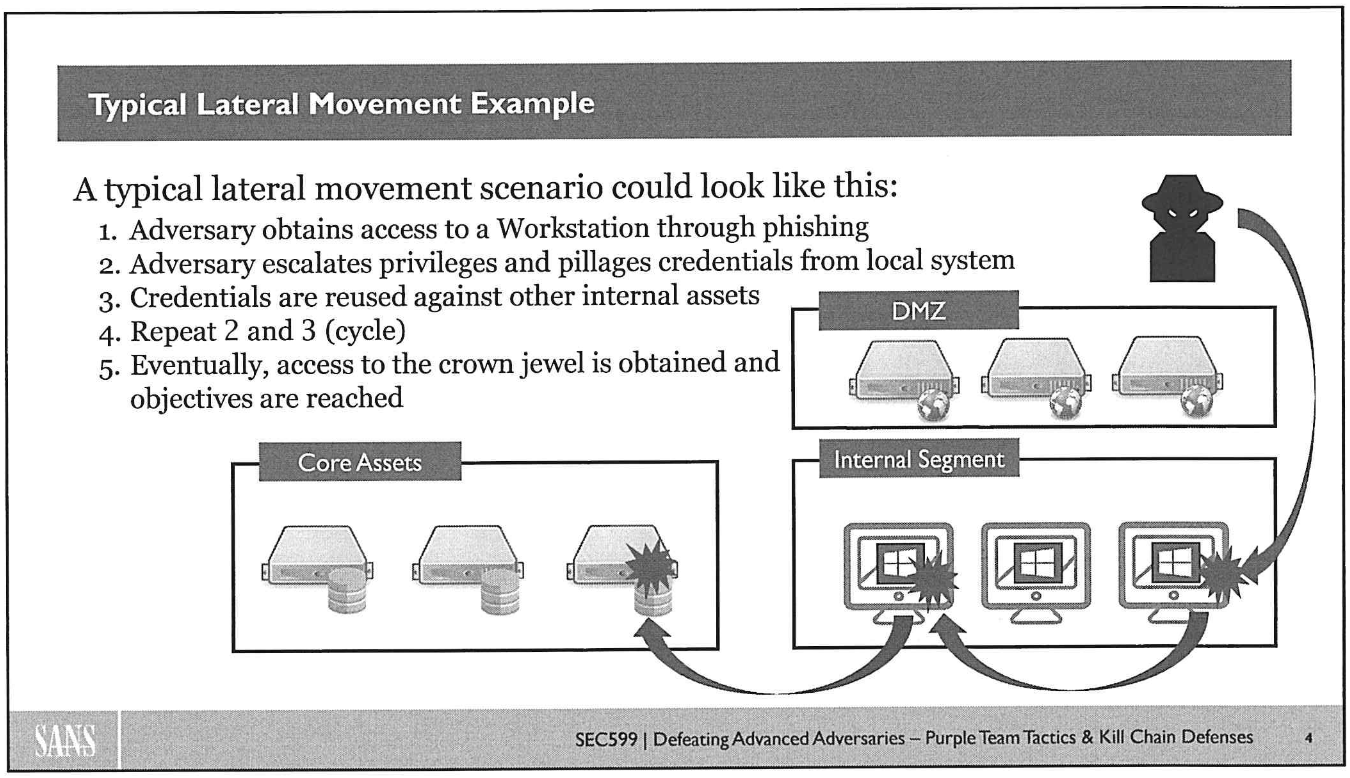
SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 3

Introducing Common Lateral Movement Strategies

Initial exploitation can happen in the blink of an eye... "**One click is all it takes.**" ☺ Once inside your environment, however, the adversary enters uncharted territory; he/she needs time to get to know your organization! Although the adversary can be well-prepared, some things that might still need to be investigated include:

- What does your architecture look like?
- Where are your key assets and crown jewels?
- What security controls do you have in place?
- ...

According to a Smokescreen study (a vendor of cyber deception technology), adversaries spend up to 80% of their full "attack time" on lateral movement! Given the above, lateral movement is an excellent phase to detect and stop adversaries that succeeded in the initial compromise!



Typical Lateral Movement Example

A typical lateral movement scenario could look like this:

1. Adversary obtains access to a Workstation through phishing
2. Adversary escalates privileges and pillages credentials from local system
3. Credentials are reused against other internal assets
4. Repeat 2 and 3 (cycle)
5. Eventually, access to the crown jewel is obtained and objectives are reached

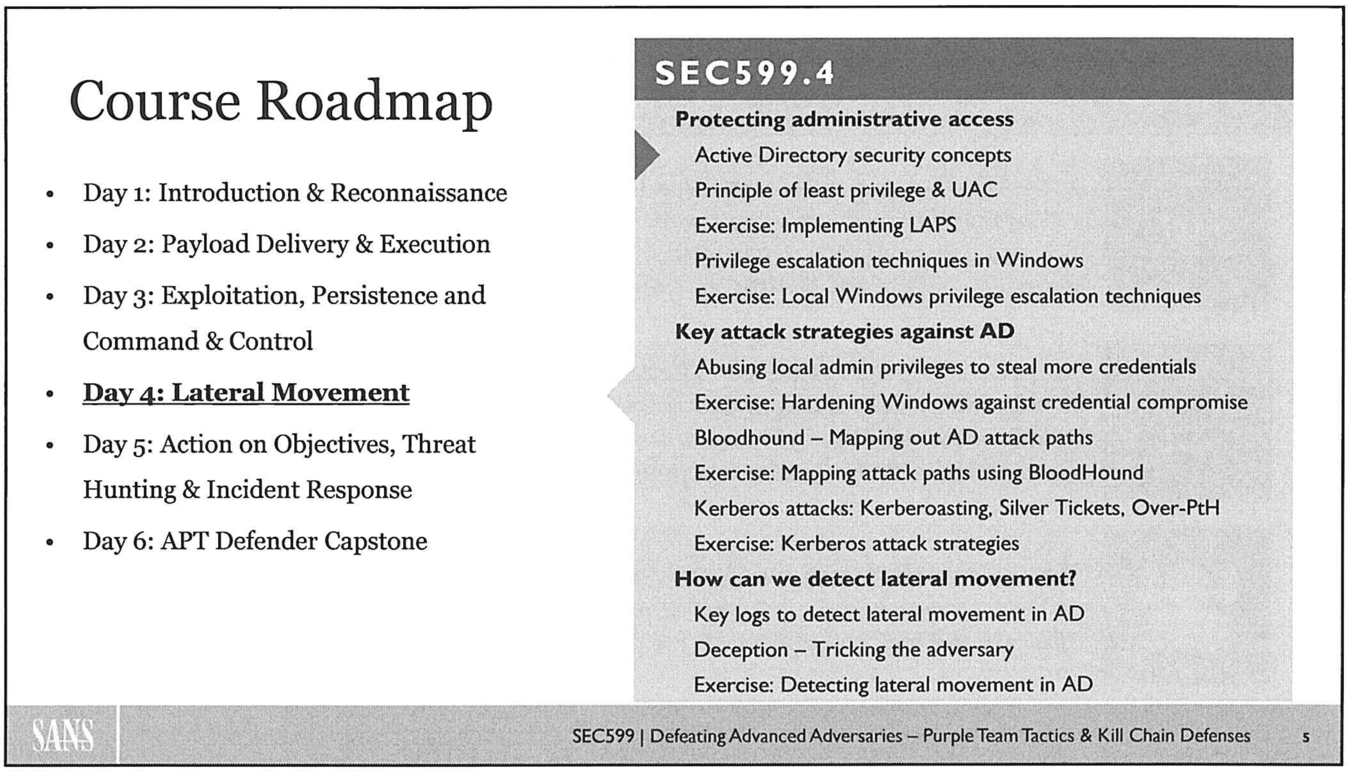
Typical Lateral Movement Example

So, what does a typical lateral movement scenario look like? It looks something like this:

- An adversary obtains initial access to the environment by compromising a workstation through phishing.
- The adversary escalates privileges on the machine and steals credentials from local system
- Credentials are reused against other internal assets.
- The adversary repeats the previous steps, thereby moving laterally between different systems to find the systems he/she requires to reach their objectives.
- Eventually, access to the crown jewel is obtained and objectives are reached.

This is, of course, a bit of a simplistic view, as lateral movement could include a number of different hops before access to the crown jewels is actually obtained. It's important to note here, however, that the adversary is "trying" stuff and thus, might make mistakes which could trigger alerts and could allow us to detect their activity!

© 2019 Erik Van Buggenhout & Stephen Sims


Course Roadmap

- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- **Day 4: Lateral Movement**
- Day 5: Action on Objectives, Threat Hunting & Incident Response
- Day 6: APT Defender Capstone

SEC599.4

Protecting administrative access

- Active Directory security concepts
- Principle of least privilege & UAC
- Exercise: Implementing LAPS
- Privilege escalation techniques in Windows
- Exercise: Local Windows privilege escalation techniques

Key attack strategies against AD

- Abusing local admin privileges to steal more credentials
- Exercise: Hardening Windows against credential compromise
- Bloodhound – Mapping out AD attack paths
- Exercise: Mapping attack paths using BloodHound
- Kerberos attacks: Kerberoasting, Silver Tickets, Over-PtH
- Exercise: Kerberos attack strategies

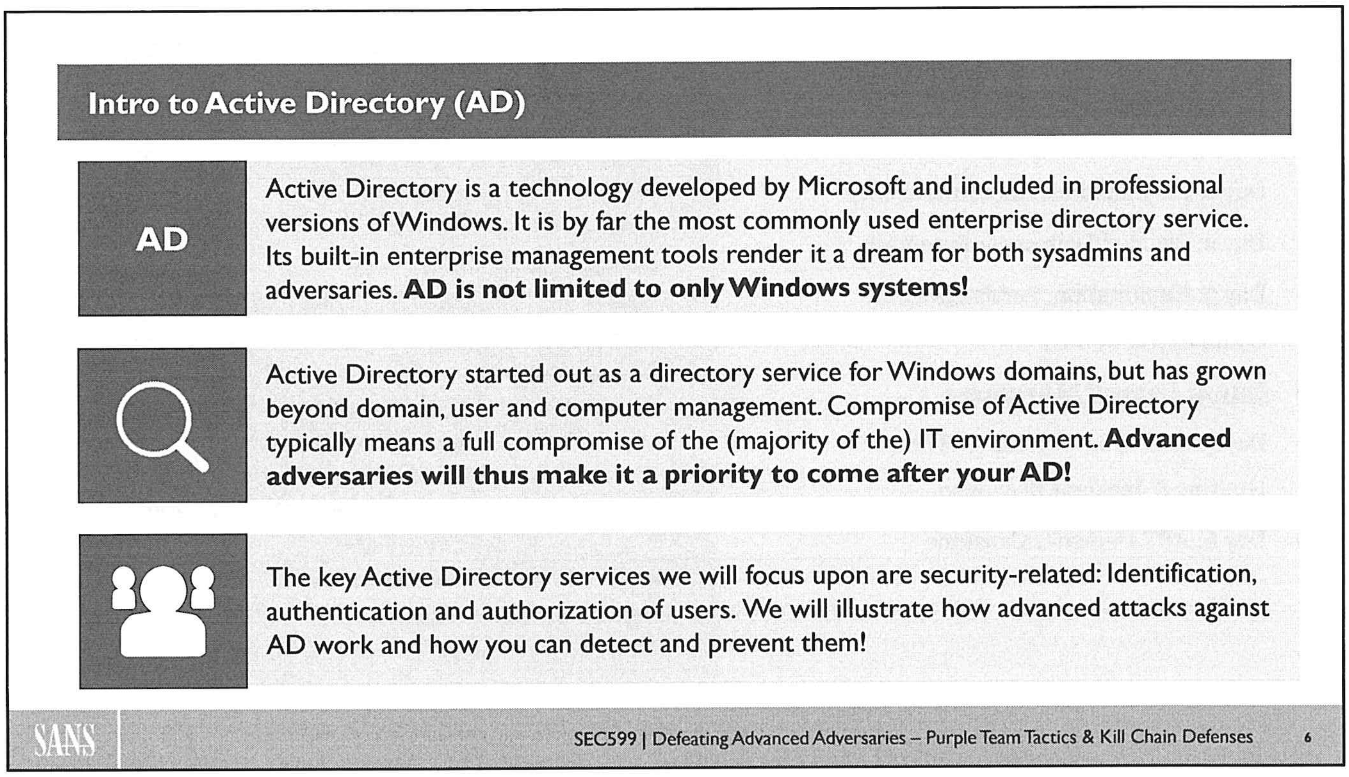
How can we detect lateral movement?

- Key logs to detect lateral movement in AD
- Deception – Tricking the adversary
- Exercise: Detecting lateral movement in AD

SANS

SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 5

This page intentionally left blank.



Intro to Active Directory (AD)

AD

Active Directory is a technology developed by Microsoft and included in professional versions of Windows. It is by far the most commonly used enterprise directory service. Its built-in enterprise management tools render it a dream for both sysadmins and adversaries. **AD is not limited to only Windows systems!**

Active Directory started out as a directory service for Windows domains, but has grown beyond domain, user and computer management. Compromise of Active Directory typically means a full compromise of the (majority of the) IT environment. **Advanced adversaries will thus make it a priority to come after your AD!**

The key Active Directory services we will focus upon are security-related: Identification, authentication and authorization of users. We will illustrate how advanced attacks against AD work and how you can detect and prevent them!

SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 6

Intro to Active Directory (AD)
In the nineties, Microsoft introduced a professional version of Windows: Windows NT (New Technology). Windows machines (servers and workstations) were grouped in "domains". Domains are logical groups of machines. They are managed by specialized Windows machines: The domain controller (primary domain controller and backup domain controllers). Domains allowed for centralized management of users and machines.

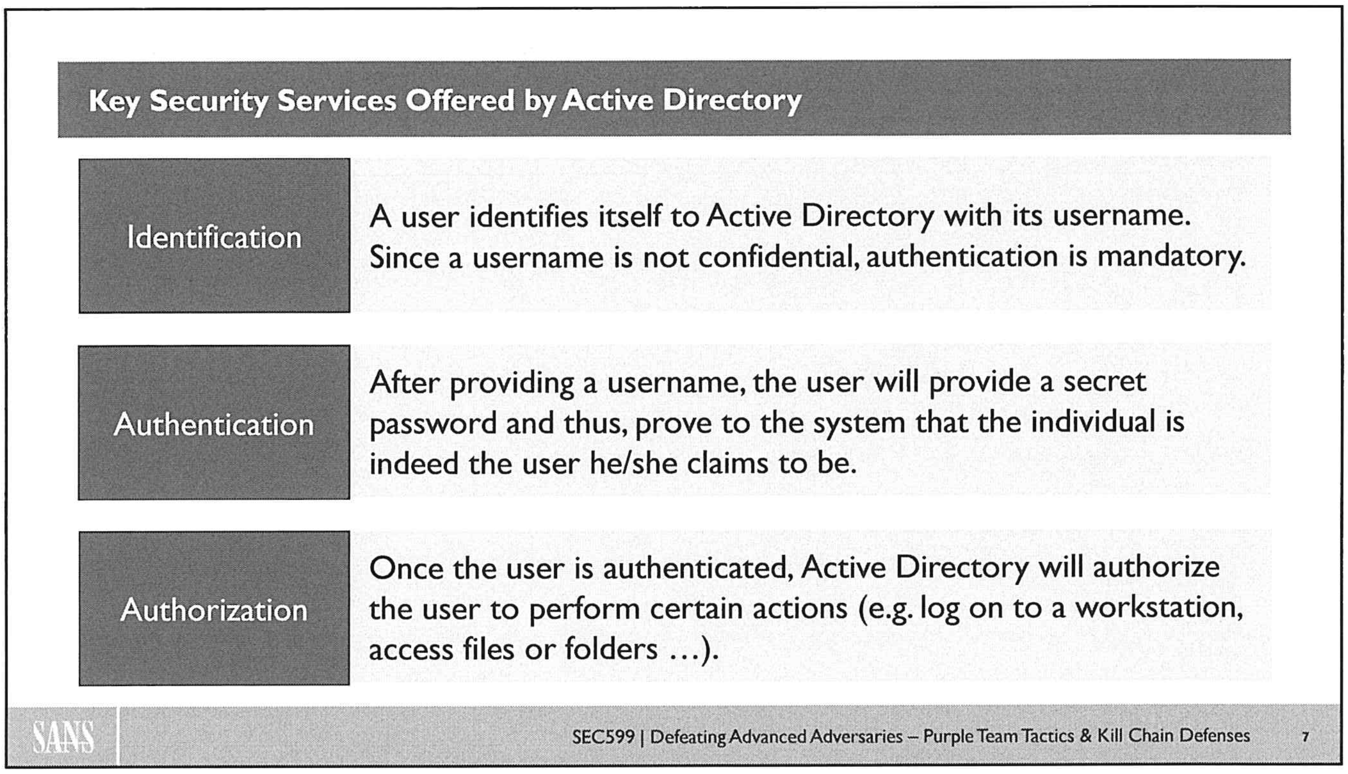
With Windows 2000 (released in 2000), Microsoft released Active Directory. Active Directory, when it was introduced, was a directory service for Windows domains: Like a phone directory, it contains metadata about all the entities it contains: Users, groups, machines ...

With Windows NT, domains were stand-alone: If an organization needed a domain for its production environment and a domain for its research environment, then 2 domains had to be created (with the corresponding domain controllers) and there were little facilities to establish relationships between the 2 domains, except sharing credentials. With the introduction of Active Directory, a hierarchy of domains could be built: A domain tree where trust relationships could be established between domains. Furthermore, forests can be built that link different domain trees together. Note that the domain is not considered a security boundary, but the forest is.

An important feature of Active Directory is Group Policy: This is made up of various configuration settings (for computers and users) that can be configured and managed centrally and applied throughout the domain.

A key feature of Active Directory is security: Identification, authentication, and authorization of users (and computers). It's this feature of Active Directory that is often attacked by advanced adversaries.

© 2019 Erik Van Buggenhout & Stephen Sims



Key Security Services Offered by Active Directory

Identification	A user identifies itself to Active Directory with its username. Since a username is not confidential, authentication is mandatory.
Authentication	After providing a username, the user will provide a secret password and thus, prove to the system that the individual is indeed the user he/she claims to be.
Authorization	Once the user is authenticated, Active Directory will authorize the user to perform certain actions (e.g. log on to a workstation, access files or folders ...).

Key Security Services Offered by Active Directory
When a user logs on to a machine via Active Directory, many actions are performed. The actions we are interested in are those that pertain to security, and more specifically: Identification, authentication, authorization.

In a normal situation, when logging onto Active Directory, the user is presented with a logon screen where credentials must be provided: Username and password. This information must be typed into the fields via a keyboard.

The username is what uniquely identifies the user; by typing a username, Active Directory knows which user account wants to authenticate. The username is not confidential, and it is displayed when the user types it. Since the username is not confidential, many users know the usernames of other users. In organizations that use an email address as an Active Directory username, it's obvious that usernames are not confidential.

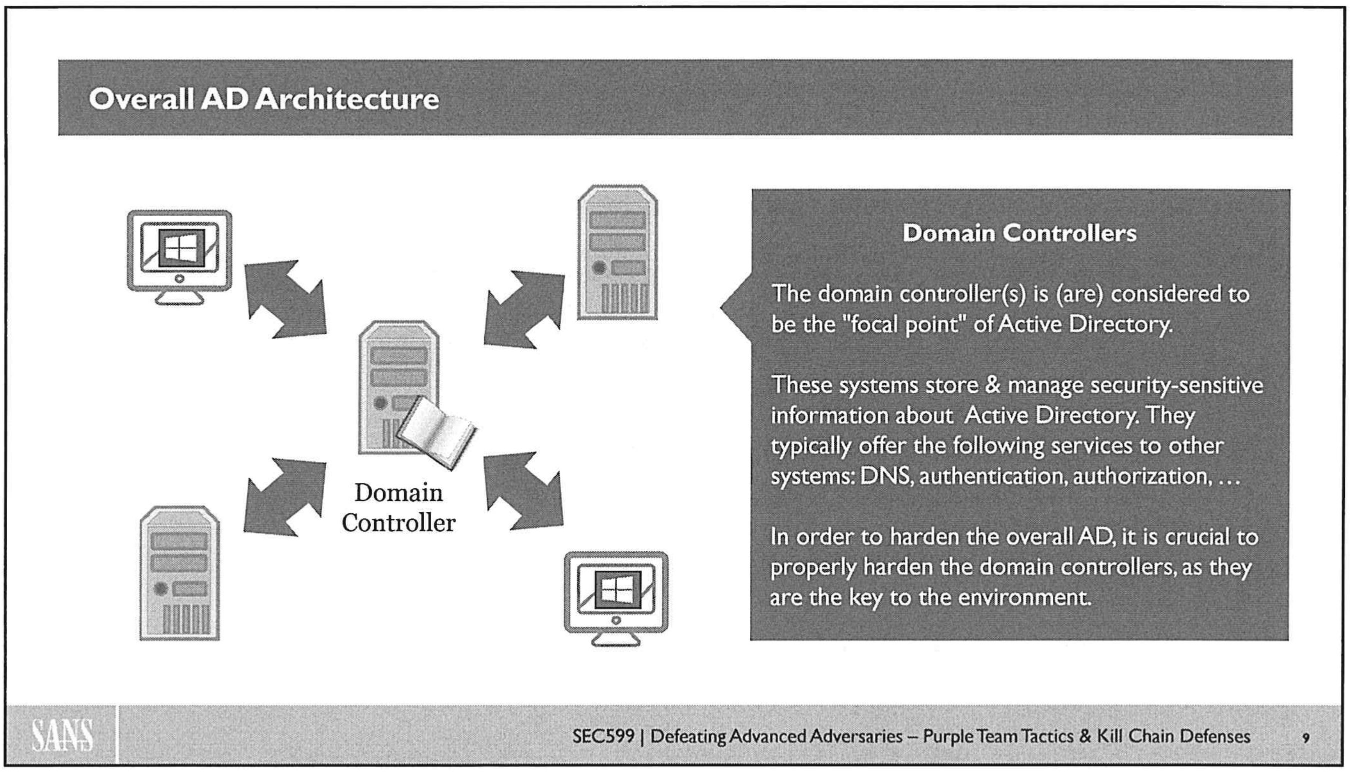
Because of the non-confidential nature of usernames, it would be easy for a user to identify as another user in Active Directory, just by typing that username. Because this is what identifies a user to Active Directory. However, when it comes to security, Active Directory does not blindly identify a user by their username without further checks.

To prove to Active Directory that a user is who they claim to be, the user has to authenticate by providing the second part of the credential: The password. Because a password authenticates a user, it has to be kept secret: Only the user must know the password. If an adversary knows the password of a user, the adversary can authenticate to Active Directory as that user, and the adversary will obtain all the rights this user has.

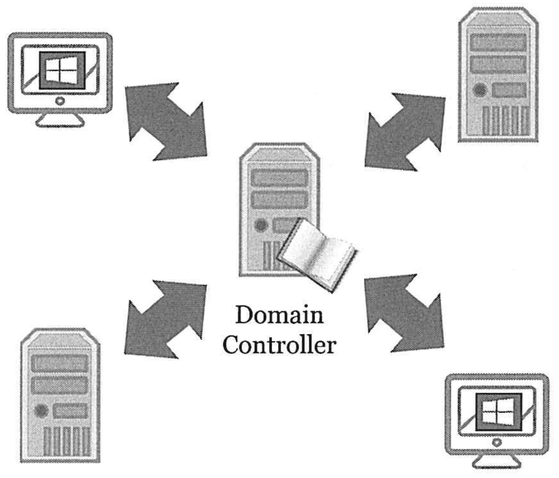
When typing a password into the password field, it is kept confidential: Each typed character of the password is hidden by showing an asterisk (*).

When a user provides valid credentials (an existing username with the corresponding, valid password) to Active Directory, Active Directory looks up information for this user account to decide what the account is authorized to do. For example, if the account is not authorized to log on to the computer, a message will be displayed, and the user will not be given access to the computer.

© 2019 Erik Van Buggenhout & Stephen Sims



Overall AD Architecture



Domain Controllers

The domain controller(s) is (are) considered to be the "focal point" of Active Directory.

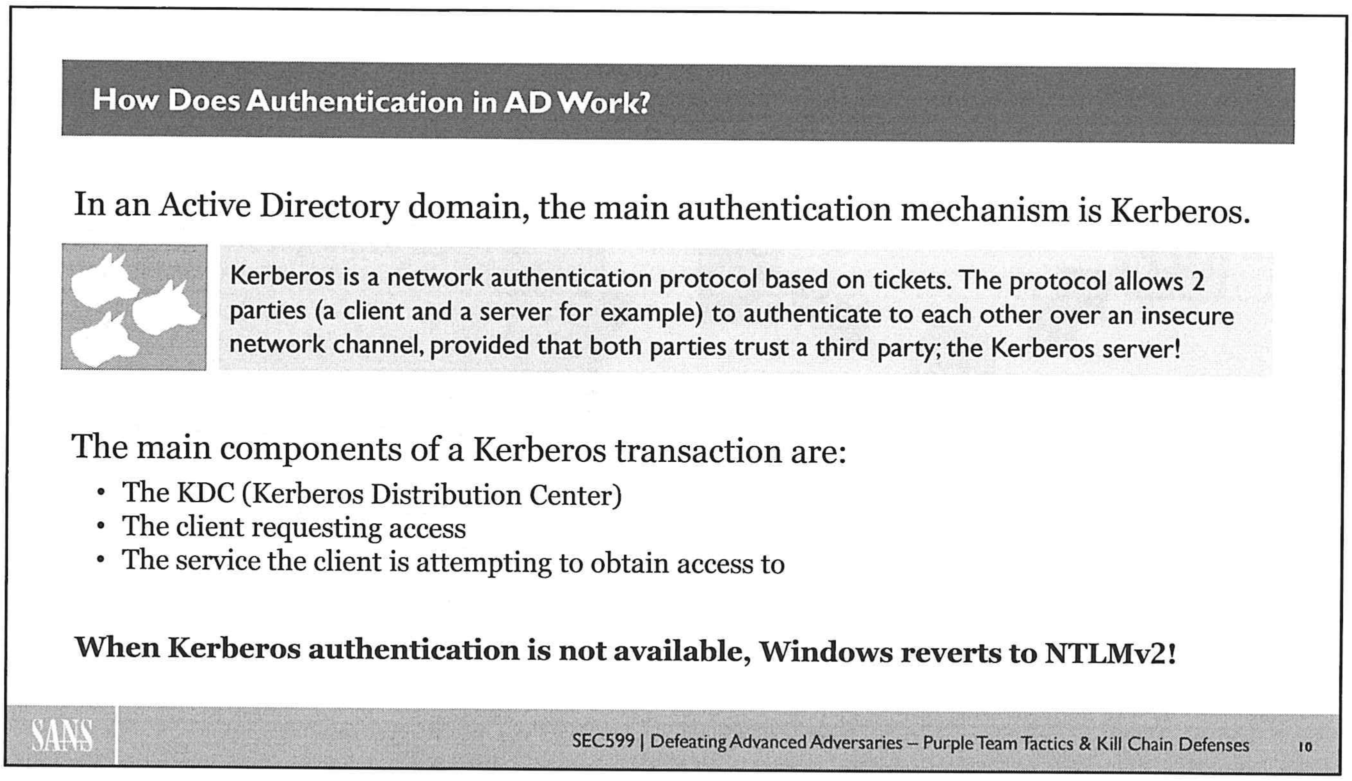
These systems store & manage security-sensitive information about Active Directory. They typically offer the following services to other systems: DNS, authentication, authorization, ...

In order to harden the overall AD, it is crucial to properly harden the domain controllers, as they are the key to the environment.

Overall AD Architecture
The security services described in the previous slide (identification, authentication & authorization) are delivered by the domain controllers in a Windows AD environment. The domain controller(s) is (are) considered to be the "focal point" of Active Directory.

These systems store & manage security-sensitive information about Active Directory. Next to the security services, they typically also offer additional services like DNS to systems in the environment.

In order to harden the overall AD, it is crucial to properly harden the domain controllers, as they are the key to the environment. If adversaries compromise a domain controller, the entire domain should be considered compromised.



How Does Authentication in AD Work?

In an Active Directory domain, the main authentication mechanism is Kerberos.

Kerberos is a network authentication protocol based on tickets. The protocol allows 2 parties (a client and a server for example) to authenticate to each other over an insecure network channel, provided that both parties trust a third party; the Kerberos server!

The main components of a Kerberos transaction are:

- The KDC (Kerberos Distribution Center)
- The client requesting access
- The service the client is attempting to obtain access to

When Kerberos authentication is not available, Windows reverts to NTLMv2!

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 10

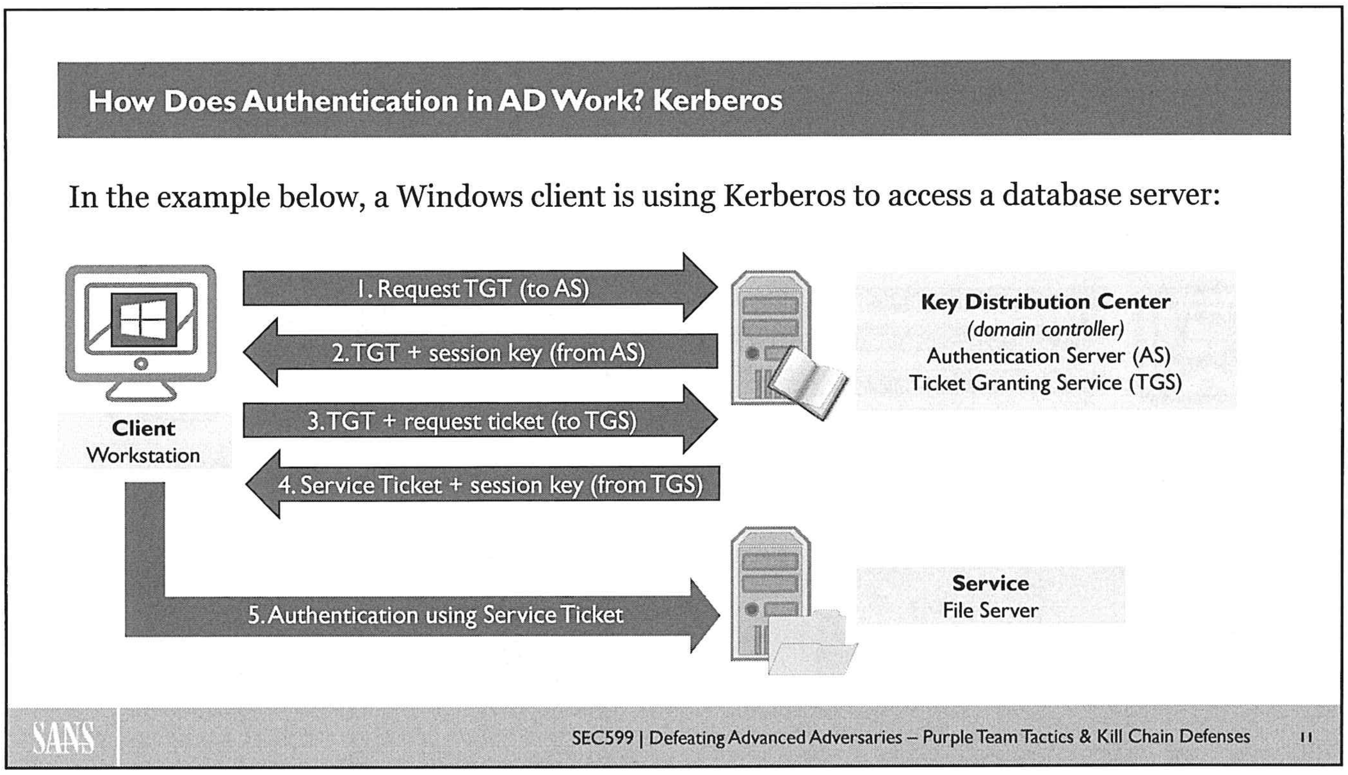
How Does Authentication in AD Work?
With Active Directory, Microsoft introduced a new authentication scheme called Kerberos. Kerberos is a network authentication protocol based on tickets, developed by MIT. The protocol allows 2 parties (a client and a server for example) to authenticate to each other over an insecure network channel, provided that both parties trust the third party: The Kerberos server. Given these three components, Kerberos is named after the three-headed mythological dog.

Each party in a Kerberos environment authenticates to the Kerberos server and receives a ticket. More precisely, this is a ticket-granting-ticket: A ticket that can be used to request tickets. When one party (client) wants to use a service from a second party (server), the first party uses its ticket-granting-ticket to obtain a ticket for the second party from the Kerberos server, and then presents it to the second party, thereby authenticating to the third party. Since both parties trust the Kerberos server, a ticket provided by the Kerberos server is trusted by both parties, leading to authentication and authorization.

Before Kerberos was introduced, pre-Active Directory Windows domains (Windows NT) relied on NTLM challenge / response authentication. NTLM provides authentication between two parties, without a third trusted party.

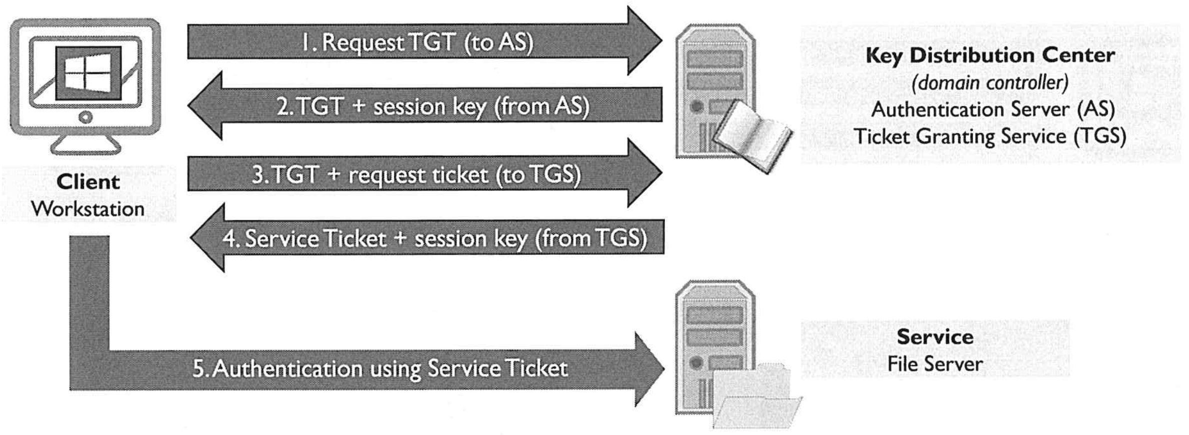
If Kerberos cannot be used (various reasons apply), Windows will fall back to NTLMv2 authentication.

© 2019 Erik Van Buggenhout & Stephen Sims



How Does Authentication in AD Work? Kerberos

In the example below, a Windows client is using Kerberos to access a database server:


1. Request TGT (to AS)
2. TGT + session key (from AS)
3. TGT + request ticket (to TGS)
4. Service Ticket + session key (from TGS)
5. Authentication using Service Ticket
Client
Workstation
Key Distribution Center
(domain controller)
Authentication Server (AS)
Ticket Granting Service (TGS)
Service
File Server

How Does Authentication in AD Work? Kerberos
In Active Directory, when a client successfully authenticates to a domain controller, the client is given a ticket-granting-ticket.

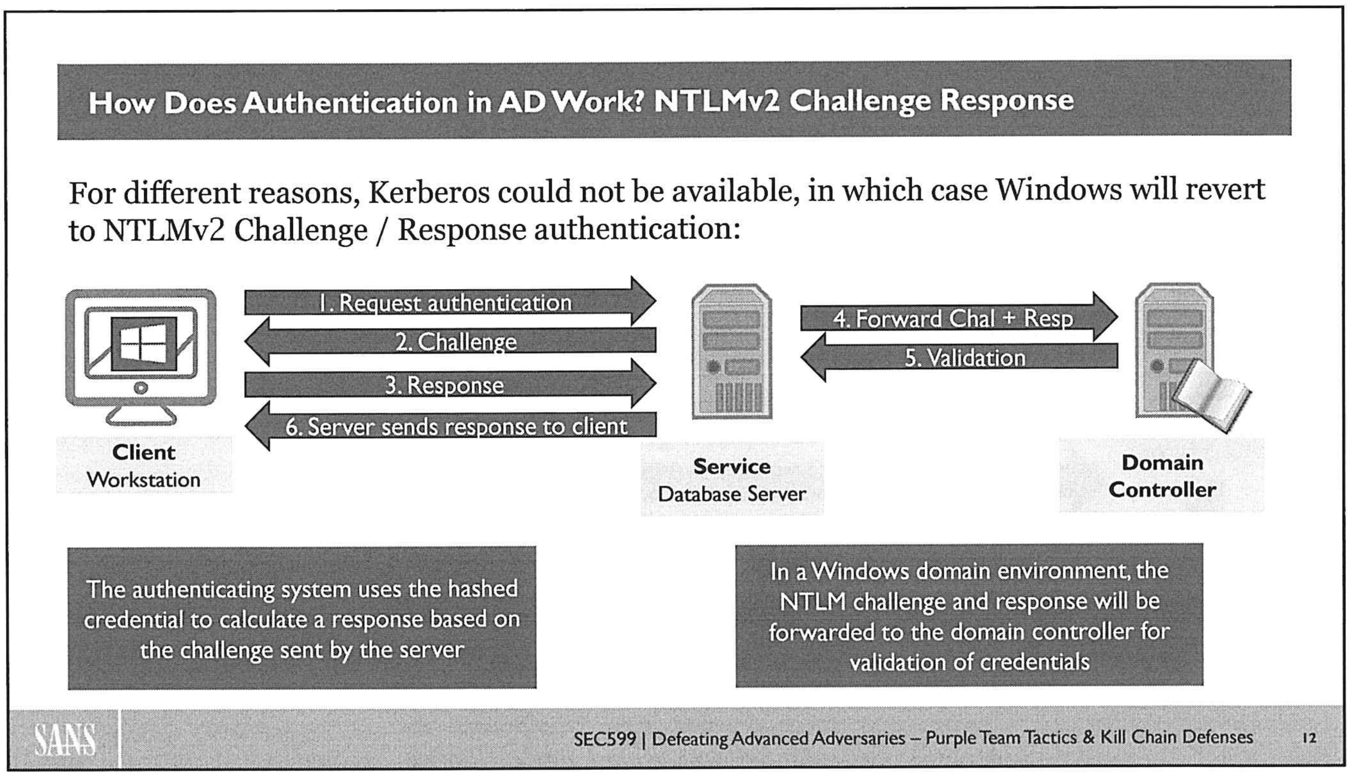
To authenticate, the user's workstation will send a request to the Authentication Server (AS) on the domain controller. This request includes the credentials (encrypted with keys obtained from the Key Distribution Center (KDC), also a service provided by a domain controller). The AS will decrypt and validate the credentials (lookup the NTLM hash) and if the NTLM hash matches, the client will be given a ticket-granting-ticket (TGT) by the Ticket Granting Service (TGS).

Kerberos works with tickets: Cryptographically secured pieces of data that grant access to a service for a particular user and during a well-defined period. For example, you can receive a ticket to access a particular share on a file server.

A ticket-granting-ticket is the ticket a client receives first, and it is a special ticket: It is a ticket for the krbtgt service and can thus be used to request other tickets. By default, a TGT is valid for 10 hours. When it expires, Windows will automatically and transparently request a new one.

Once a client has obtained the TGT, it can request access to other services, for example, a file share. To obtain this access, the client sends its TGT to the TGS along with the details of the service it wants to access. The TGS will validate the TGT, and then check Active Directory if the user is authorized to access the service. When this is the case, the TGS will issue a ticket to the client for the service it wants to access.

To access the service (for example a file share), the client will send the ticket to the service (the file server for example). Since the file server also trusts Kerberos (the TGS), it will accept tickets and grant access to the service once it has validated through cryptographic means that this is an authentic ticket issued by the TGS trusted by the service.



How Does Authentication in AD Work? NTLMv2 Challenge Response

For different reasons, Kerberos could not be available, in which case Windows will revert to NTLMv2 Challenge / Response authentication:

The authenticating system uses the hashed credential to calculate a response based on the challenge sent by the server

In a Windows domain environment, the NTLM challenge and response will be forwarded to the domain controller for validation of credentials

SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 12

How Does Authentication in AD Work? NTLMv2 Challenge Response
When Kerberos authentication is not possible, Windows will fall back to NTLM authentication.

This can even happen between machines that are members of the same domain, but when all necessary conditions to use Kerberos are not in place. For example, Kerberos works with so-called service names. If we don't have a name, Kerberos cannot be used. This is the case when we access a share on a file server by using the IP address of the server instead of its servername.

NTLM authentication is a 2-party authentication: The client and the server. It takes 3 steps:

1. Negotiate
2. Challenge
3. Response

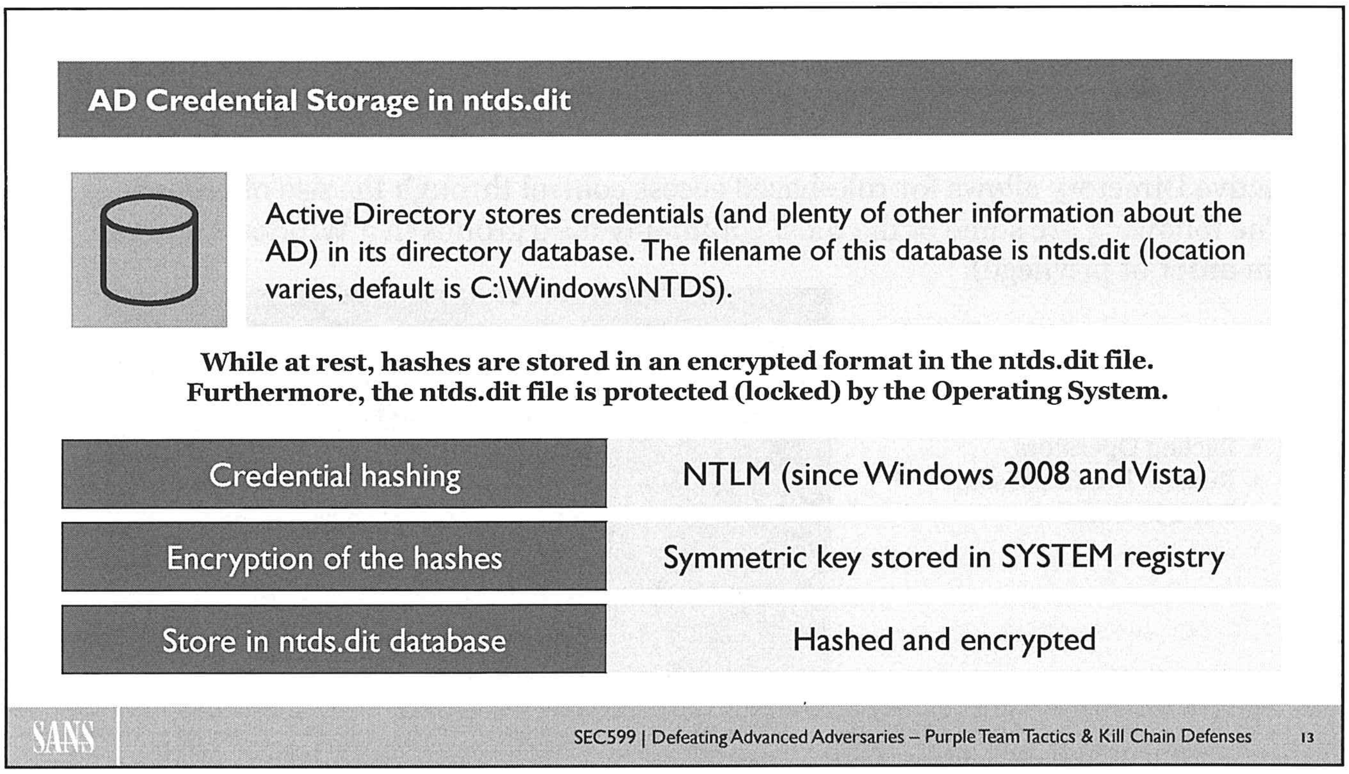
First, the client sends a negotiate packet to the server to request the authentication. There are different parameters and versions for NTLM, and the client has to inform the server what it is capable of. This is done with a negotiate packet.

Next, (step 2) the server sends a challenge packet to the client. This challenge contains a so-called "nonce". A nonce is a random number of 16 bytes. Finally (step 3), the client sends the response to the server: It calculates a response based on its password and the nonce and sends that to the server. Using a nonce allows the 2 parties to perform authentication without having to send the password (cleartext or encrypted) over the network.

The server checks the credentials of the client by performing the same calculation as the client for the response, and if the response calculated by the server is the same as the response calculated by the client, then the client is authenticated to the server.

The server needs the credentials of the client to perform the calculation In an Active Directory environment, the server obtains the credentials from a domain controller.

© 2019 Erik Van Buggenhout & Stephen Sims



AD Credential Storage in ntds.dit

Active Directory stores credentials (and plenty of other information about the AD) in its directory database. The filename of this database is ntds.dit (location varies, default is C:\Windows\NTDS).

While at rest, hashes are stored in an encrypted format in the ntds.dit file. Furthermore, the ntds.dit file is protected (locked) by the Operating System.

Credential hashing	NTLM (since Windows 2008 and Vista)
Encryption of the hashes	Symmetric key stored in SYSTEM registry
Store in ntds.dit database	Hashed and encrypted

SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 13

AD Credential Storage in ntds.dit
Active Directory has to store credentials in a database: The username and the password.

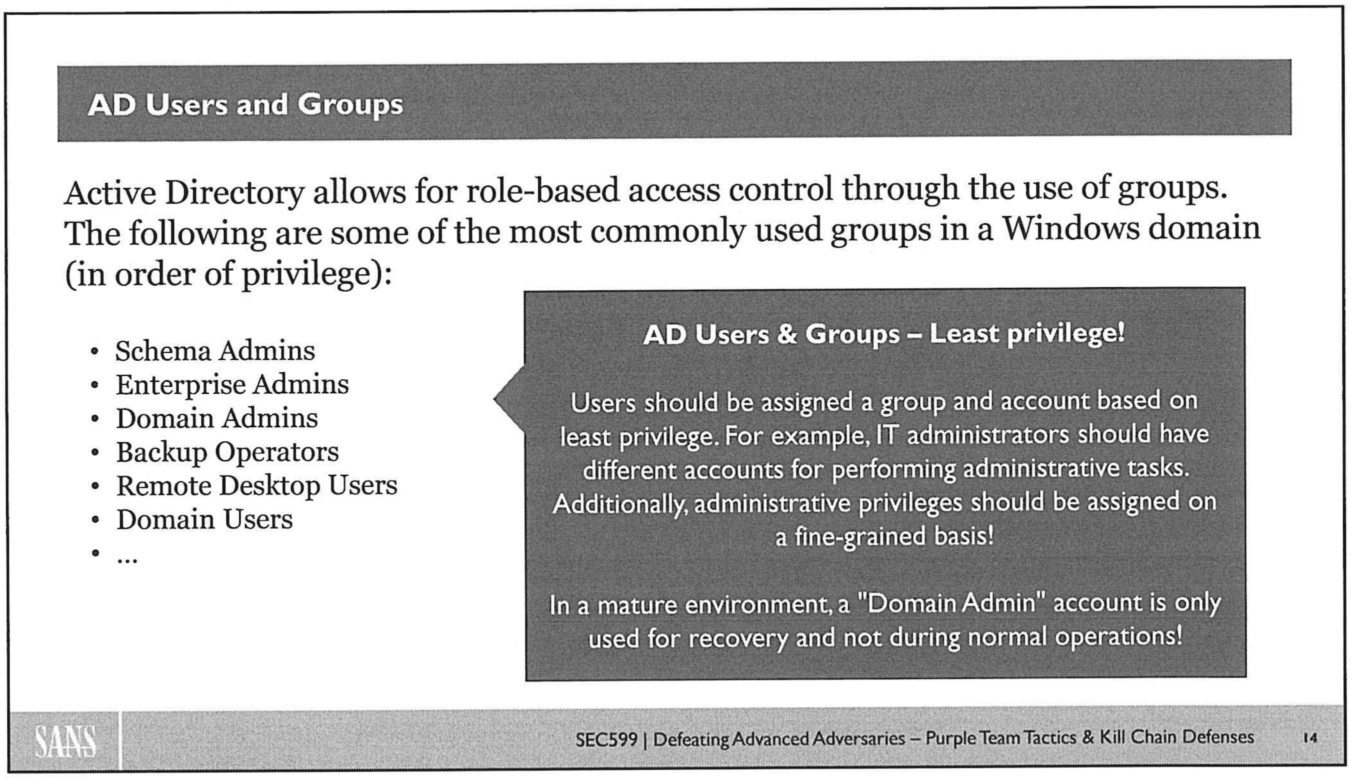
This database file is ntds.dit. This is a database in the ESE format (Extensible Storage Engine) from Microsoft, this format is also used by Microsoft Exchange, for example. By default, this file is stored in folder C:\Windows\NTDS on domain controllers.

To protect passwords, the hashed value of the passwords is stored in the ntds.dit database.
Windows uses a hashing function that Microsoft officially calls NTOWF. This stands for NTLM One-Way Function. However, in most literature on Windows passwords, the name NTLM will be used for the hashing function instead of NTOWF. We will do the same in this training: NTLM hashing function.

The NTLM hashing function is an MD4 hash of the little-endian UNICODE representation of the password. MD4 is a cryptographic hashing algorithm; its design was later used to design the well-known MD5 hashing cryptographic hashing algorithm. MD4 (and MD5 too) is no longer a strong, cryptographic hashing algorithm; it has been broken.

On older Windows versions, another hash function will be used, too; LM hashing (officially LMOWF). Storing passwords as LM hashes should be avoided at all costs (it is no longer used on modern Windows versions and can be disabled on older versions). LM hashing is very weak.

To provide further protection, the hashes stored inside the NTDS database are also encrypted with a symmetric encryption cipher. The key used for encryption is unique to every domain controller and stored inside the SYSTEM registry hive.



AD Users and Groups

Active Directory allows for role-based access control through the use of groups. The following are some of the most commonly used groups in a Windows domain (in order of privilege):

- Schema Admins
- Enterprise Admins
- Domain Admins
- Backup Operators
- Remote Desktop Users
- Domain Users
- ...

AD Users & Groups – Least privilege!

Users should be assigned a group and account based on least privilege. For example, IT administrators should have different accounts for performing administrative tasks. Additionally, administrative privileges should be assigned on a fine-grained basis!

In a mature environment, a "Domain Admin" account is only used for recovery and not during normal operations!

SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 14

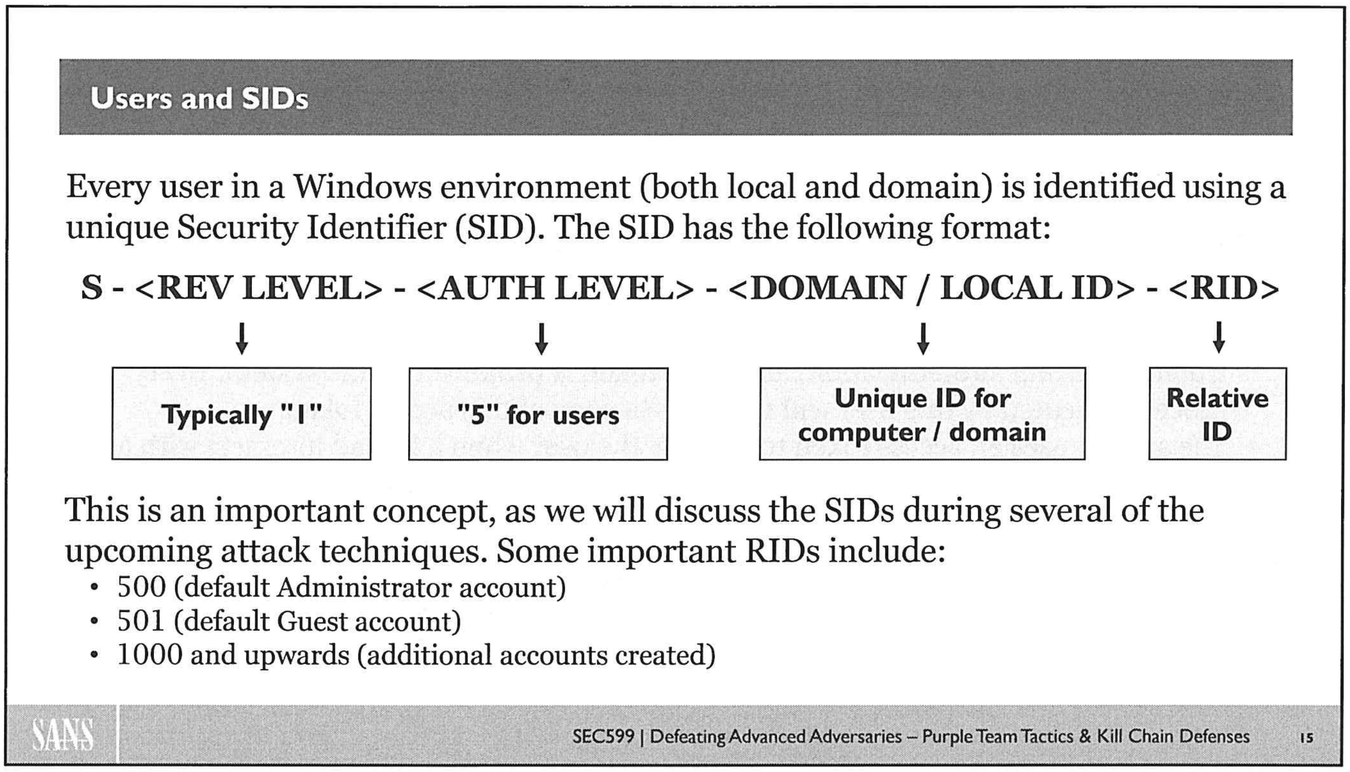
AD Users and Groups

Active Directory allows for role-based access control through the use of groups. The following are some of the most commonly used groups in a Windows domain (in order of privilege):

- Schema Admins: Members of this group can modify Active Directory schema. By default, the Administrator account is a member of this group.
- Enterprise Admins: Members of this group have full control of all domains in the forest. By default, this group is a member of the Administrators group on all domain controllers in the forest.
- Domain Admins: Members of this group have full control of the domain. By default, this group is a member of the Administrators group on all domain controllers, all domain workstations, and all domain member servers at the time they are joined to the domain.
- Backup Operators: Members of this group can back up and restore files on the server, regardless of any permissions that protect those files. This is because the right to perform a backup takes precedence over all file permissions.
- Remote Desktop Users: Members of this group can log on remotely to a server.
- Domain Users: This group contains all domain users. By default, any user account that is created in the domain becomes a member of this group automatically.
- ...

Users should be assigned a group and account based on least privilege. For example, IT administrators should have different accounts for performing administrative tasks. Additionally, administrative privileges should be assigned on a fine-grained basis! Many organizations will use specific groups, based on least privilege (e.g. workstation administrators, web server administrators, database administrators, ...). In a mature environment, a "Domain Admin" account is only used for recovery and not during normal operations.

© 2019 Erik Van Buggenhout & Stephen Sims



Users and SIDs

Every user in a Windows environment (both local and domain) is identified using a unique Security Identifier (SID). The SID has the following format:

S - <REV LEVEL> - <AUTH LEVEL> - <DOMAIN / LOCAL ID> - <RID>

Typically "1"
"5" for users
Unique ID for computer / domain
Relative ID

This is an important concept, as we will discuss the SIDs during several of the upcoming attack techniques. Some important RIDs include:

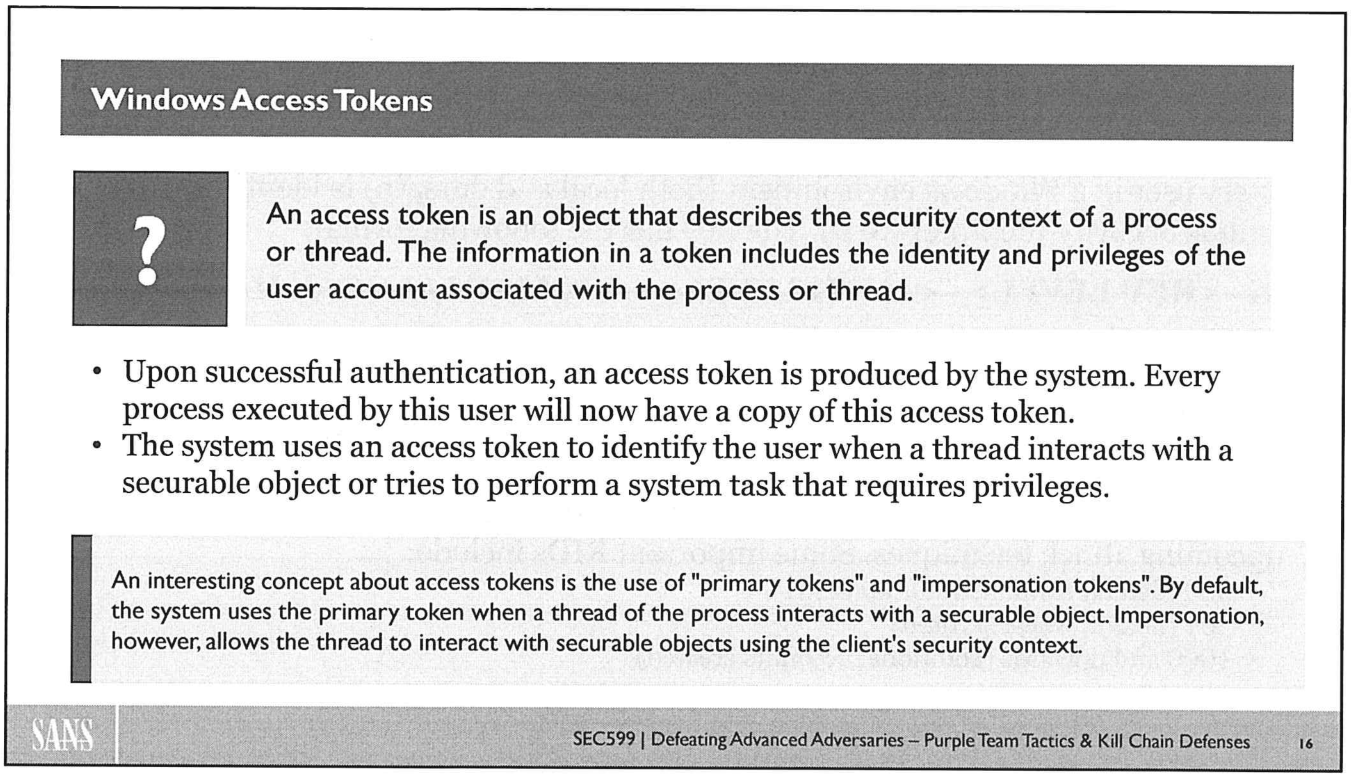
- 500 (default Administrator account)
- 501 (default Guest account)
- 1000 and upwards (additional accounts created)

Users and SIDs
Every user in a Windows environment (both local and domain) is identified using a unique Security Identifier (SID). The SID has the following format:

S - <REV LEVEL> - <AUTH LEVEL> - <DOMAIN / LOCAL ID> - <RID>

- <REV LEVEL> is typically 1
- <AUTH LEVEL> will be 5 for users
- <DOMAIN / LOCAL ID> will be a unique ID for a computer or a domain. All local users on the same machine will share the same ID. Furthermore, all domain users in the same domain will share the same ID.
- <RID> is a relative ID assigned to this specific user. Some well-known RIDs include:
  - 500 (default Administrator account)
  - 501 (default Guest account)
  - 1000 and upwards (additional accounts created)

This is an important concept, as we will discuss the SIDs during several of the upcoming attack techniques.



Windows Access Tokens

?

An access token is an object that describes the security context of a process or thread. The information in a token includes the identity and privileges of the user account associated with the process or thread.

- Upon successful authentication, an access token is produced by the system. Every process executed by this user will now have a copy of this access token.
- The system uses an access token to identify the user when a thread interacts with a securable object or tries to perform a system task that requires privileges.

An interesting concept about access tokens is the use of "primary tokens" and "impersonation tokens". By default, the system uses the primary token when a thread of the process interacts with a securable object. Impersonation, however, allows the thread to interact with securable objects using the client's security context.

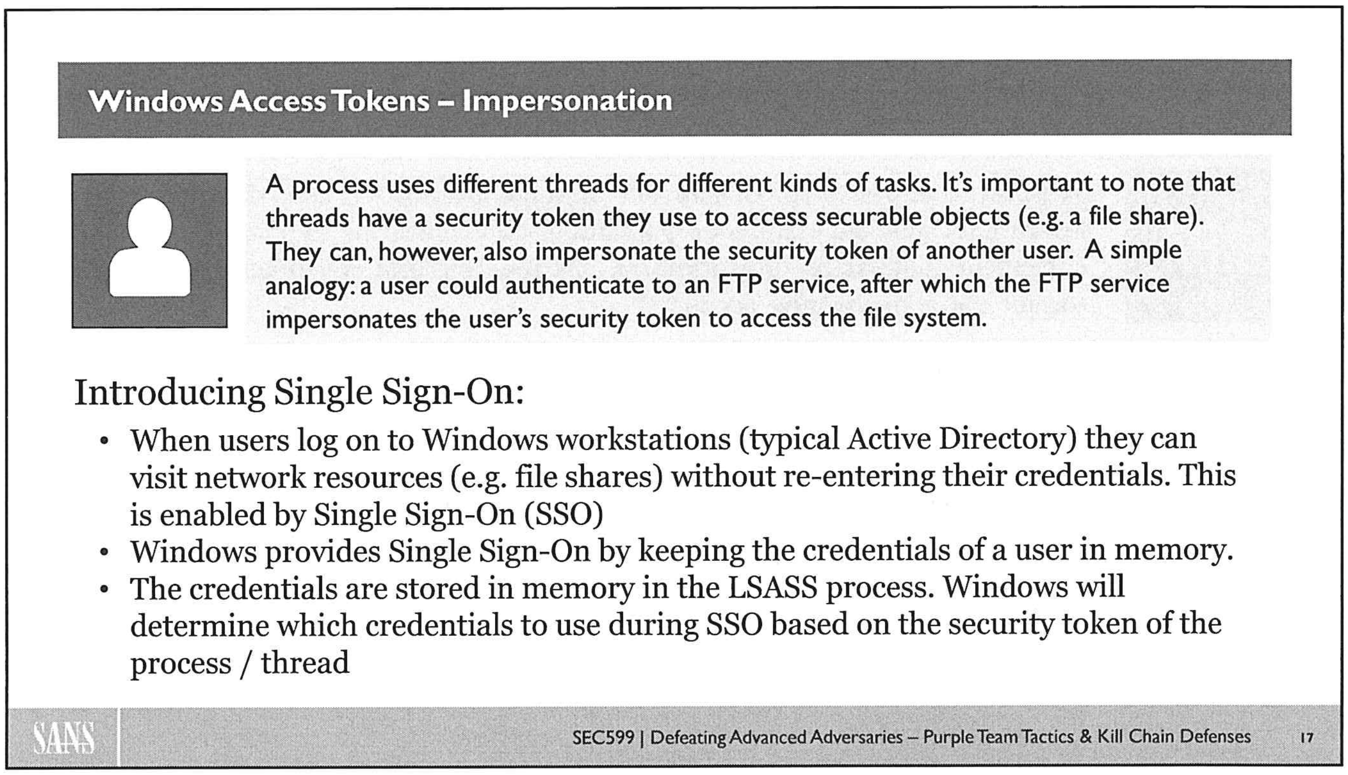
SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 16

Windows Access Tokens
An access token is an object that describes the security context of a process or thread. The information in a token includes the identity and privileges of the user account associated with the process or thread.

- Upon successful authentication, an access token is produced by the system. Every process executed by this user will now have a copy of this access token.
- The system uses an access token to identify the user when a thread interacts with a securable object or tries to perform a system task that requires privileges.

An interesting concept about access tokens is the use of "primary tokens" and "impersonation tokens". By default, the system uses the primary token when a thread of the process interacts with a securable object. Impersonation, however, allows the thread to interact with securable objects using the client's security context. This opens up a number of attack vectors that can be used by adversaries!

© 2019 Erik Van Buggenhout & Stephen Sims



Windows Access Tokens – Impersonation

A process uses different threads for different kinds of tasks. It's important to note that threads have a security token they use to access securable objects (e.g. a file share). They can, however, also impersonate the security token of another user. A simple analogy: a user could authenticate to an FTP service, after which the FTP service impersonates the user's security token to access the file system.

Introducing Single Sign-On:

- When users log on to Windows workstations (typical Active Directory) they can visit network resources (e.g. file shares) without re-entering their credentials. This is enabled by Single Sign-On (SSO)
- Windows provides Single Sign-On by keeping the credentials of a user in memory.
- The credentials are stored in memory in the LSASS process. Windows will determine which credentials to use during SSO based on the security token of the process / thread

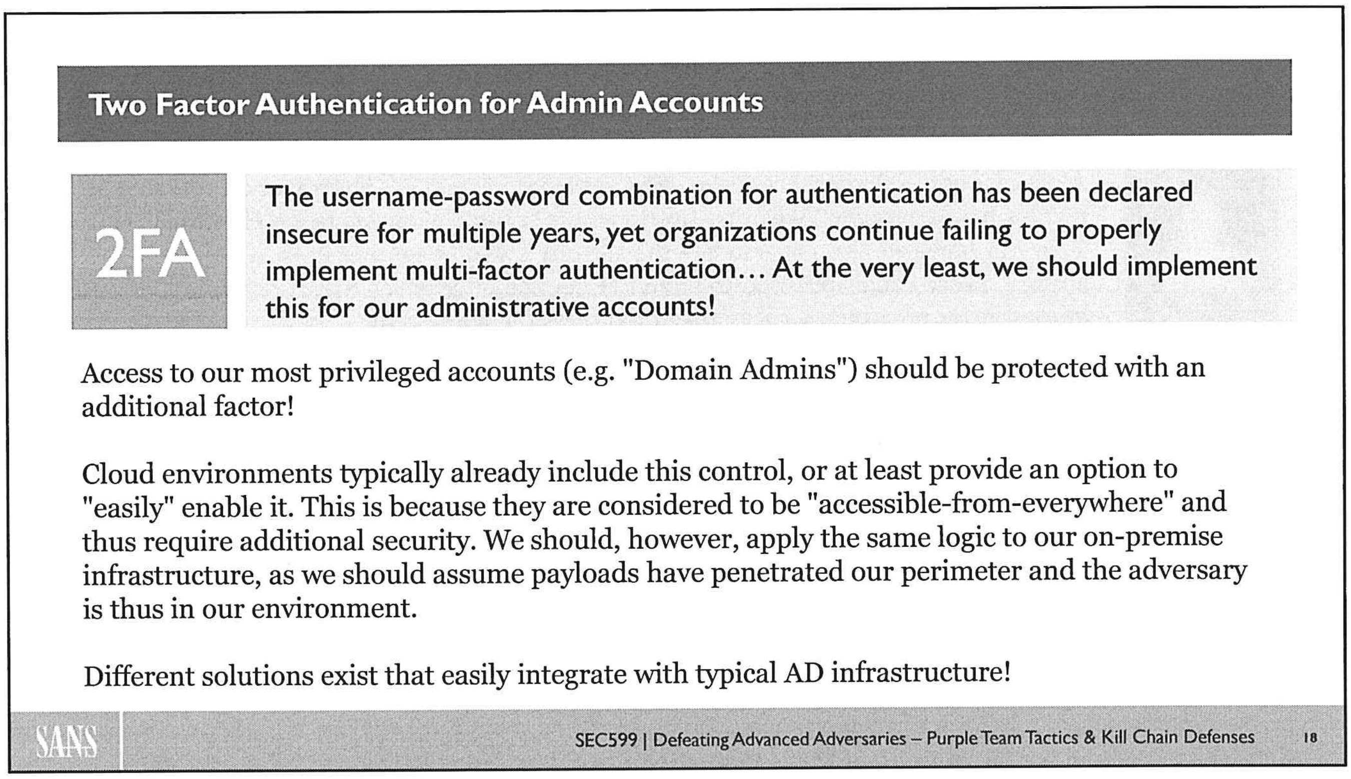
SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 17

Windows Access Tokens – Impersonation
A process uses different threads for different kinds of tasks. It's important to note that threads have a security token they use to access securable objects (e.g. a file share). They can, however, also impersonate the security token of another user. A simple analogy: a user could authenticate to an FTP service, after which the FTP service impersonates the user's security token to access the file system.

This is, for example, especially useful during Single Sign-On:

- When users log on to Windows workstations (typical Active Directory) they can visit network resources (e.g. file shares) without re-entering their credentials. This is enabled by Single Sign-On (SSO).
- Windows provides Single Sign-On by keeping the credentials of a user in memory.
- The credentials are stored in memory in the LSASS process. Windows will determine which credentials to use during SSO based on the security token of the process / thread.

During AD attacks overview, we will further discuss attacks against this mechanism!



Two Factor Authentication for Admin Accounts

2FA

The username-password combination for authentication has been declared insecure for multiple years, yet organizations continue failing to properly implement multi-factor authentication... At the very least, we should implement this for our administrative accounts!

Access to our most privileged accounts (e.g. "Domain Admins") should be protected with an additional factor!

Cloud environments typically already include this control, or at least provide an option to "easily" enable it. This is because they are considered to be "accessible-from-everywhere" and thus require additional security. We should, however, apply the same logic to our on-premise infrastructure, as we should assume payloads have penetrated our perimeter and the adversary is thus in our environment.

Different solutions exist that easily integrate with typical AD infrastructure!

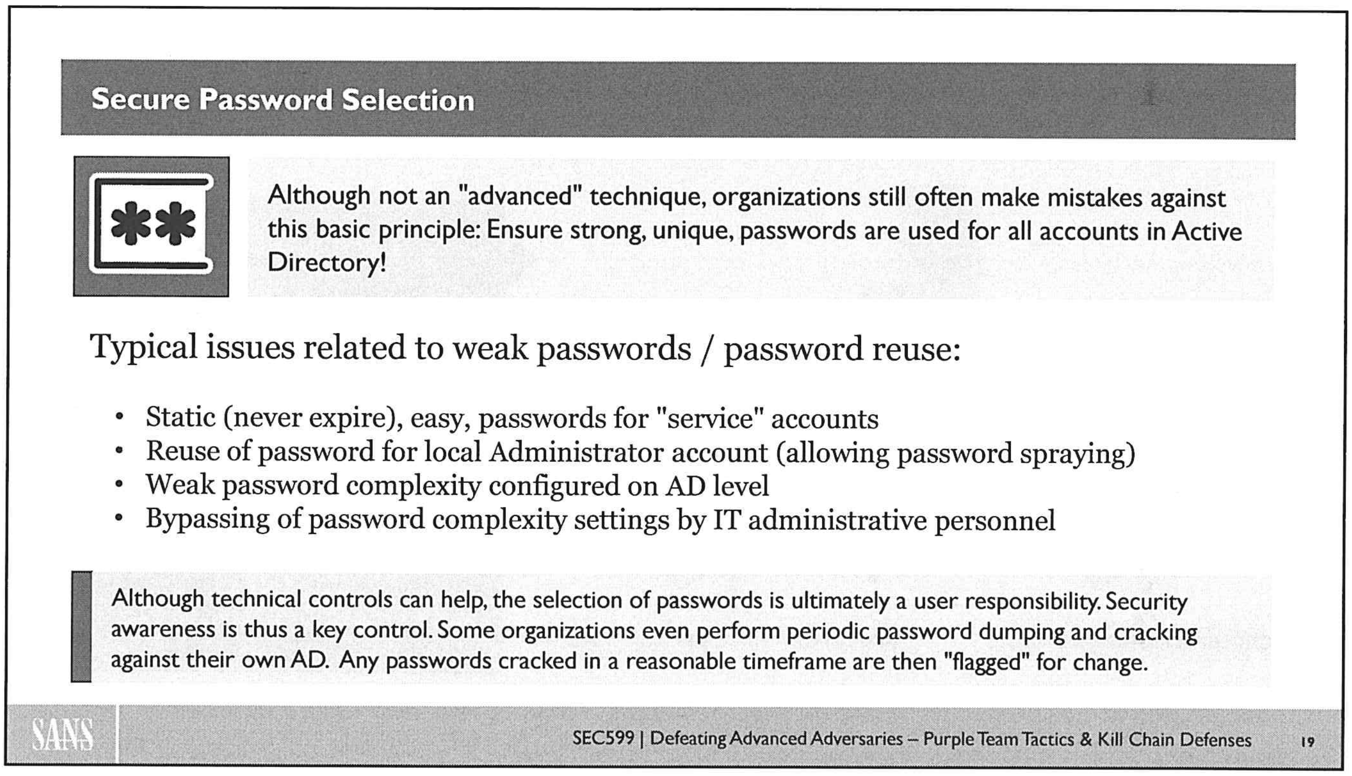
SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 18

Two Factor Authentication for Admin Accounts
The username-password combination for authentication has been declared insecure for multiple years, yet organizations continue failing to properly implement multi-factor authentication... At the very least, we should implement this for our administrative accounts!

Cloud environments typically already include this control, or at least provide an option to "easily" enable it. This is because they are considered to be "accessible-from-everywhere" and thus require additional security. We should, however, apply the same logic to our on-premise infrastructure, as we should assume payloads have penetrated our perimeter and the adversary is thus in our environment.

In order to implement this, many different vendors offer solutions that easily integrate with an existing AD infrastructure. Some examples include, for example, RSA SecurID, and WikID (they offer a free license for up to 5 accounts, which could be enough for small organizations that only want to protect a subset of accounts).

© 2019 Erik Van Buggenhout & Stephen Sims



Secure Password Selection

Although not an "advanced" technique, organizations still often make mistakes against this basic principle: Ensure strong, unique, passwords are used for all accounts in Active Directory!

Typical issues related to weak passwords / password reuse:

- Static (never expire), easy, passwords for "service" accounts
- Reuse of password for local Administrator account (allowing password spraying)
- Weak password complexity configured on AD level
- Bypassing of password complexity settings by IT administrative personnel

Although technical controls can help, the selection of passwords is ultimately a user responsibility. Security awareness is thus a key control. Some organizations even perform periodic password dumping and cracking against their own AD. Any passwords cracked in a reasonable timeframe are then "flagged" for change.

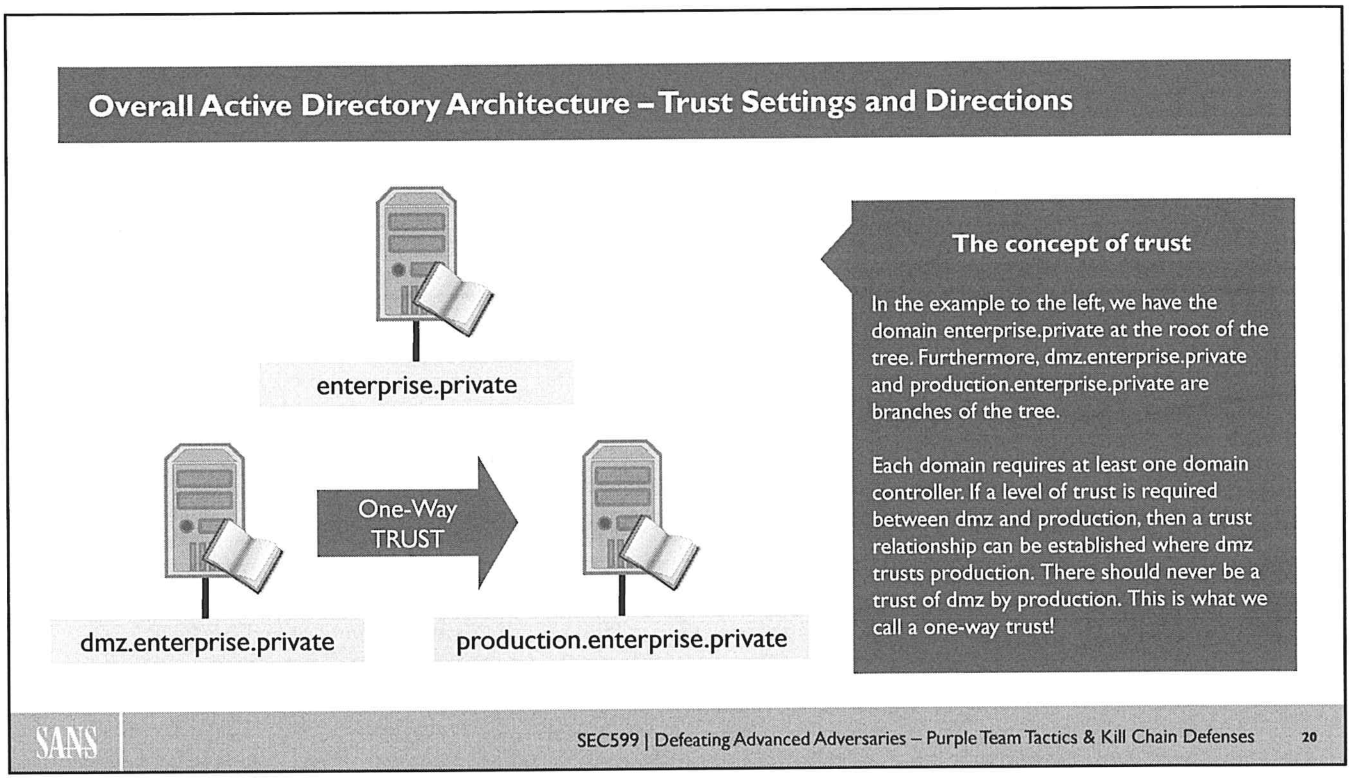
SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 19

Secure Password Selection
Although not an "advanced" technique, organizations still often make mistakes against this basic principle: Ensure strong, unique, passwords are used for all accounts in Active Directory! Throughout our careers as cyber security professionals, the course authors have observed the following typical issues on a frequent basis:

- "Technical" or "Service" accounts that are configured with a weak, static, password ("never expire"), often even with administrative privileges!
- Use of the local administrator account as a "recovery" account with the same password on all Windows hosts.
- Overall, weak password complexity settings configured on AD / domain level.
- Even if strong password complexity settings are configured, we've seen IT administrative personnel circumvent / bypass the controls for their own accounts!

As a conclusion, we'd like to note that, although technical controls can help, the selection of passwords is ultimately a user responsibility. This means security awareness is of paramount importance. If users are not convinced of the need / value of strong password selection, they will find a way to configure a weak password that meets your technical complexity controls. Even worse: They might select a strong password but then store it in an insecure way ("password.txt on the Desktop"). NIST has an excellent password standard that can be used to provide some insights in what a strong password looks like.

Some organizations have implemented offensive techniques themselves: They perform periodic password dumping and cracking against their own AD. Any passwords cracked in a reasonable timeframe are then "flagged" and users are forced to change them.



Overall Active Directory Architecture – Trust Settings and Directions

enterprise.private
One-Way TRUST
dmz.enterprise.private
production.enterprise.private

The concept of trust

In the example to the left, we have the domain enterprise.private at the root of the tree. Furthermore, dmz.enterprise.private and production.enterprise.private are branches of the tree.

Each domain requires at least one domain controller. If a level of trust is required between dmz and production, then a trust relationship can be established where dmz trusts production. There should never be a trust of dmz by production. This is what we call a one-way trust!

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 20

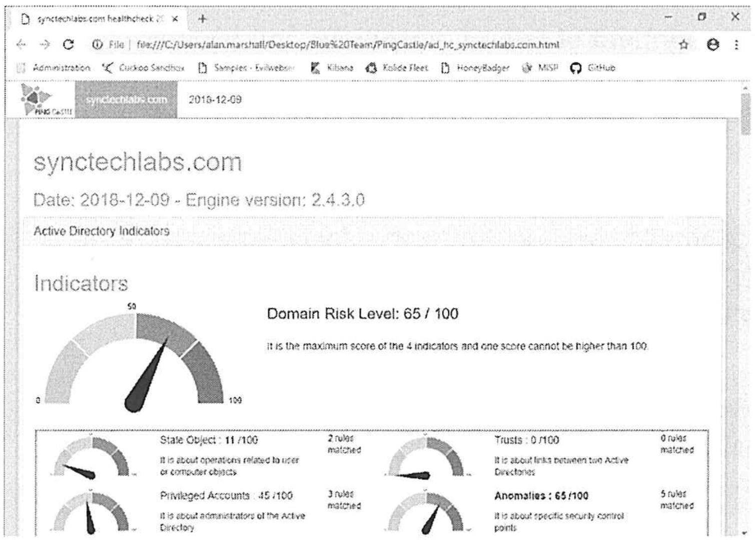
Overall Active Directory Architecture – Trust Settings and Directions
Active Directory has hierarchical structures at different levels. For example, domains can be put into a tree structure. In the example above, we have the domain enterprise.private at the root of the tree. dmz.enterprise.private and production.enterprise.private are branches of the tree.

Each domain requires at least one domain controller.

If a level of trust is required between dmz and production, then a trust relationship can be established where dmz trusts production. There should never be a trust of dmz by production. This is what we call a one-way trust! Although this is a very easy example, trust settings in large organizations can become highly complex and thus difficult to manage!

© 2019 Erik Van Buggenhout & Stephen Sims

Overall Active Directory Architecture – Audit Trust Settings using PingCastle



synctechlabs.com

Date: 2018-12-09 - Engine version: 2.4.3.0

Active Directory Indicators

Indicators

Domain Risk Level: 65 / 100

It is the maximum score of the 4 indicators and one score cannot be higher than 100

State Object : 11 /100 — 2 rules matched — It is about operations related to user or computer objects

Trusts : 0 /100 — 0 rules matched — It is about links between two Active Directories

Privileged Accounts : 45 /100 — 3 rules matched — It is about administrators of the Active Directory

Anomalies : 65 /100 — 5 rules matched — It is about specific security control points

PingCastle

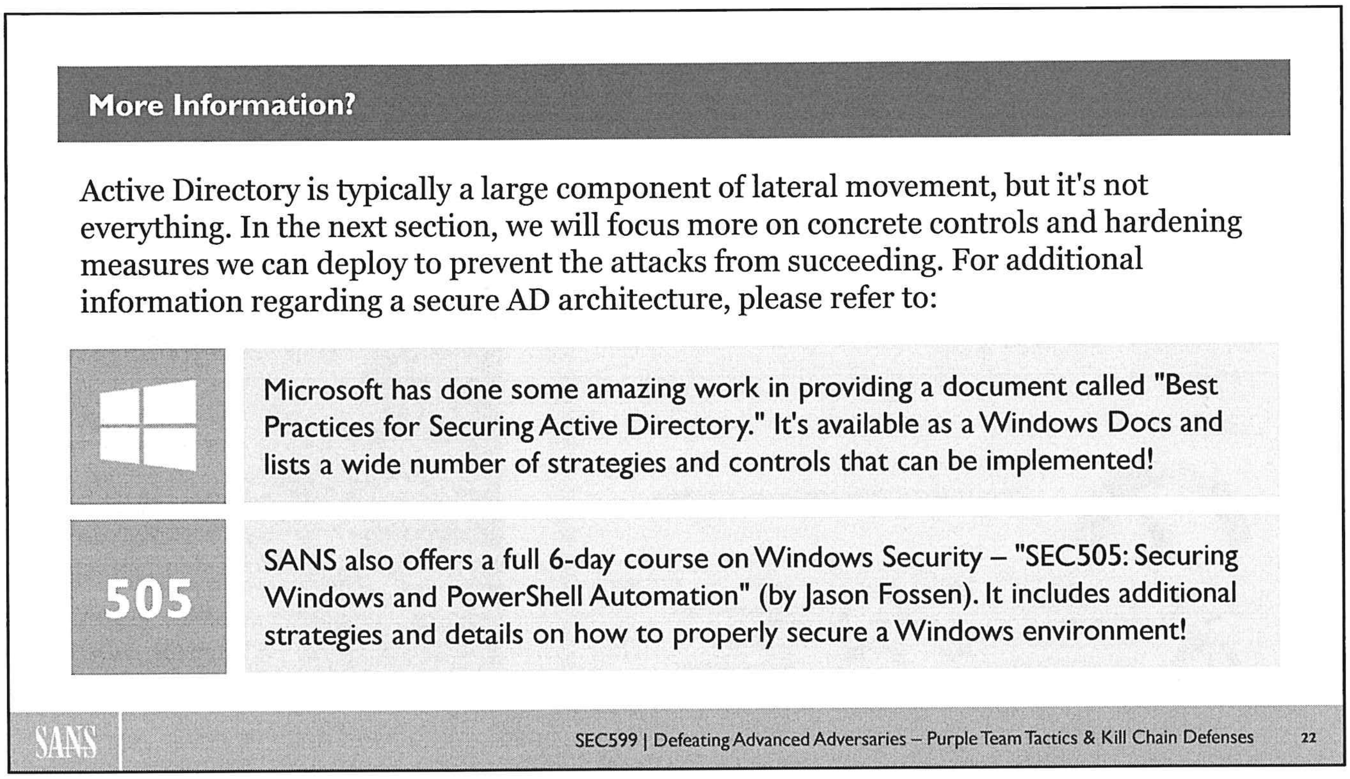
PingCastle was built by Vincent Le Toux (co-author of Mimikatz) and aims to provide a "simple" security score of a domain configuration. Its motto is "Get Active Directory Security at 80% in 20% of the time" and aims to provide pragmatic security recommendations.

Among many controls, it also reviews the domain trust settings and provides a graph overview of all domains reviewed, highlighting the scores using a color scheme!

Overall Active Directory Architecture – Audit Trust Settings using PingCastle
PingCastle was built by Vincent Le Toux (co-author of Mimikatz) and aims to provide a "simple" security score of a domain configuration. Its motto is, "Get Active Directory Security at 80% in 20% of the time" and aims to provide pragmatic security recommendations. Amongst many controls, it also reviews the domain trust settings and provides a graph overview of all domains reviewed, highlighting the scores using a color scheme!

It's interesting to note that ALL security controls performed by PingCastle can be run without administrative access (this is one of their key principles).

You can freely download PingCastle on the following web site: https://www.pingcastle.com/



More Information?

Active Directory is typically a large component of lateral movement, but it's not everything. In the next section, we will focus more on concrete controls and hardening measures we can deploy to prevent the attacks from succeeding. For additional information regarding a secure AD architecture, please refer to:

Microsoft has done some amazing work in providing a document called "Best Practices for Securing Active Directory." It's available as a Windows Docs and lists a wide number of strategies and controls that can be implemented!

505

SANS also offers a full 6-day course on Windows Security – "SEC505: Securing Windows and PowerShell Automation" (by Jason Fossen). It includes additional strategies and details on how to properly secure a Windows environment!

SANS

SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 22

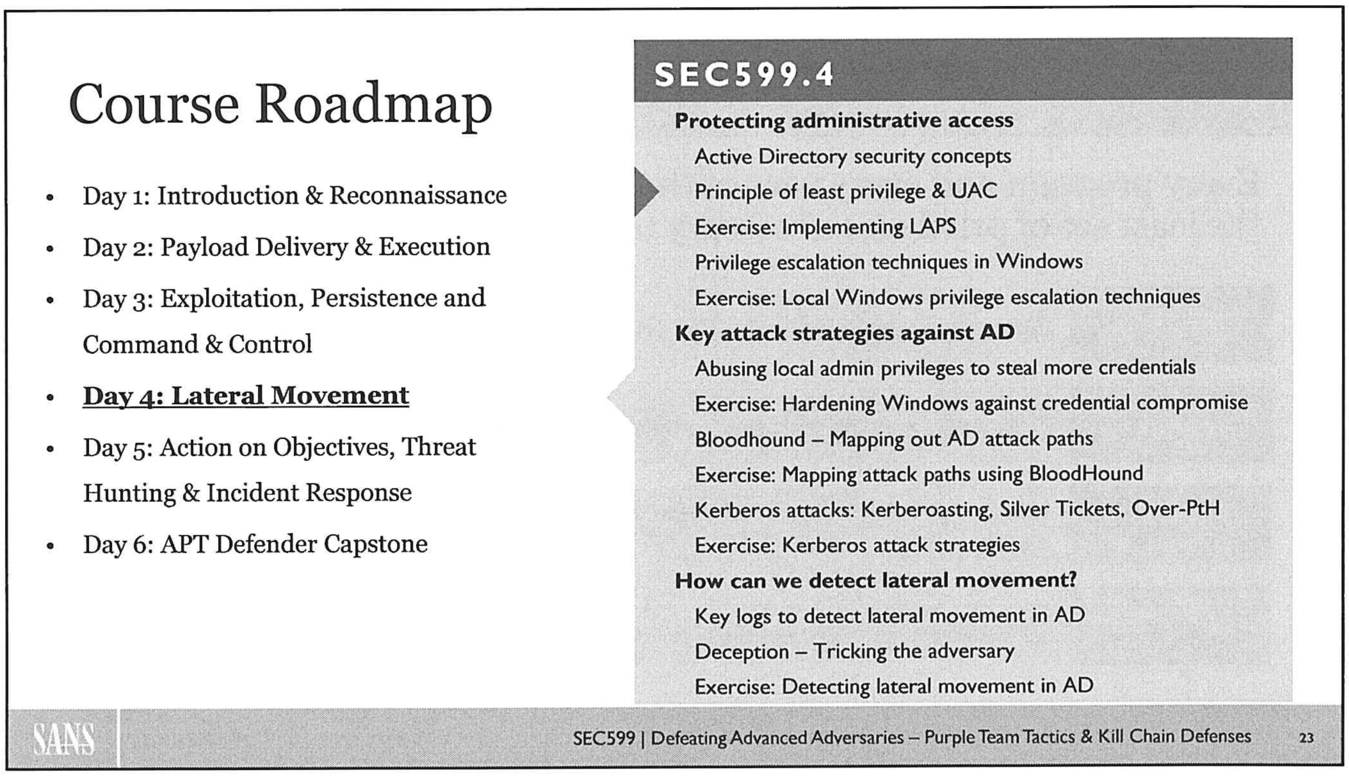
More Information?
Active Directory is typically a large component of lateral movement, but it's not everything. In the next section, we will focus more on concrete controls and hardening measures we can deploy to prevent the attacks from succeeding. For additional information regarding a secure AD architecture, please refer to:

- Microsoft has done some amazing work in providing a document called "Best Practices for Securing Active Directory." It's available as a Windows Docs and lists a wide number of strategies and controls that can be implemented! The document can be found here: https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/best-practices-for-securing-active-directory.

- SANS also offers a full 6-day course on Windows Security – "SEC505: Securing Windows and PowerShell Automation" (by Jason Fossen). It includes additional strategies and details on how to properly secure a Windows environment!

In our next section, we will implement practical controls aimed at hardening our Windows environment!

© 2019 Erik Van Buggenhout & Stephen Sims



Course Roadmap

- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- **Day 4: Lateral Movement**
- Day 5: Action on Objectives, Threat Hunting & Incident Response
- Day 6: APT Defender Capstone

SEC599.4

Protecting administrative access
- Active Directory security concepts
- Principle of least privilege & UAC
- Exercise: Implementing LAPS
- Privilege escalation techniques in Windows
- Exercise: Local Windows privilege escalation techniques

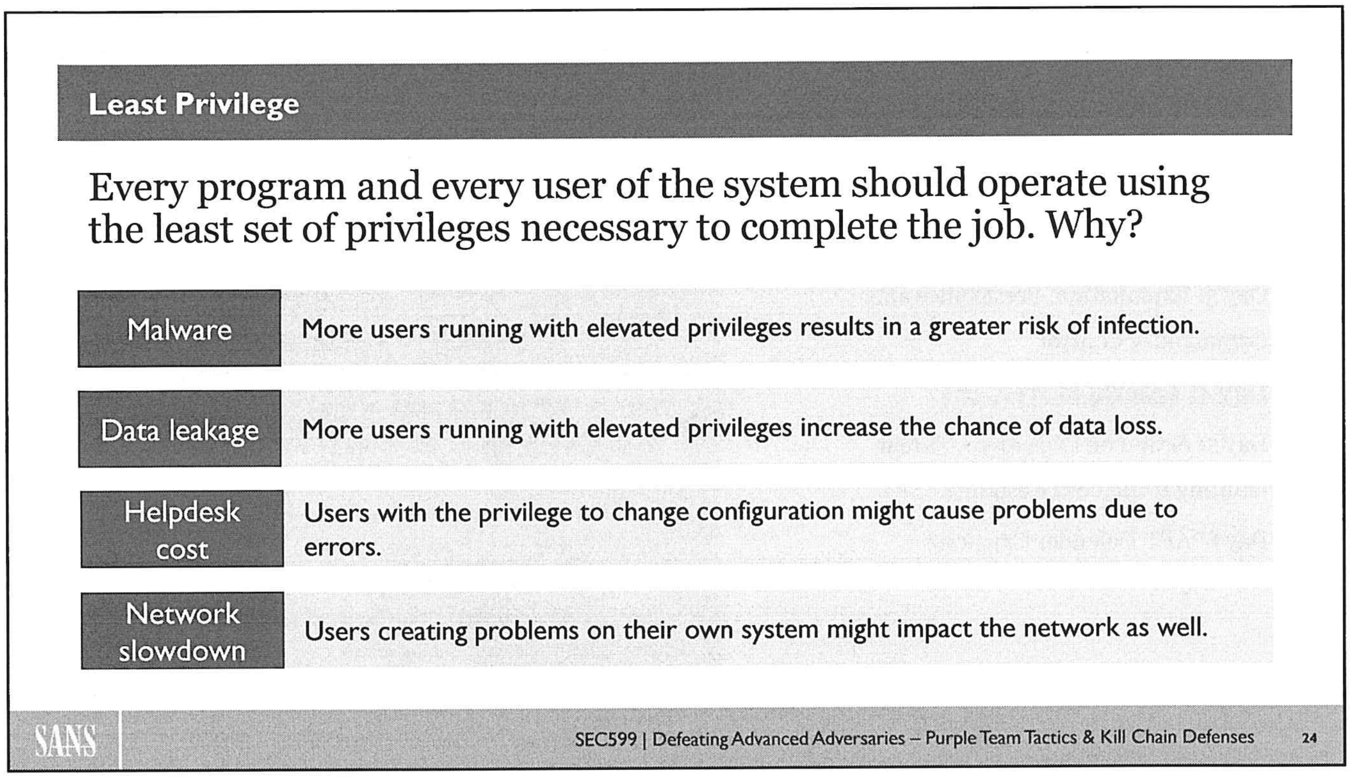
Key attack strategies against AD
- Abusing local admin privileges to steal more credentials
- Exercise: Hardening Windows against credential compromise
- Bloodhound – Mapping out AD attack paths
- Exercise: Mapping attack paths using BloodHound
- Kerberos attacks: Kerberoasting, Silver Tickets, Over-PtH
- Exercise: Kerberos attack strategies

How can we detect lateral movement?
- Key logs to detect lateral movement in AD
- Deception – Tricking the adversary
- Exercise: Detecting lateral movement in AD

SANS

SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 23

This page intentionally left blank.



Least Privilege

Every program and every user of the system should operate using the least set of privileges necessary to complete the job. Why?

Malware	More users running with elevated privileges results in a greater risk of infection.
Data leakage	More users running with elevated privileges increase the chance of data loss.
Helpdesk cost	Users with the privilege to change configuration might cause problems due to errors.
Network slowdown	Users creating problems on their own system might impact the network as well.

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 24

Least Privilege

The principle of least privilege could be defined as follows: "Every program and every user of the system should operate using the least set of privileges necessary to complete the job." - Saltzer and Schroeder. Granting permissions to a user beyond the scope of the necessary rights of an action can allow that user to obtain or change information in unwanted ways. Some specific examples of why implementing least privilege in your organization are listed below.

First of all, there's a greater risk of malware infection. Allowing users to download and/or install any application they want means they are allowed to install potential malware. Not only malware downloaded from the internet, but also malware making use of a USB key as an infection vector could be installed.

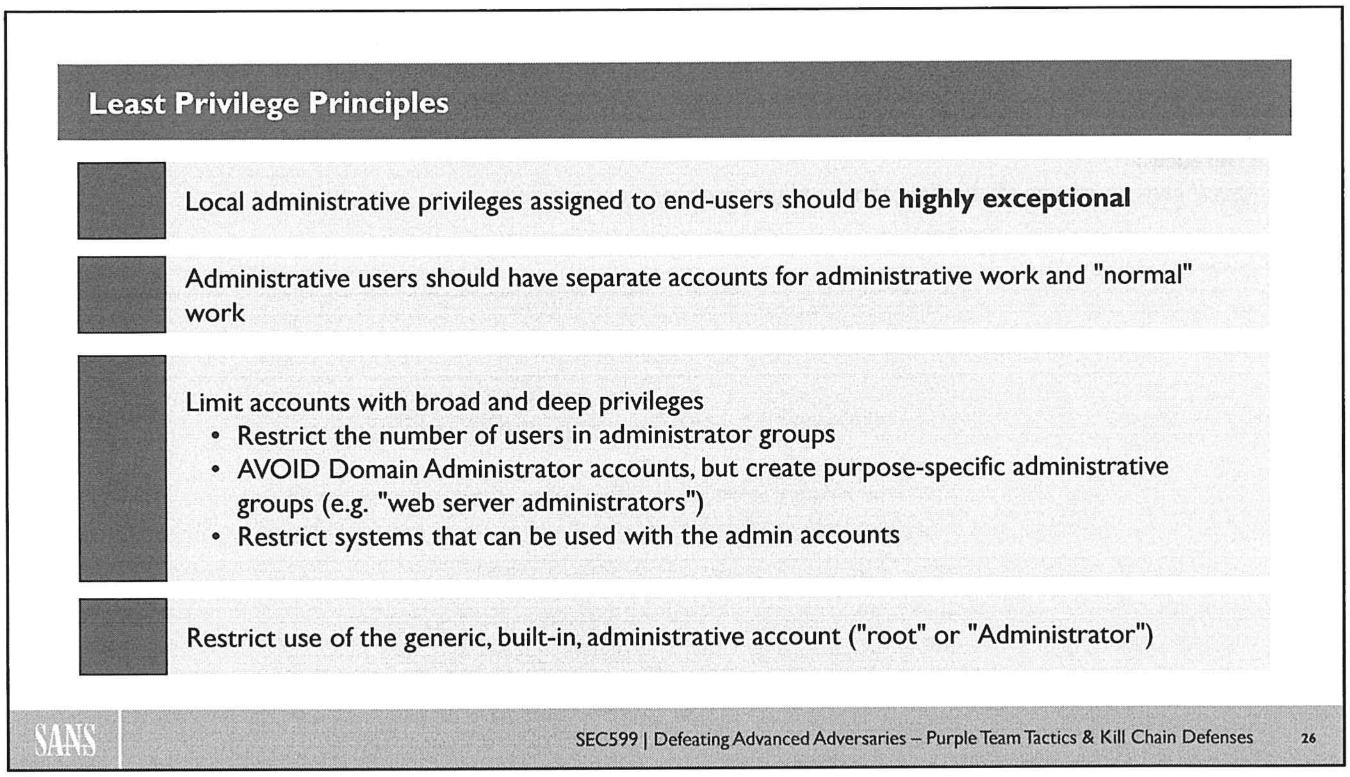
In attacks in which the target is an organization's intellectual property, accounts that have been granted powerful privileges within applications can be targeted to allow exfiltration of data. Although the accounts that have access to sensitive data may have been granted no elevated privileges in the domain or the operating system, accounts that can manipulate the configuration of an application or access the information the application provides present an increased risk. Additionally, adversaries could target the accounts that control access to the file stores, accounts that have direct access to the files, or even groups or roles that have access to the files. For example, if a file server is used to store contract documents and access is granted to the documents by the use of an Active Directory group, an adversary who can modify the membership of the group can add compromised accounts to the group and access the contract documents. In cases where access to documents is provided by applications such as SharePoint, adversaries can target the applications as described earlier.

Another potential issue is the increased cost of helpdesk and IT support. If users have the privileges to change system configurations, they might make changes that result in malfunctions. Be it accidentally, or on purpose (i.e., they think they know what they are doing), various things could go wrong, such as files or shares that are suddenly no longer accessible, network connection issues, etc. This could result in an increased number of calls to the IT helpdesk.

© 2019 Erik Van Buggenhout & Stephen Sims

The issue might not just be limited to their own system but could result in network issues for other users as well. For example, when changing properties of a networked file share, other users might lose their ability to view or modify files and thus have their ability to perform their work reduced.

References:
https://www.us-cert.gov/bsi/articles/knowledge/principles/least-privilege


Least Privilege Principles

Local administrative privileges assigned to end-users should be **highly exceptional**

Administrative users should have separate accounts for administrative work and "normal" work

Limit accounts with broad and deep privileges
- Restrict the number of users in administrator groups
- AVOID Domain Administrator accounts, but create purpose-specific administrative groups (e.g. "web server administrators")
- Restrict systems that can be used with the admin accounts

Restrict use of the generic, built-in, administrative account ("root" or "Administrator")

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 26

Least Privilege Principles
Unfortunately, the path of least resistance in many environments has proven to be the overuse of accounts with broad and deep privilege. Broad privileges are rights and permissions that allow an account to perform specific activities across a large cross-section of the environment, for example, an account that is administrator on all servers.

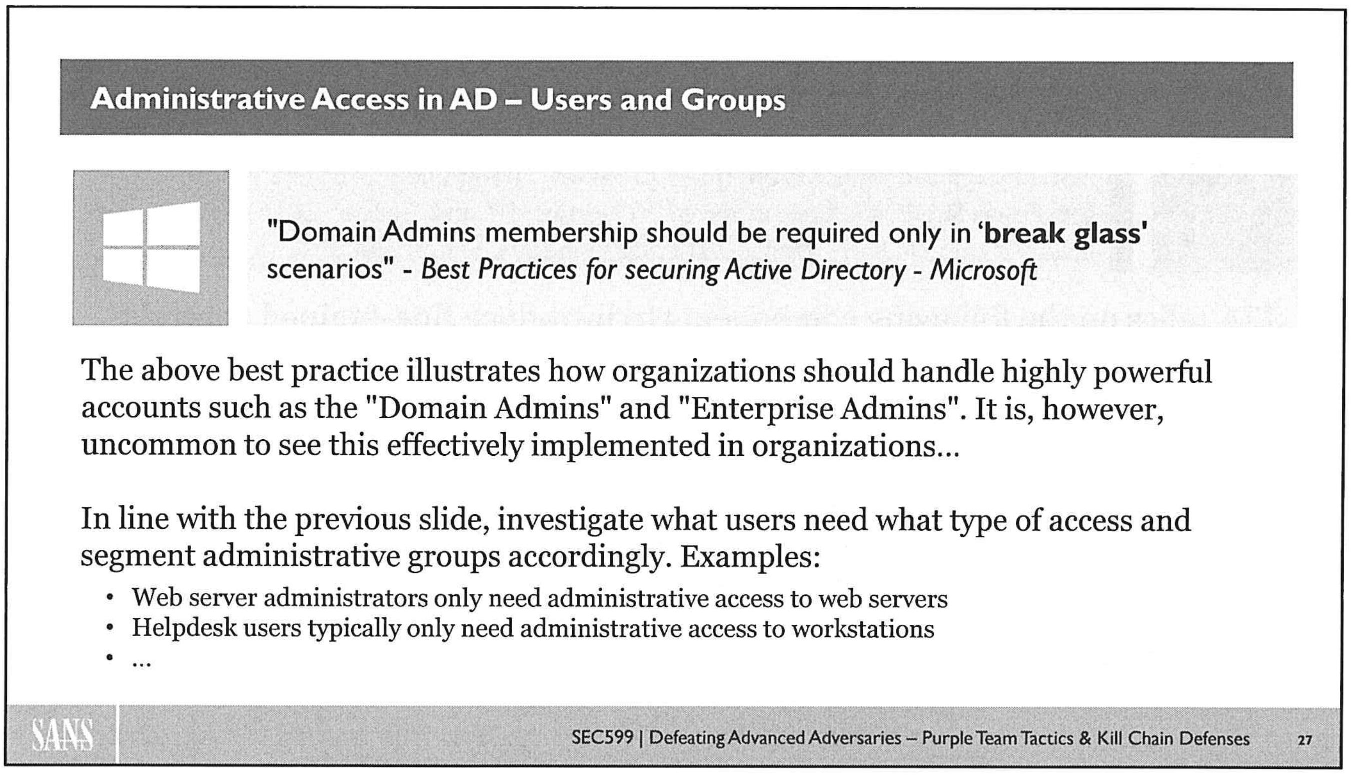
You should consider carefully whether users require administrative rights on their workstations. In most cases, the answer to the question: "Does this user need local administrative privileges?" is "no".

In case a user does need administrative privileges (again, which should be an exception), make sure these privileges are assigned to a dedicated privileged account. Furthermore, restrict use of the generic, built-in, administrative account ("root" or "Administrator")

Deep privileges are powerful privileges that are applied to a small number of users, such as giving an engineer full administrator rights on a certain server. Neither broad privilege nor deep privilege is necessarily dangerous, but when many accounts in the domain are permanently granted broad and deep privilege, an adversary can easily compromise one account and use it to move laterally through the network and potentially compromise additional accounts. We will see some examples later this day, during the "attacks on AD" section. Unsurprisingly, the number of administrators should be limited, along with the number of systems where on which admins can use their accounts.

Generally speaking, role-based access controls (RBAC) are a mechanism for grouping users and providing access to resources based on business rules. In the case of Active Directory, implementing RBAC for AD is the process of creating roles to which rights and permissions are delegated to allow members of the role to perform day-to-day administrative tasks without granting them excessive privilege. RBAC for Active Directory can be designed and implemented via native tooling and interfaces, by leveraging software you may already own, by purchasing third-party products, or any combination of these approaches.

© 2019 Erik Van Buggenhout & Stephen Sims



Administrative Access in AD – Users and Groups

"Domain Admins membership should be required only in **'break glass'** scenarios" - *Best Practices for securing Active Directory - Microsoft*

The above best practice illustrates how organizations should handle highly powerful accounts such as the "Domain Admins" and "Enterprise Admins". It is, however, uncommon to see this effectively implemented in organizations...

In line with the previous slide, investigate what users need what type of access and segment administrative groups accordingly. Examples:

- Web server administrators only need administrative access to web servers
- Helpdesk users typically only need administrative access to workstations
- ...

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 27

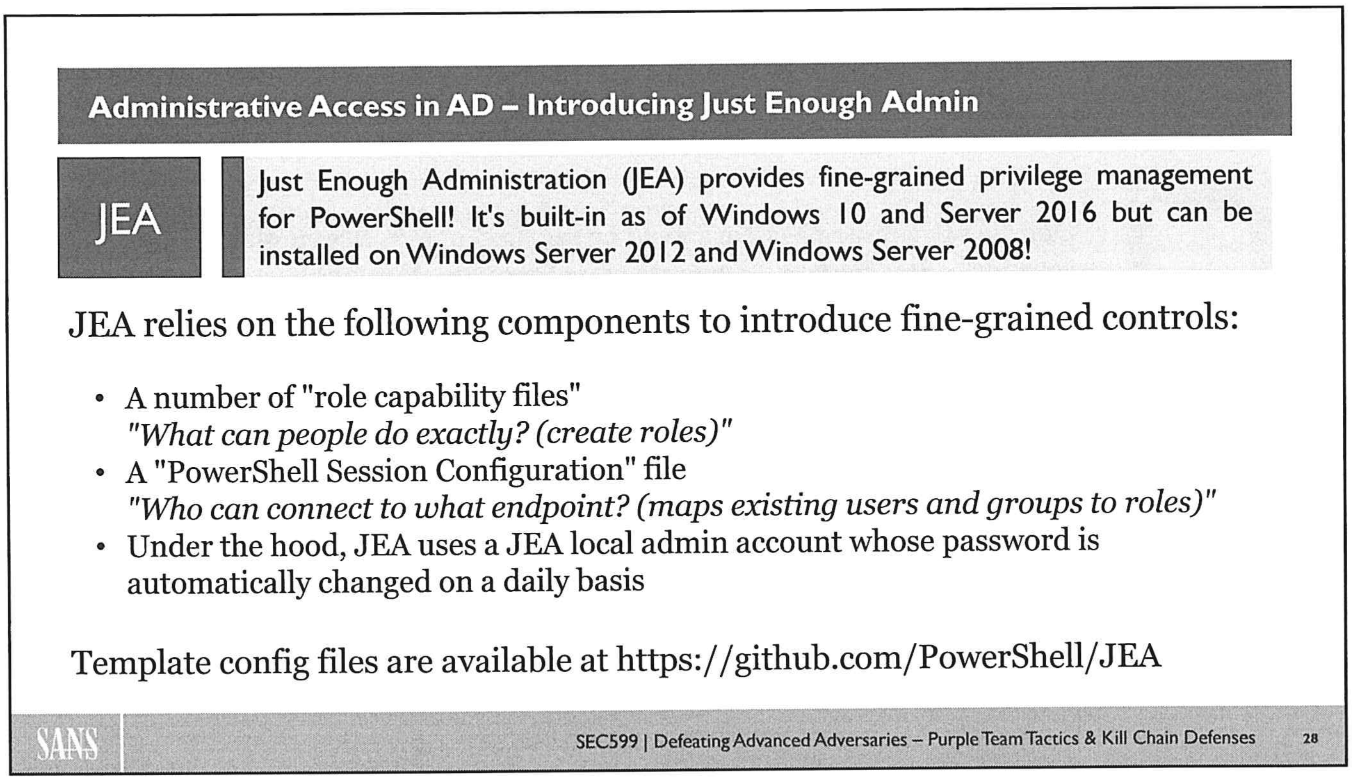
Administrative Access in AD – Users and Groups
An interesting extract from Microsoft's "Best Practices for securing Active Directory" is:

"Domain Admins membership should be required only in 'break glass' scenarios"

The above best practice illustrates how organizations should handle highly powerful accounts such as the "Domain Admins" and "Enterprise Admins". It is, however, uncommon to see this effectively implemented in organizations. At the time of writing, I am yet to encounter a client that has fully implemented this approach. It is, however, a fundamental control to highly limit the number and scope of administrative accounts.

In line with what we discussed on the previous slide, investigate what users need what type of access and segment administrative groups accordingly. This is linked to the entire IAM process and will require effort at the start, but it is the only way of building a truly secured environment. Some examples of such a segmentation include:

- Web server administrators only need administrative access to web servers, so they are placed in a "Web Server Administrators" group.
- Helpdesk users typically only need administrative access to workstations, so they are placed in a "Workstation Administrators" group.
- ...



Administrative Access in AD – Introducing Just Enough Admin

JEA

Just Enough Administration (JEA) provides fine-grained privilege management for PowerShell! It's built-in as of Windows 10 and Server 2016 but can be installed on Windows Server 2012 and Windows Server 2008!

JEA relies on the following components to introduce fine-grained controls:

- A number of "role capability files"
  "What can people do exactly? (create roles)"
- A "PowerShell Session Configuration" file
  "Who can connect to what endpoint? (maps existing users and groups to roles)"
- Under the hood, JEA uses a JEA local admin account whose password is automatically changed on a daily basis

Template config files are available at https://github.com/PowerShell/JEA

SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 28

Administrative Access in AD – Introducing Just Enough Admin
Just Enough Administration (JEA) provides fine-grained privilege management for PowerShell! It's built-in as of Windows 10 and Server 2016 but can be installed on Windows Server 2012 and Windows Server 2008! The idea is to implement fine-grained restrictions on what people can do on different machines. For this, it relies on two main components:

- "Role Capability Files" – These are configuration files that specify what actions can be performed by the different roles on the systems. This could include:

  - Running a script.
  - Starting or stopping a service.
  - Running programs.
  - …

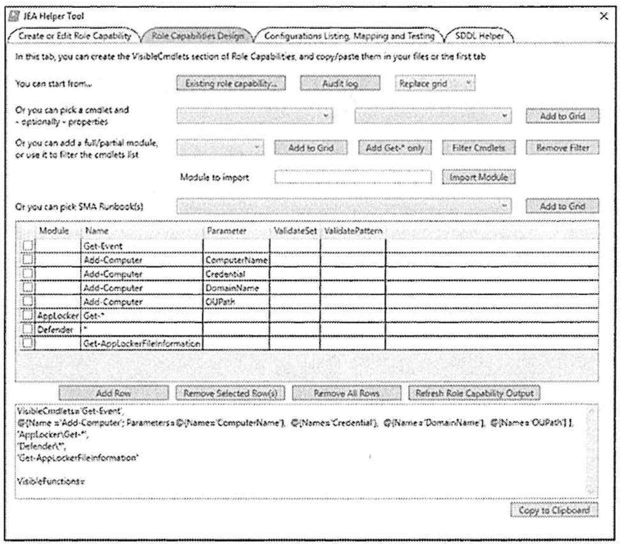
- "PowerShell Session Configuration File" – A PowerShell Session Configuration file is created per system. In this file, we can map users or groups to the roles defined in the "Role Capability Files".

- Under the hood, JEA actually uses a JEA-dedicated local administrator account whose password is changed on a daily basis (automatically).

Several template configuration files can be found here:
https://github.com/PowerShell/JEA

© 2019 Erik Van Buggenhout & Stephen Sims

Administrative Access in AD – Introducing Just Enough Admin – Example



In this tab, you can create the VisibleCmdlets section of Role Capabilities, and copy/paste them in your files or the first tab

You can start from...

Or you can pick a cmdlet and - optionally - properties

Or you can add a full/partial module, or use it to filter the cmdlets list

Module to import

Or you can pick SMA Runbook(s)

	Module	Name	Parameter	ValidateSet	ValidatePattern
☐		Get-Event			
☐		Add-Computer	ComputerName		
☐		Add-Computer	Credential		
☐		Add-Computer	DomainName		
☐		Add-Computer	OUPath		
☐	AppLocker	Get-*			
☐	Defender	*			
☐		Get-AppLockerFileInformation			

```
VisibleCmdlets='Get-Event',
@{Name ='Add-Computer'; Parameters=@{Name='ComputerName'}, @{Name='Credential'}, @{Name='DomainName'}, @{Name='OUPath'} },
'AppLocker\Get-*',
'Defender\*',
'Get-AppLockerFileInformation'

VisibleFunctions=
```

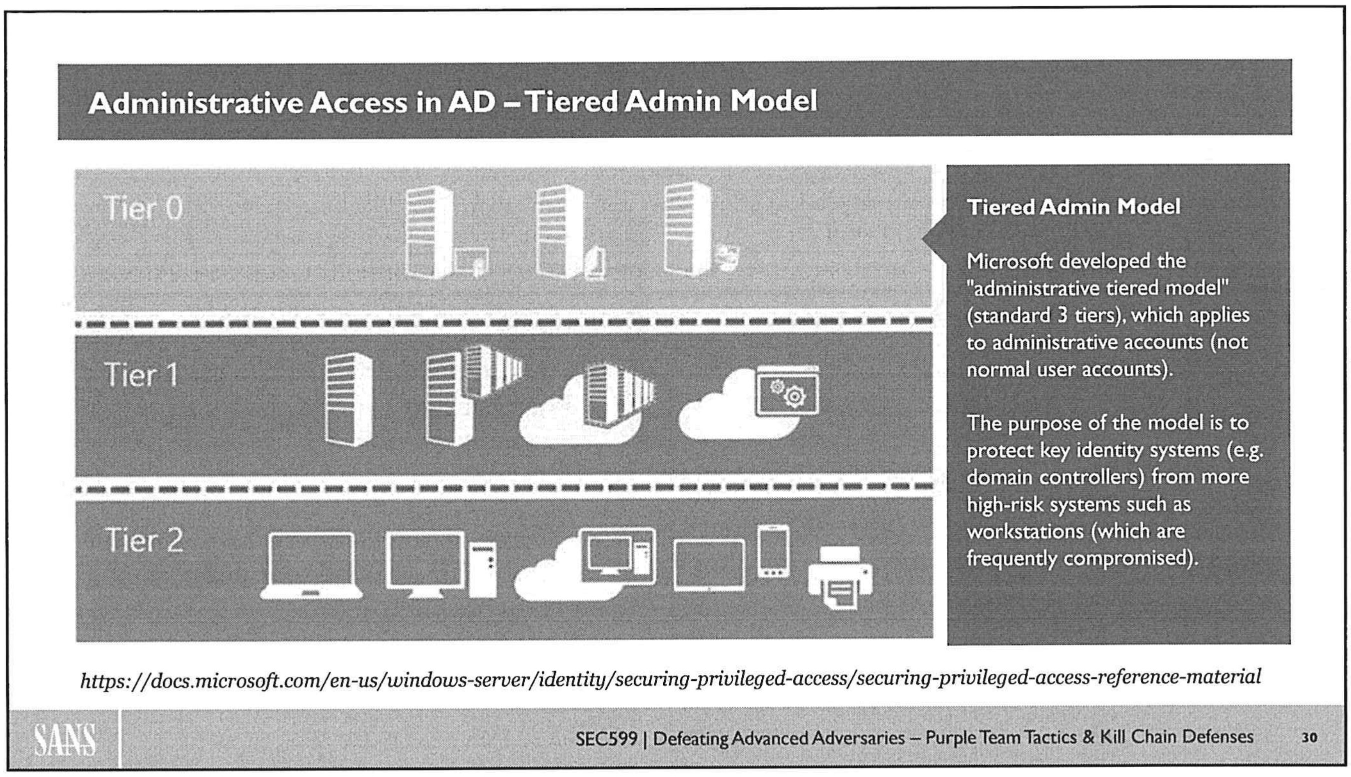
The screenshot on the left is Microsoft "JEA Helper" tool, which allows system administrators to craft session configuration files and role capability files in a nice GUI.

https://blogs.technet.microsoft.com/privatecloud/2015/12/20/introducing-the-updated-jea-helper-tool/

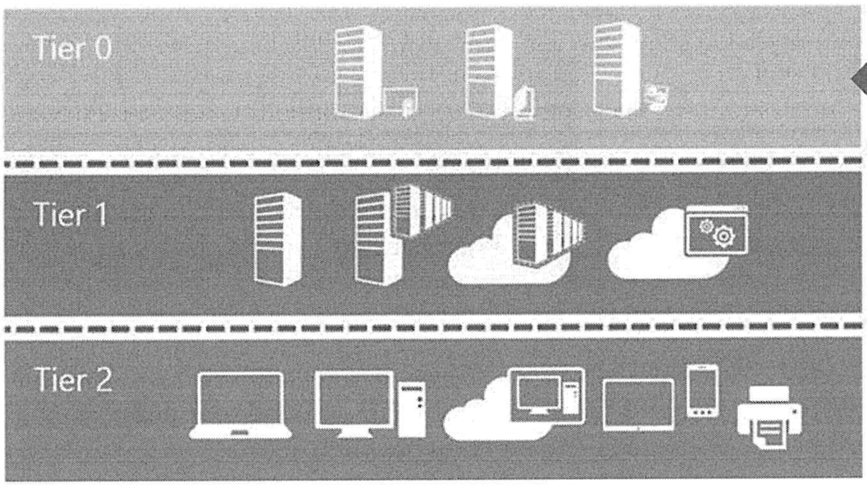
Administrative Access in AD – Introducing Just Enough Admin – Example

So how can we create these configuration files in an easy way? The screenshot on the slide introduces Microsoft's "JEA Helper" tool. This GUI tool was built to provide system administrators with an intuitive capability to develop JEA configuration files that would allow for "easy" management.

If you would like to learn more about Microsoft "JEA Helper" tool, please refer to the Microsoft TechNet blog: https://blogs.technet.microsoft.com/privatecloud/2015/12/20/introducing-the-updated-jea-helper-tool/



Administrative Access in AD – Tiered Admin Model



Tiered Admin Model

Microsoft developed the "administrative tiered model" (standard 3 tiers), which applies to administrative accounts (not normal user accounts).

The purpose of the model is to protect key identity systems (e.g. domain controllers) from more high-risk systems such as workstations (which are frequently compromised).

https://docs.microsoft.com/en-us/windows-server/identity/securing-privileged-access/securing-privileged-access-reference-material

SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 30

Administrative Access in AD – Tiered Admin Model
Microsoft developed the "administrative tiered model" (standard 3 tiers), which applies to administrative accounts (not normal user accounts). The purpose of the model is to protect key identity systems (e.g. domain controllers) from more high-risk systems such as workstations (which are frequently compromised).

The defined tiers are:

- **Tier 0**: Administrative access to directory services such as domain controllers, where central management is performed. Typical profiles include Domain Administrators.
- **Tier 1**: Administrative access to enterprise servers and applications, where sensitive data is typically centralized. Typical profiles include server administrators.
- **Tier 2**: Administrative access to workstations, where possibly sensitive data can be stolen. Typical profiles include Help / Service Desk personnel.

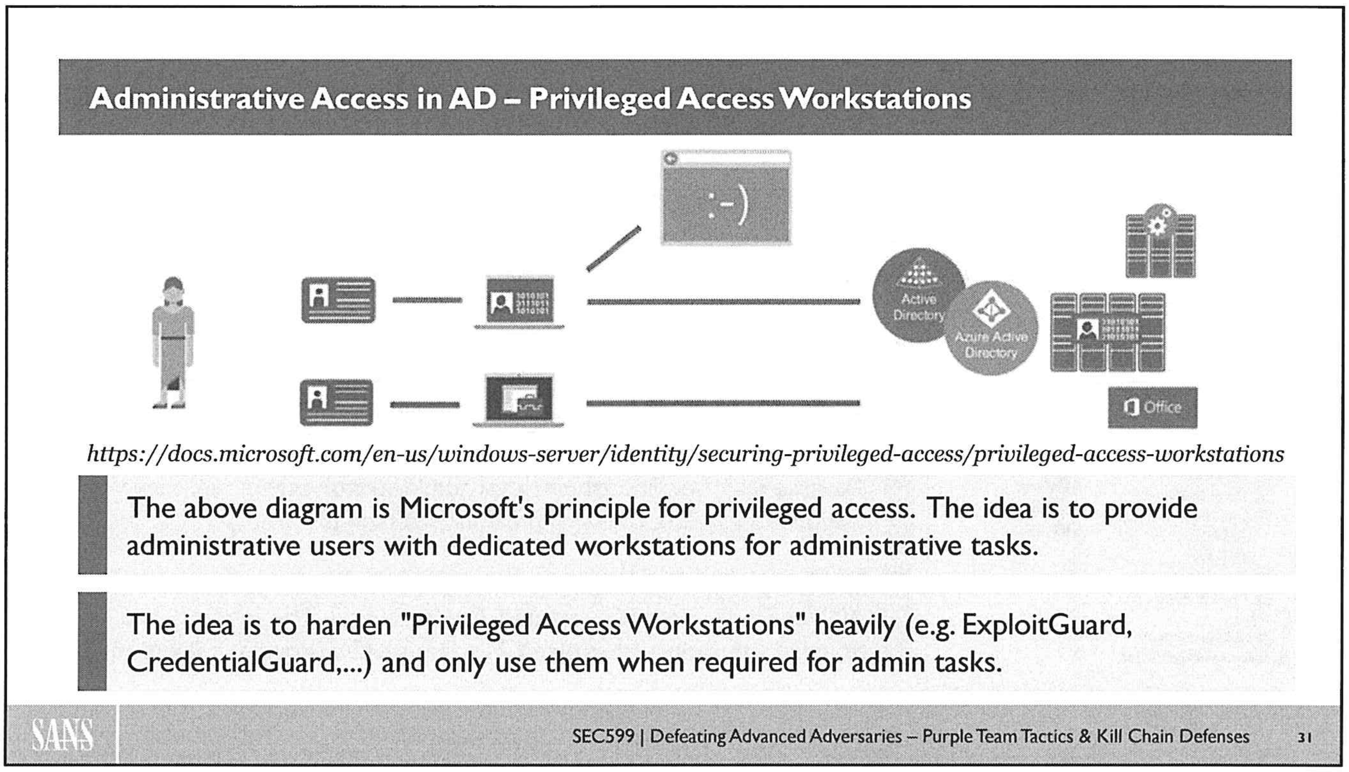
The full description can be found on Microsoft's knowledge base:

https://docs.microsoft.com/en-us/windows-server/identity/securing-privileged-access/securing-privileged-access-reference-material

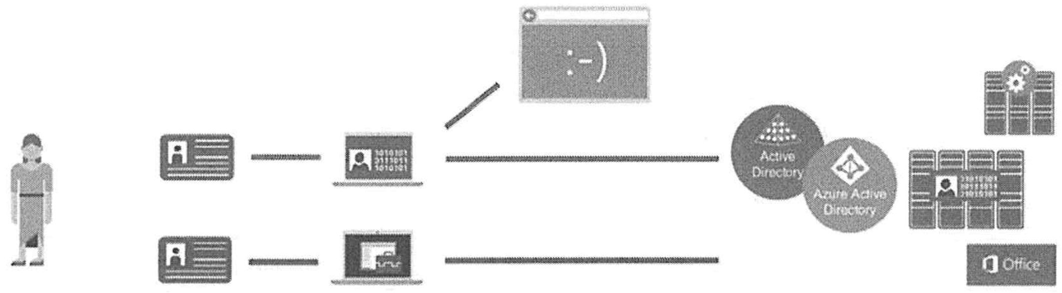
The "buffers" can be enforced using group policies. As a simple example, Domain Administrators should not be able to authenticate to tier 1 or tier 2 systems (e.g. member servers or workstations). Additional documentation can be found here:

https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/appendix-f--securing-domain-admins-groups-in-active-directory

© 2019 Erik Van Buggenhout & Stephen Sims



Administrative Access in AD – Privileged Access Workstations



https://docs.microsoft.com/en-us/windows-server/identity/securing-privileged-access/privileged-access-workstations

The above diagram is Microsoft's principle for privileged access. The idea is to provide administrative users with dedicated workstations for administrative tasks.

The idea is to harden "Privileged Access Workstations" heavily (e.g. ExploitGuard, CredentialGuard,...) and only use them when required for admin tasks.

SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 31

Administrative Access in AD – Privileged Access Workstations
The above diagram is Microsoft's principle for privileged access. The idea is to provide administrative users with dedicated workstations for administrative tasks. The idea is to harden "Privileged Access Workstations" heavily (e.g. ExploitGuard, CredentialGuard,...) and only use them when required for admin tasks.

Microsoft has built a document explaining how to build out a "Privileged Access Workstations Architecture". It includes a few phases:

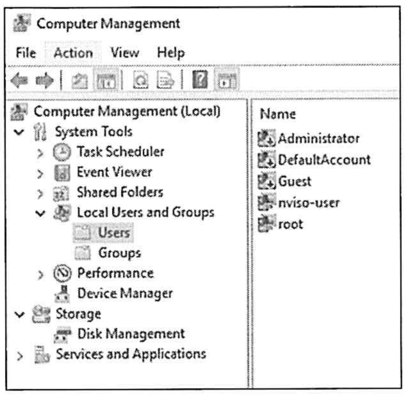
1. In a first phase, the model should be applied to the most critical user roles in the environment (Enterprise and Domain Administrators).
2. In a second phase, the model should be applied to all administrators in the environment.
3. In a third phase, the Privileged Access Workstation hardening principles can be further challenged and improved.

Please refer to the below Microsoft documentation:

https://docs.microsoft.com/en-us/windows-server/identity/securing-privileged-access/privileged-access-workstations

Administrative Access in AD – Local Users and Groups?

Next to domain-level users and groups, Windows endpoints (both servers and workstations) will also have a number of local accounts configured. The default accounts are "Administrator" and "Guest".



Local accounts are difficult to manage at enterprise level and should typically be avoided. Beware of forgotten local accounts!

Adversaries often first obtain local administrator access, and they will use this to escalate to domain (administrative) privileges.

Some organizations choose to set the same complex password for ALL local administrator accounts (for recovery). This is a very bad practice and should be avoided. Microsoft LAPS provides a solution.

Administrative Access in AD – Local Users and Groups?
Next to domain-level users and groups, Windows endpoints (both servers and workstations) will also have a number of local accounts configured. The default accounts are "Administrator" and "Guest".

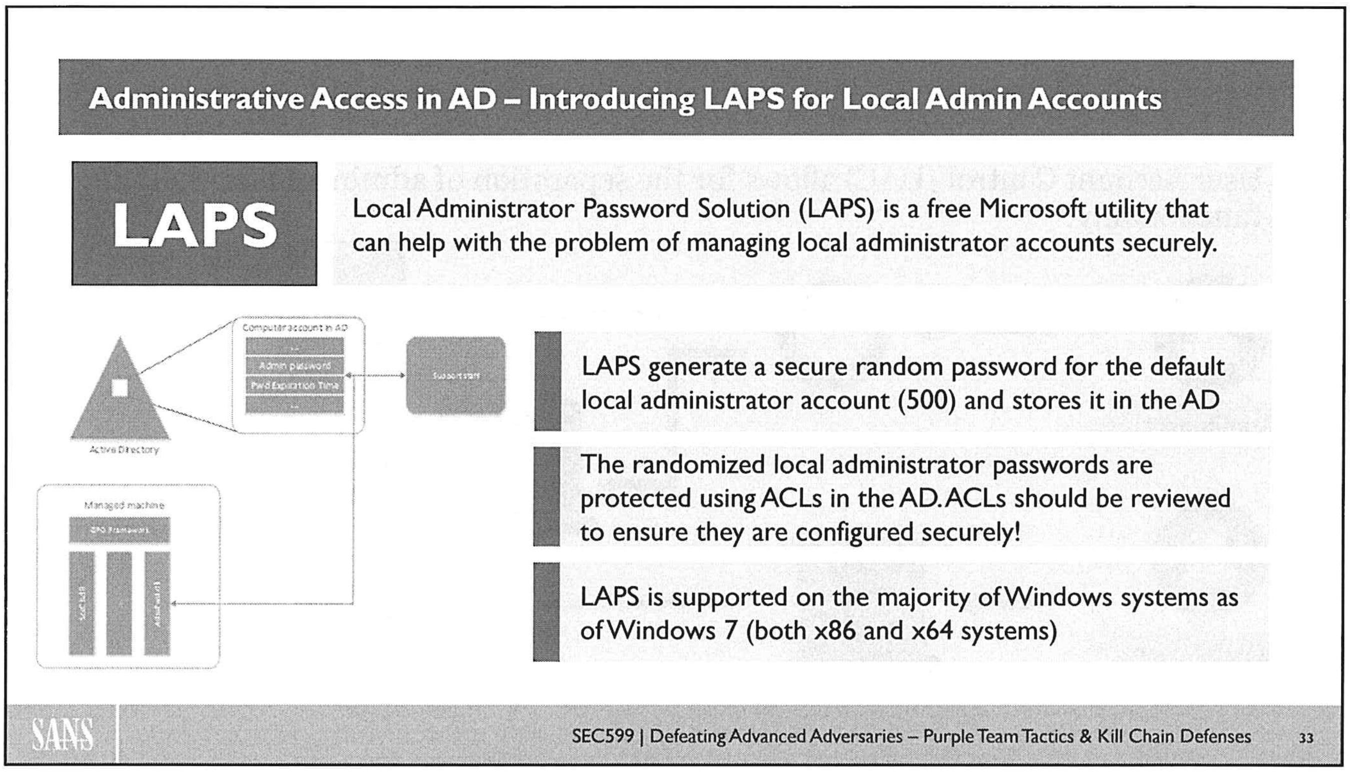
Once an AD environment is set up, there's a number of items to take into account:

- Local accounts are difficult to manage at enterprise level and should typically be avoided. Beware of forgotten local accounts! Even if local privileges have to be configured, it's better to implement and manage these centrally.

- Adversaries often first obtain local administrator access, and they will use this to escalate to domain (administrative) privileges. Again, it is vital we can limit administrative privileges as much as possible. There is typically no good reason for local end-users to have local administrator privileges.

- Some organizations choose to set the same complex password for ALL local administrator accounts (for recovery). This is a very bad practice and should be avoided. Microsoft LAPS provides a solution. Microsoft LAPS will generate a random password for the local administrator account for every machine in the network. This password is stored in Active Directory.

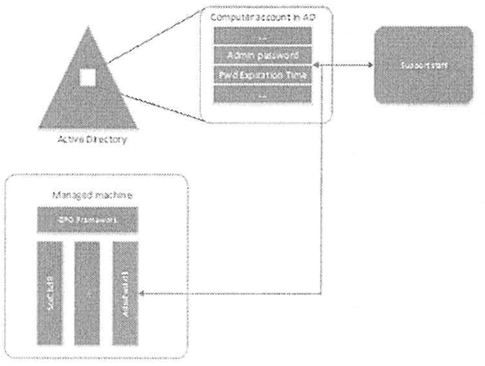
© 2019 Erik Van Buggenhout & Stephen Sims



Administrative Access in AD – Introducing LAPS for Local Admin Accounts

LAPS

Local Administrator Password Solution (LAPS) is a free Microsoft utility that can help with the problem of managing local administrator accounts securely.



- LAPS generate a secure random password for the default local administrator account (500) and stores it in the AD
- The randomized local administrator passwords are protected using ACLs in the AD. ACLs should be reviewed to ensure they are configured securely!
- LAPS is supported on the majority of Windows systems as of Windows 7 (both x86 and x64 systems)

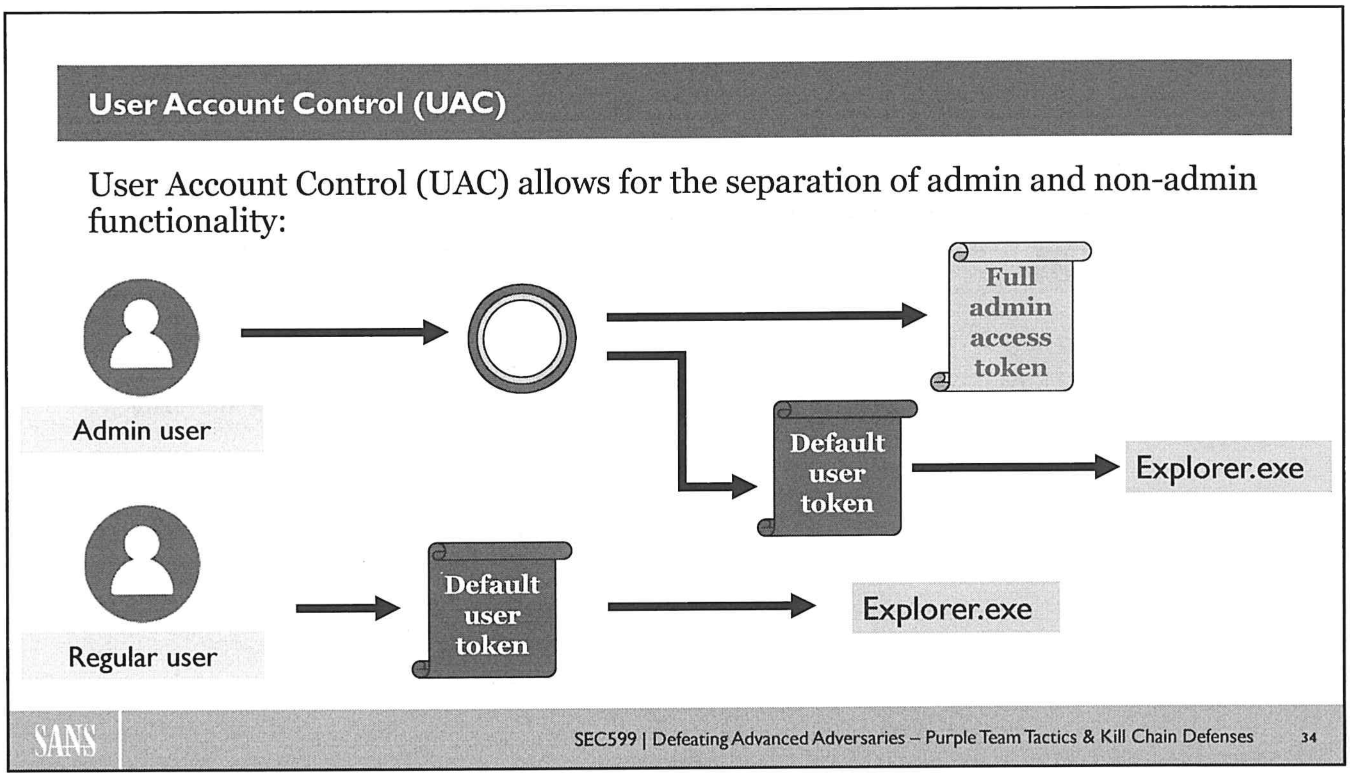
Administrative Access in AD – Introducing LAPS for Local Admin Accounts
Local Administrator Password Solution (LAPS) is a free Microsoft utility that can help with the problem of managing local administrator accounts securely. So how does it work?

LAPS generate a secure random password for the default local administrator account (500) and stores it in the AD. Note that LAPS can configure a maximum of two local administrator accounts per system.

The randomized local administrator passwords are protected using ACLs in the AD. The passwords are stored in cleartext in Active Directory, though, so the ACLs should be thoroughly reviewed to ensure they are configured securely!

LAPS is supported on the majority of Windows systems as of Windows 7 (both x86 and x64 systems).

More information on LAPS can be found at https://docs.microsoft.com/en-us/previous-versions/mt227395(v=msdn.10). Jason Fossen (a SANS Instructor highly focused on Windows security) provides an interesting alternative to LAPS here: https://cyber-defense.sans.org/blog/2013/08/01/reset-local-administrator-password-automatically-with-a-different-password-across-the-enterprise



User Account Control (UAC)

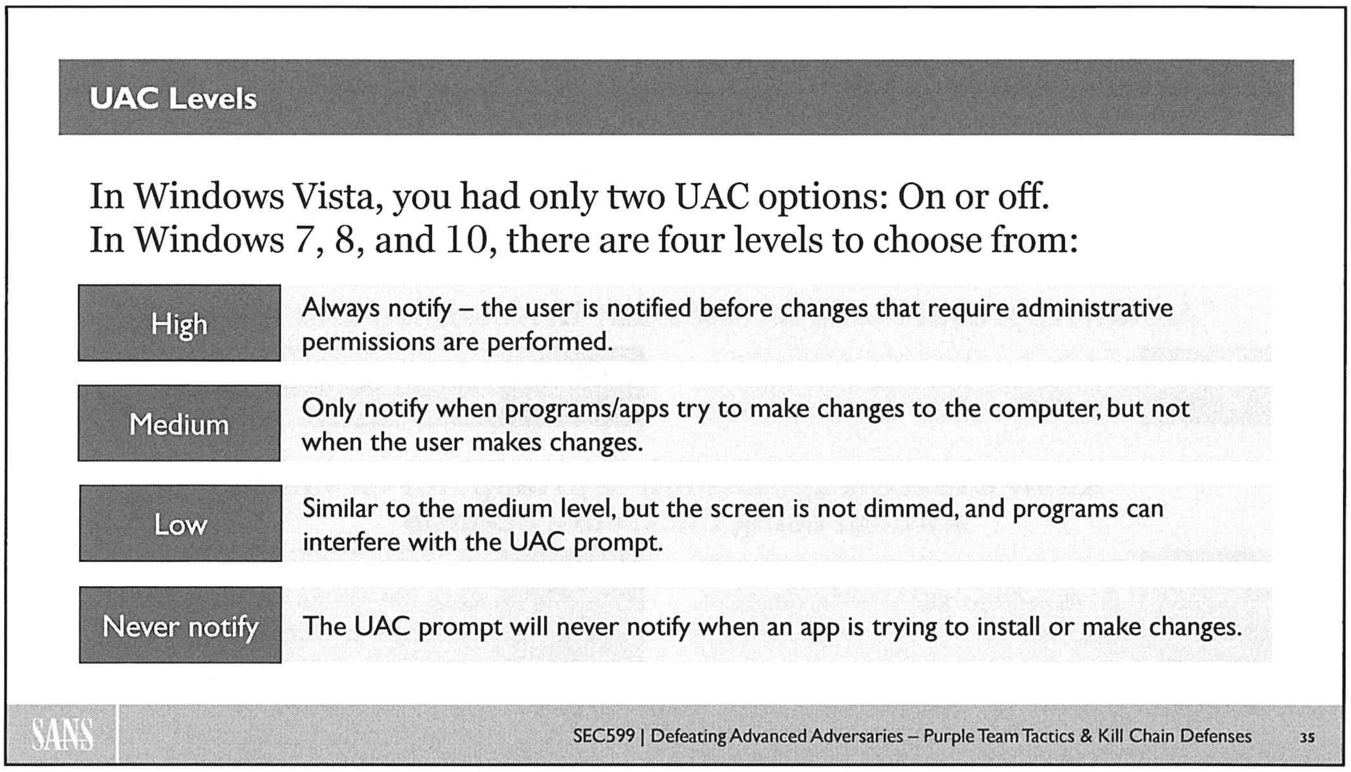
User Account Control (UAC) allows for the separation of admin and non-admin functionality:

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 34

User Account Control (UAC)
Microsoft has continued to improve the security in each version of Windows and, as a result, has reduced the operating system's attack surface. One of the most common and widespread security holes has been the granting of administrator privileges to otherwise standard users, often to resolve application and configuration issues. User Account Control (UAC) is a security component that enables users to perform common tasks as non-administrators or as administrators without having to switch users, log off, or use Run As. User accounts that are members of the local Administrators group run most applications as a standard user. By separating user and administrator functions, UAC helps users move toward using standard user rights by default. So, if a user is logged on as a local administrator, then UAC disables a user's administrator rights and prompts the user when their administrator rights are required. If a user logs on as a standard user, they are asked to provide the credentials of an administrator account if they attempt to perform any task that requires administrator rights. Ultimately the user has control over when to use administrator privileges as UAC does not have the ability to be controlled via centralized policy.

By default, standard users and administrators access resources and run apps in the security context of standard users. When a user logs on to a computer, the system creates an access token for that user. The access token contains information about the level of access that the user is granted, including specific security identifiers (SIDs) and Windows privileges. With the built-in UAC elevation component, standard users can easily perform an administrative task by entering valid credentials for a local administrator account. The default, built-in UAC elevation component for standard users is the credential prompt.

When an administrator logs on, two separate access tokens are created for the user: A standard user access token and an administrator access token. The standard user access token contains the same user-specific information as the administrator access token, but the administrative Windows privileges and SIDs are removed. The standard user access token is used to start apps that do not perform administrative tasks (standard user apps). The standard user access token is then used to display the desktop (explorer.exe). Explorer.exe is the parent process from which all other user-initiated processes inherit their access token. As a result, all apps run as a standard user unless a user provides consent or credentials to approve an app to use a full administrative access token.



UAC Levels

In Windows Vista, you had only two UAC options: On or off.
In Windows 7, 8, and 10, there are four levels to choose from:

High	Always notify – the user is notified before changes that require administrative permissions are performed.
Medium	Only notify when programs/apps try to make changes to the computer, but not when the user makes changes.
Low	Similar to the medium level, but the screen is not dimmed, and programs can interfere with the UAC prompt.
Never notify	The UAC prompt will never notify when an app is trying to install or make changes.

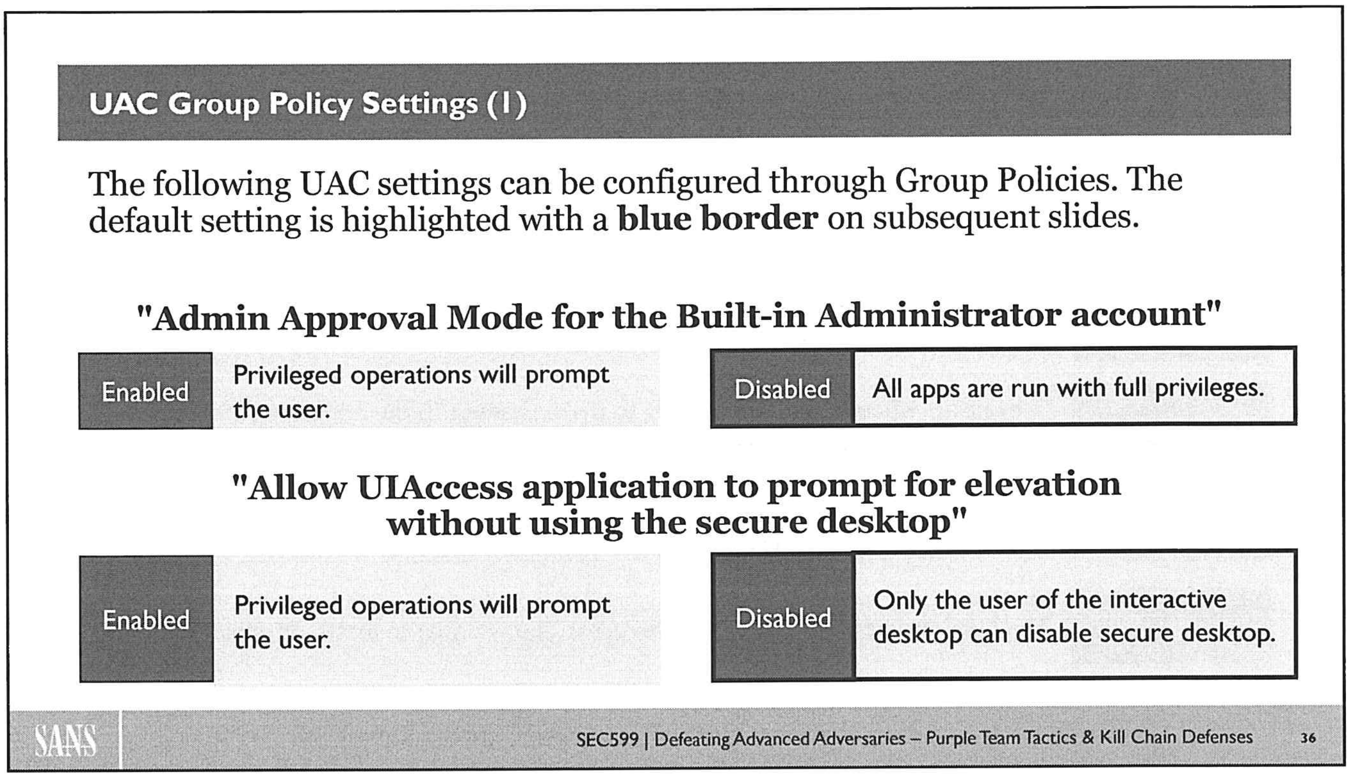
UAC Levels
Unlike Windows Vista, where you had only two options—UAC turned On or Off—in Windows 7, 8, and 10, there are four levels to choose from. The differences between them are the following:

- High: Always notify – at this level, you are notified before applications and users make changes that require administrative permissions. When a UAC prompt shows up, the desktop is dimmed as shown in the screenshot below. You must choose Yes or No before you can do anything else on the computer. Security Impact: This is the most secure setting and the most intruding as well.

- Medium: Notify me only when programs/apps try to make changes to my computer – this is the default level and UAC notifies you only before programs make changes that require administrative permissions. If you manually make changes to Windows, then a UAC prompt is not shown. This level is less intruding as it doesn't stop the user from making changes to the system, it only shows prompts if an application wants to make changes. When a UAC prompt is shown, the desktop is dimmed, and you must choose Yes or No before you can do anything else on your computer. Security Impact: This is less secure than the first setting because malicious programs can be created to simulate the keystrokes or mouse movements made by a user and change Windows settings.

- Low: Notify me only when programs/apps try to make changes to my computer (do not dim my desktop) – this level is identical to the one above except that when a UAC prompt is shown, the desktop is not dimmed, and other programs are able to interfere with it. Security Impact: This level is even less secure as it makes it easy for malicious programs to simulate keystrokes or mouse moves that interfere with the UAC prompt.

- Never notify: At this level, UAC is turned off and it doesn't offer any protection against unauthorized system changes. Security Impact: If you don't have a good security solution, you are very likely to encounter security issues with your PC. With UAC turned off, it is much easier for malicious programs to infect your computer and take control.



UAC Group Policy Settings (1)

The following UAC settings can be configured through Group Policies. The default setting is highlighted with a **blue border** on subsequent slides.

"Admin Approval Mode for the Built-in Administrator account"

Enabled	Privileged operations will prompt the user.
Disabled	All apps are run with full privileges.

"Allow UIAccess application to prompt for elevation without using the secure desktop"

Enabled	Privileged operations will prompt the user.
Disabled	Only the user of the interactive desktop can disable secure desktop.

UAC Group Policy Settings (1)
You can use security policies to configure how User Account Control works in your organization. They can be configured locally by using the Local Security Policy snap-in (secpol.msc) or configured for the domain, OU, or specific groups by Group Policy. In every slide, the default setting is marked with a blue border.

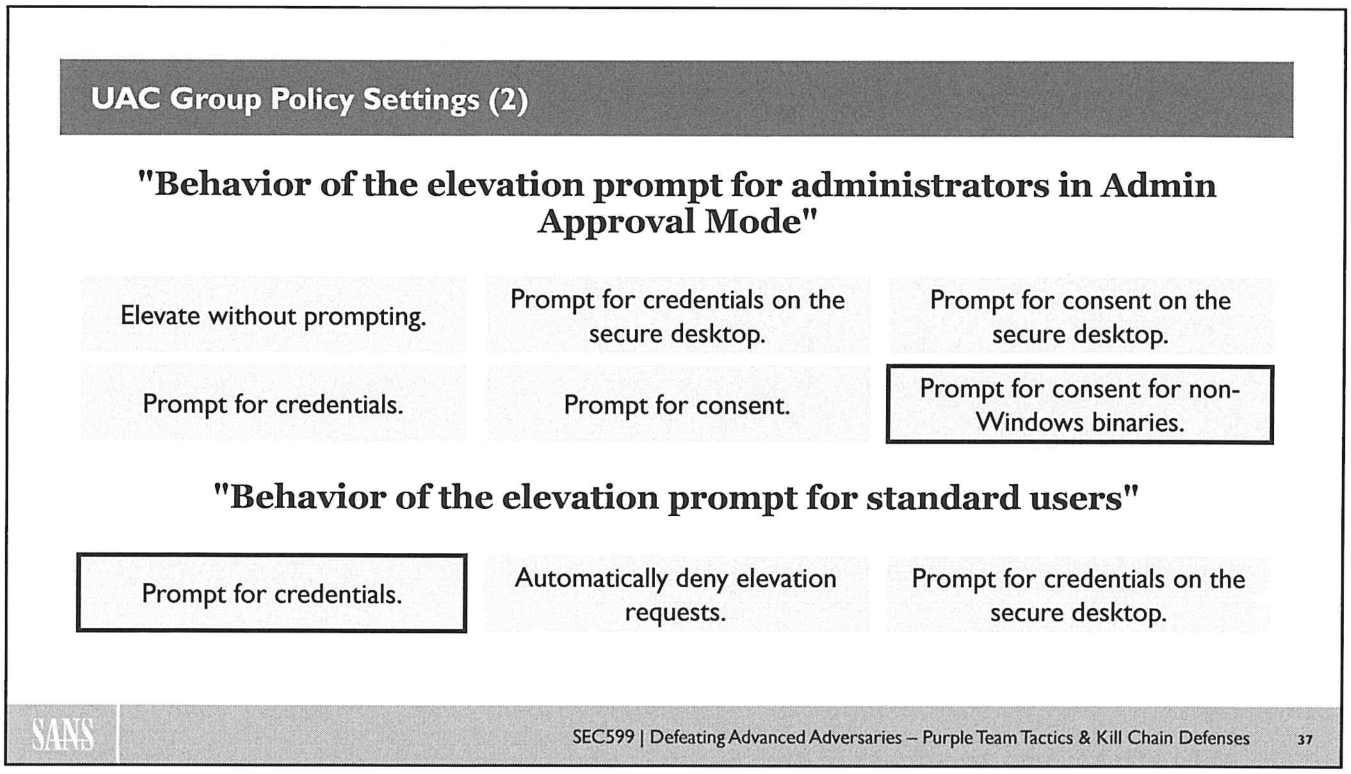
User Account Control: Admin Approval Mode for the Built-in Administrator account
This policy setting controls the behavior of Admin Approval Mode for the built-in Administrator account.

- **Enabled** – The built-in Administrator account uses Admin Approval Mode. By default, any operation that requires elevation of privilege will prompt the user to approve the operation.
- **Disabled (Default)** – The built-in Administrator account runs all applications with full administrative privilege.

Admin Approval Mode (AAM) is a UAC configuration in which a split user access token is created for an administrator. When an administrator logs in, the user is assigned two separate access tokens. Without AAM, an administrator account receives only one access token, which grants that administrator access to all Windows resources. As it is disabled by default, it is recommended to enable this option.

User Account Control: Allow UIAccess application to prompt for elevation without using the secure desktop
This policy setting controls whether User Interface Accessibility (UIAccess or UIA) programs can automatically disable the secure desktop for elevation prompts used by a standard user.

- **Enabled** – UIA programs, including Windows Remote Assistance, automatically disable the secure desktop for elevation prompts. If you do not disable the "User Account Control: Switch to the secure desktop when prompting for elevation" policy setting, the prompts appear on the interactive user's desktop instead of the secure desktop.
- **Disabled (Default)** – The secure desktop can be disabled only by the user of the interactive desktop or by disabling the "User Account Control: Switch to the secure desktop when prompting for elevation" policy setting.



UAC Group Policy Settings (2)

"Behavior of the elevation prompt for administrators in Admin Approval Mode"

Elevate without prompting.	Prompt for credentials on the secure desktop.	Prompt for consent on the secure desktop.
Prompt for credentials.	Prompt for consent.	**[✓ Prompt for consent for non-Windows binaries.]**

"Behavior of the elevation prompt for standard users"

[✓ Prompt for credentials.]	Automatically deny elevation requests.	Prompt for credentials on the secure desktop.

UAC Group Policy Settings (2)

User Account Control: Behavior of the elevation prompt for administrators in Admin Approval Mode

This policy setting controls the behavior of the elevation prompt for administrators.

- **Elevate without prompting** – Allows privileged accounts to perform an operation that requires elevation without requiring consent or credentials. Note: Use this option only in the most constrained environments.
- **Prompt for credentials on the secure desktop** – When an operation requires elevation of privilege, the user is prompted on the secure desktop to enter a privileged username and password. If the user enters valid credentials, the operation continues with the user's highest available privilege.
- **Prompt for consent on the secure desktop** – When an operation requires elevation of privilege, the user is prompted on the secure desktop to select either Permit or Deny. If the user selects Permit, the operation continues with the user's highest available privilege.
- **Prompt for credentials** – When an operation requires elevation of privilege, the user is prompted to enter an administrative username and password. If the user enters valid credentials, the operation continues with the applicable privilege.
- **Prompt for consent** – When an operation requires elevation of privilege, the user is prompted to select either Permit or Deny. If the user selects Permit, the operation continues with the user's highest available privilege.
- **Prompt for consent for non-Windows binaries** (Default) – When an operation for a non-Microsoft application requires elevation of privilege, the user is prompted on the secure desktop to select either Permit or Deny. If the user selects Permit, the operation continues with the user's highest available privilege.

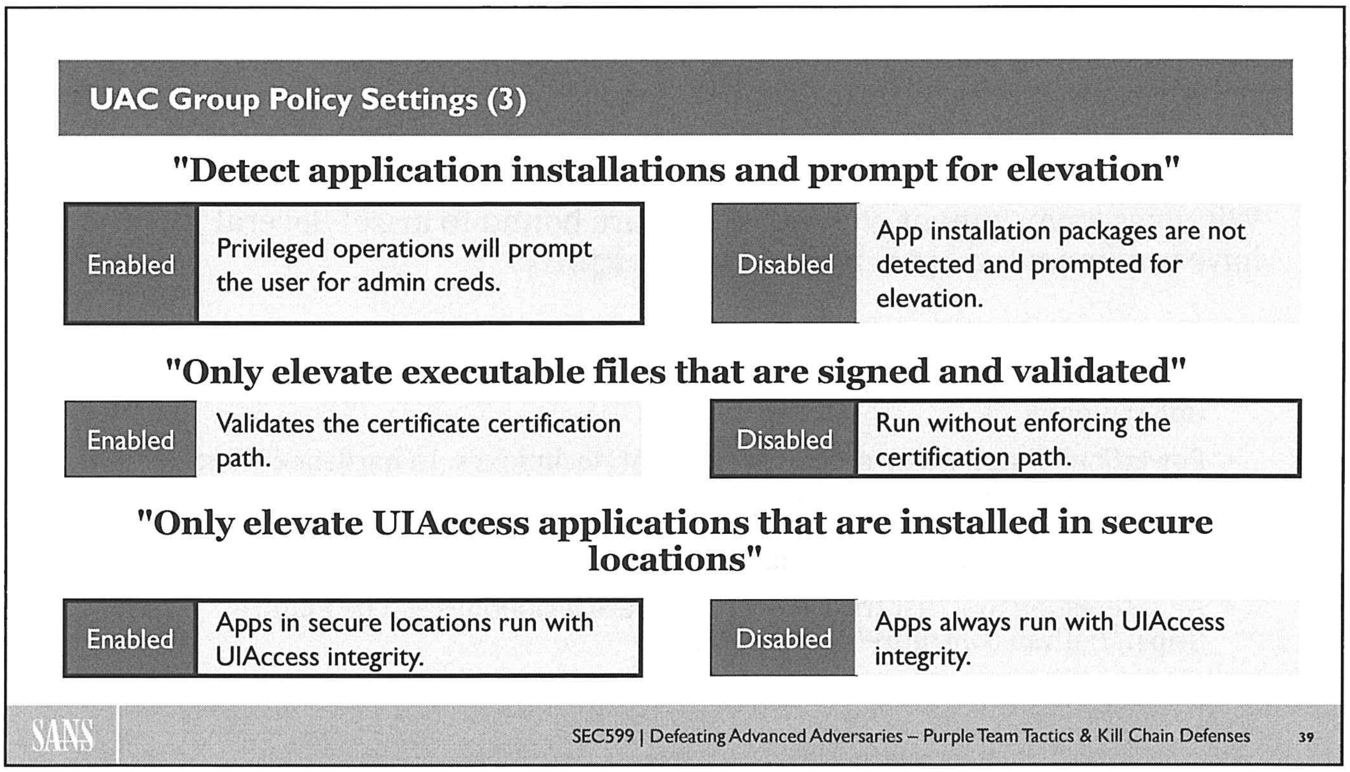
User Account Control: Behavior of the elevation prompt for standard users

This policy setting controls the behavior of the elevation prompt for standard users.

- **Prompt for credentials** (Default) – When an operation requires elevation of privilege, the user is prompted to enter an administrative username and password. If the user enters valid credentials, the operation continues with the applicable privilege.

- **Automatically deny elevation requests** – When an operation requires elevation of privilege, a configurable access denied error message is displayed. An enterprise that is running desktops as standard user may choose this setting to reduce help desk calls.
- **Prompt for credentials on the secure desktop** – When an operation requires elevation of privilege, the user is prompted on the secure desktop to enter a different username and password. If the user enters valid credentials, the operation continues with the applicable privilege.

© 2019 Erik Van Buggenhout & Stephen Sims



UAC Group Policy Settings (3)

"Detect application installations and prompt for elevation"

Enabled	Privileged operations will prompt the user for admin creds.	Disabled	App installation packages are not detected and prompted for elevation.

"Only elevate executable files that are signed and validated"

Enabled	Validates the certificate certification path.	Disabled	Run without enforcing the certification path.

"Only elevate UIAccess applications that are installed in secure locations"

Enabled	Apps in secure locations run with UIAccess integrity.	Disabled	Apps always run with UIAccess integrity.

SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 39

UAC Group Policy Settings (3)

User Account Control: Detect application installations and prompt for elevation

This policy setting controls the behavior of application installation detection for the computer.

- **Enabled** (Default) – When an app installation package is detected that requires elevation of privilege, the user is prompted to enter an administrative username and password. If the user enters valid credentials, the operation continues with the applicable privilege.
- **Disabled** – App installation packages are not detected and prompted for elevation. Enterprises that are running standard user desktops and use delegated installation technologies, such as Group Policy or System Center Configuration Manager should disable this policy setting. In this case, installer detection is unnecessary.

User Account Control: Only elevate executable files that are signed and validated

This policy setting enforces public key infrastructure (PKI) signature checks for any interactive applications that request elevation of privilege. Enterprise administrators can control which applications are allowed to run by adding certificates to the Trusted Publishers certificate store on local computers.

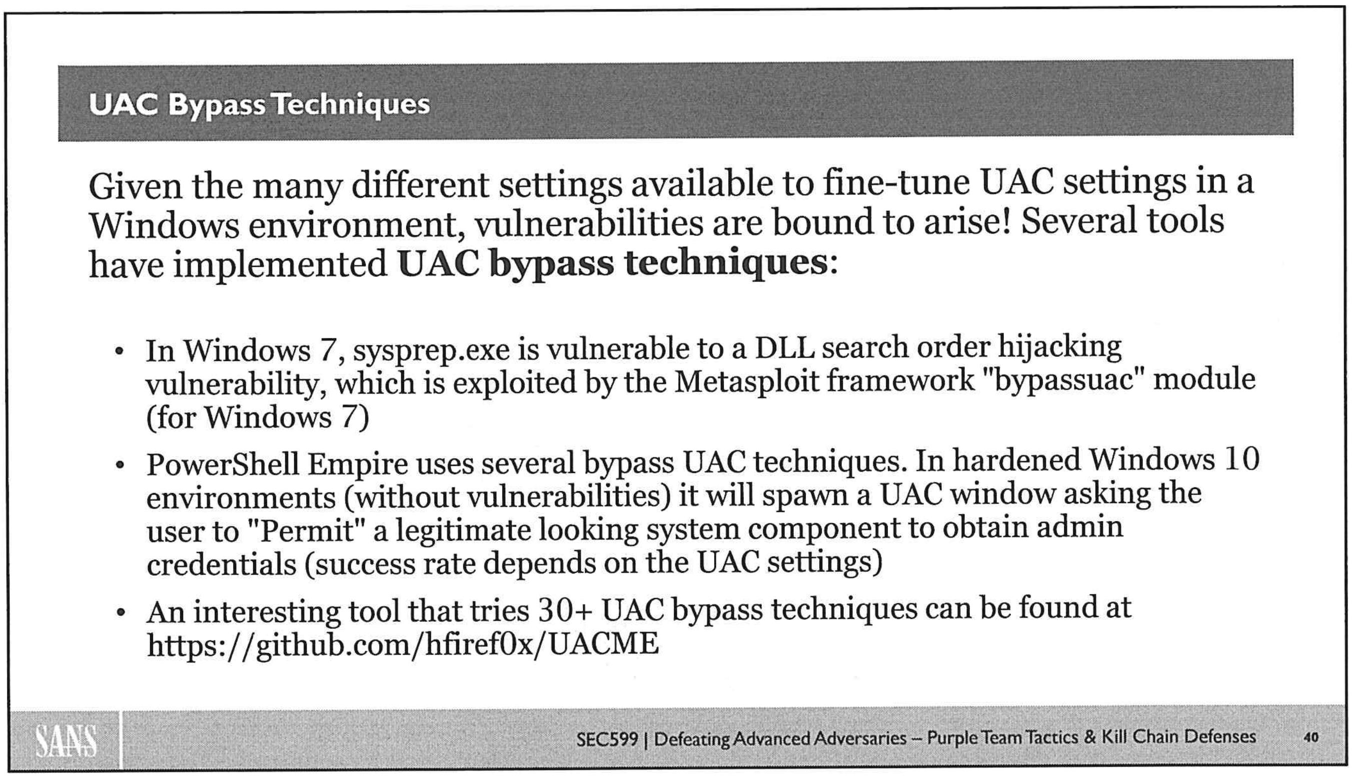
- **Enabled** – Enforces certificate certification path validation for a given executable file before it is permitted to run.
- **Disabled** (Default) – Does not enforce the certificate certification path validation before a given executable file is permitted to run.

User Account Control: Only elevate UIAccess applications that are installed in secure locations

This policy setting controls whether applications that request to run with a User Interface Accessibility (UIAccess) integrity level must reside in a secure location in the filesystem. Secure locations are limited to the following: - ...\Program Files\, including subfolders - ...\Windows\system32\ - ...\Program Files (x86)\, including subfolders for 64-bit versions of Windows+

Note: Windows enforces a digital signature check on any interactive app that requests to run with a UIAccess integrity level regardless of the state of this security setting.

- **Enabled** (Default) – If an app resides in a secure location in the filesystem, it runs only with UIAccess integrity.
- **Disabled** – An app runs with UIAccess integrity, even if it does not reside in a secure location in the filesystem.



UAC Bypass Techniques

Given the many different settings available to fine-tune UAC settings in a Windows environment, vulnerabilities are bound to arise! Several tools have implemented **UAC bypass techniques**:

- In Windows 7, sysprep.exe is vulnerable to a DLL search order hijacking vulnerability, which is exploited by the Metasploit framework "bypassuac" module (for Windows 7)
- PowerShell Empire uses several bypass UAC techniques. In hardened Windows 10 environments (without vulnerabilities) it will spawn a UAC window asking the user to "Permit" a legitimate looking system component to obtain admin credentials (success rate depends on the UAC settings)
- An interesting tool that tries 30+ UAC bypass techniques can be found at https://github.com/hfiref0x/UACME

SANS

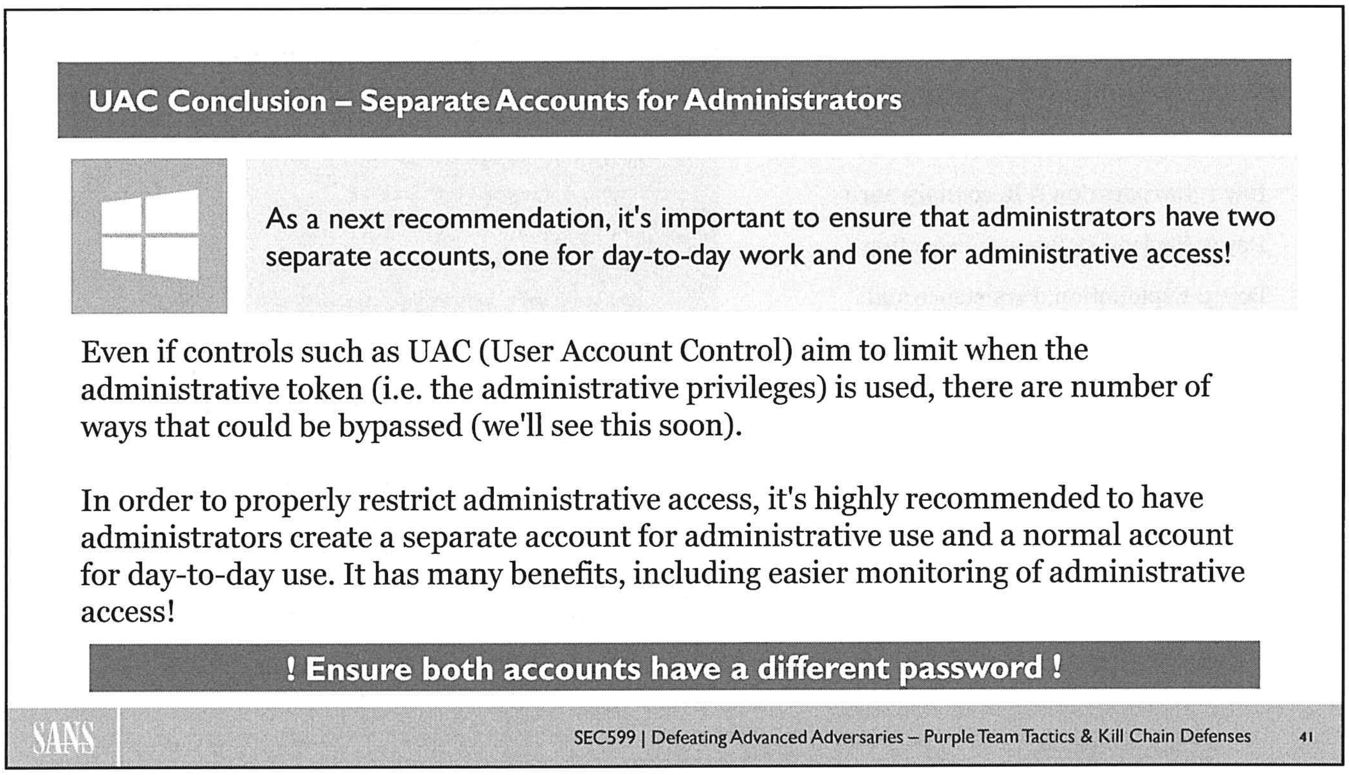
SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 40

UAC Bypass Techniques
Given the many different settings available to fine-tune UAC settings in a Windows environment, vulnerabilities due to misconfigurations are bound to arise! Several tools have implemented UAC bypass techniques:

- In Windows 7, sysprep.exe (used to configure a Windows installation upon install) is vulnerable to a DLL search order hijacking vulnerability, which is exploited by the Metasploit framework "bypassuac" module (for Windows 7).
- PowerShell Empire uses several bypass UAC techniques. In hardened Windows 10 environments (without vulnerabilities) it will spawn a UAC window asking the user to "Permit" a legitimate looking system component to obtain admin credentials (success rate depends on the UAC settings).
- An interesting tool that tries 30+ UAC bypass techniques can be found at https://github.com/hfiref0x/UACME .

We will attempt to use these tools in an upcoming exercise!

© 2019 Erik Van Buggenhout & Stephen Sims



UAC Conclusion – Separate Accounts for Administrators

As a next recommendation, it's important to ensure that administrators have two separate accounts, one for day-to-day work and one for administrative access!

Even if controls such as UAC (User Account Control) aim to limit when the administrative token (i.e. the administrative privileges) is used, there are number of ways that could be bypassed (we'll see this soon).

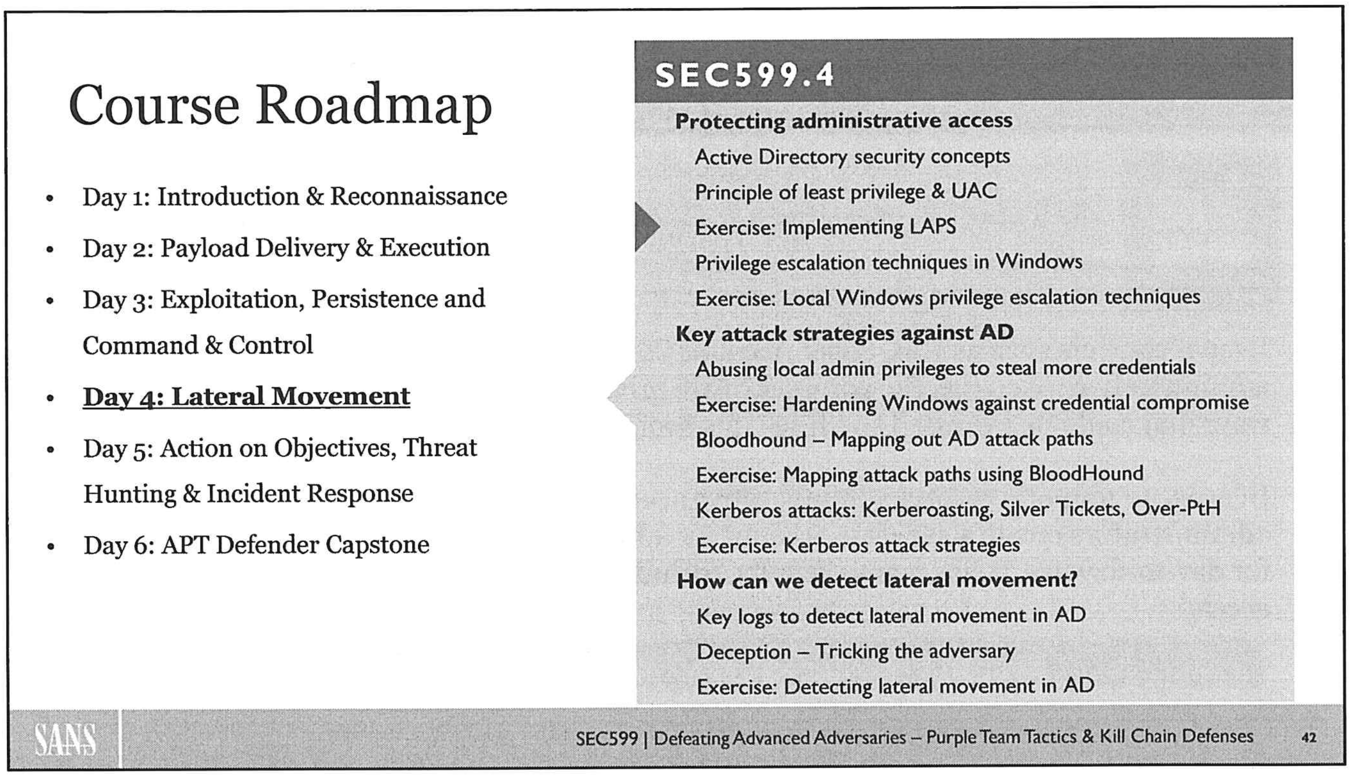
In order to properly restrict administrative access, it's highly recommended to have administrators create a separate account for administrative use and a normal account for day-to-day use. It has many benefits, including easier monitoring of administrative access!

! Ensure both accounts have a different password !

UAC Conclusion – Separate Accounts for Administrators
As a next recommendation, it's important to ensure that administrators have two separate accounts, one for day-to-day work and one for administrative access! Although this slightly increases complexity, it greatly hinders typical lateral movement strategies where IT administrators are tempted to open a malicious payload, after which their credentials are abused to pivot through the network.

As cyber security experts, you might make the reflection: "But User Account Control (UAC) already ensures administrative privileges are not used." While this is partially true, it's important to note that we've seen several examples of how UAC can be bypassed. Furthermore, separate accounts also facilitate logging and monitoring of privileged account use.

Finally, it's important to highlight that when an administrator creates two separate accounts, a different password should be used for each account.



Course Roadmap

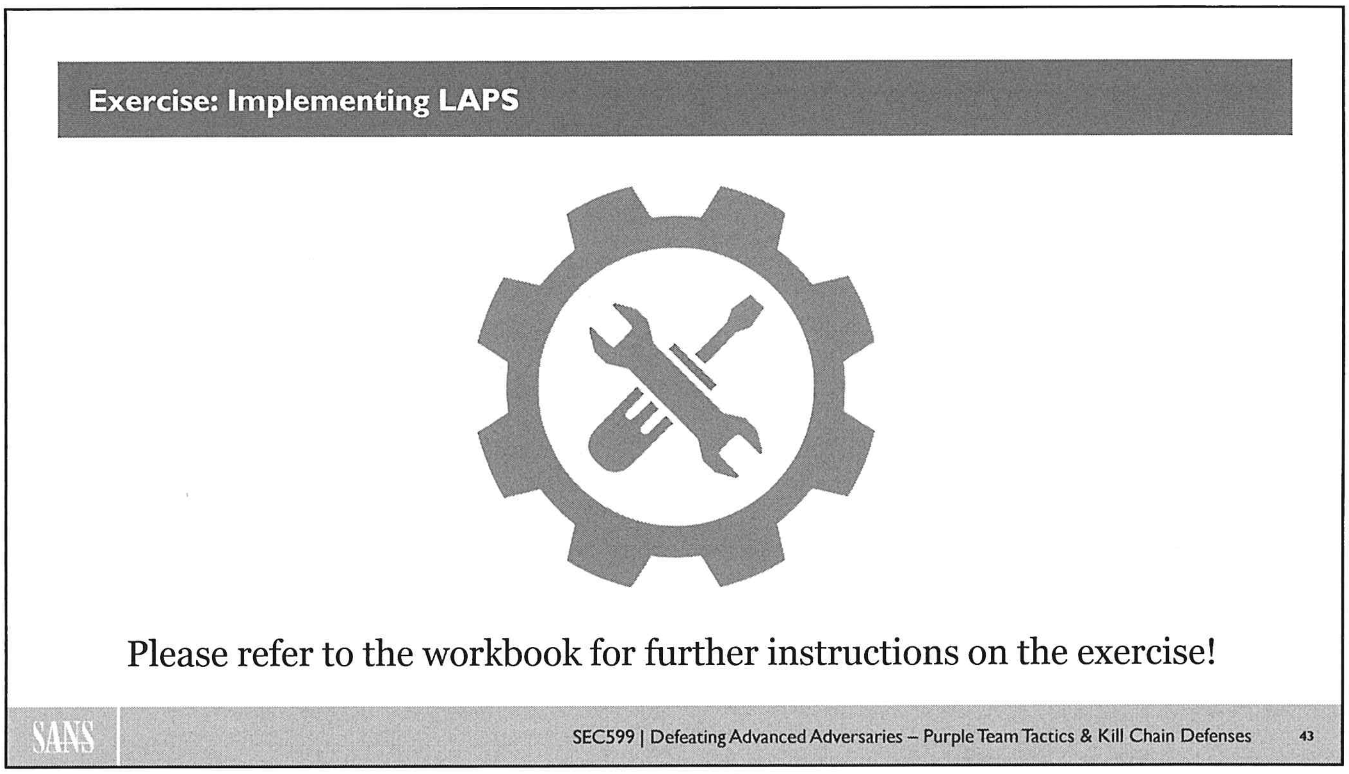
- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- **<u>Day 4: Lateral Movement</u>**
- Day 5: Action on Objectives, Threat Hunting & Incident Response
- Day 6: APT Defender Capstone

SEC599.4

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 42

This page intentionally left blank.

© 2019 Erik Van Buggenhout & Stephen Sims

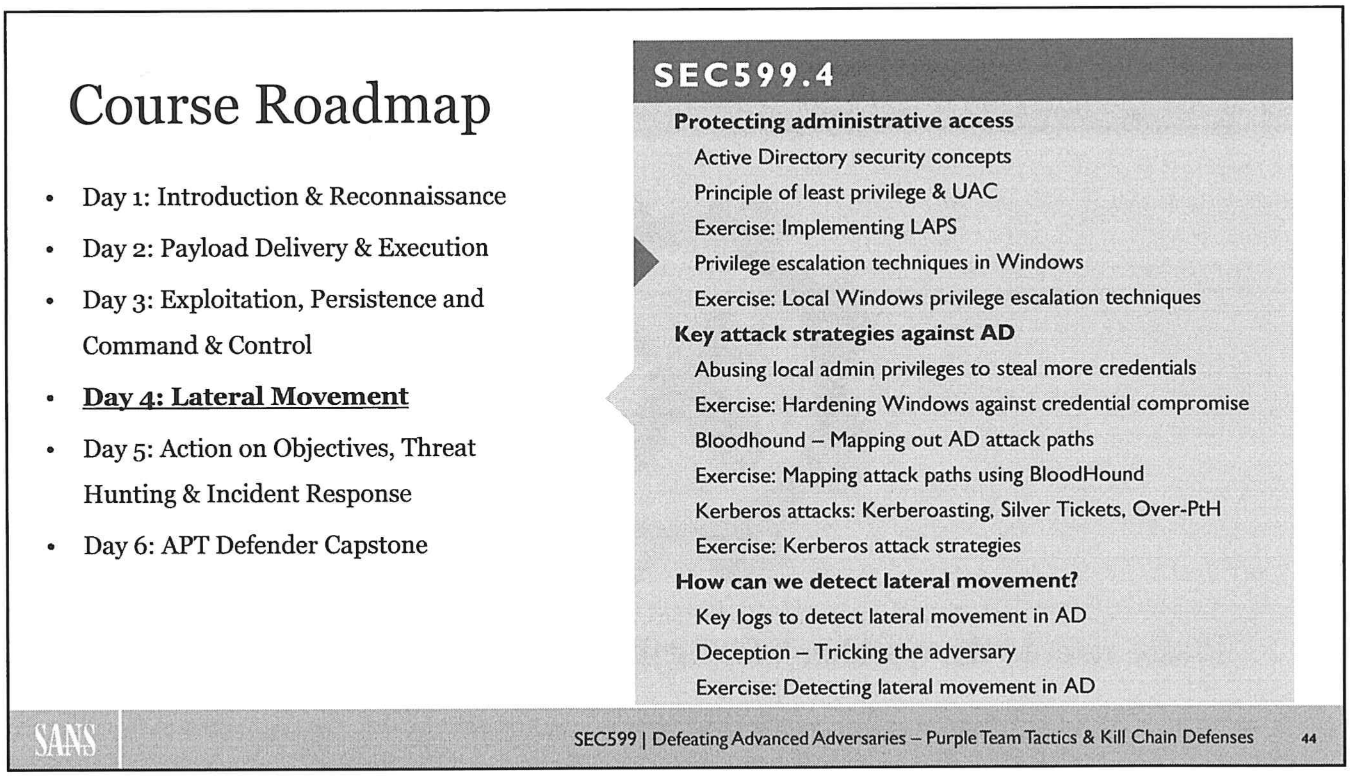


Exercise: Implementing LAPS

Please refer to the workbook for further instructions on the exercise!

This page intentionally left blank.

© 2019 Erik Van Buggenhout & Stephen Sims



Course Roadmap

- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- **Day 4: Lateral Movement**
- Day 5: Action on Objectives, Threat Hunting & Incident Response
- Day 6: APT Defender Capstone

SEC599.4

Protecting administrative access
- Active Directory security concepts
- Principle of least privilege & UAC
- Exercise: Implementing LAPS
- Privilege escalation techniques in Windows
- Exercise: Local Windows privilege escalation techniques

Key attack strategies against AD
- Abusing local admin privileges to steal more credentials
- Exercise: Hardening Windows against credential compromise
- Bloodhound – Mapping out AD attack paths
- Exercise: Mapping attack paths using BloodHound
- Kerberos attacks: Kerberoasting, Silver Tickets, Over-PtH
- Exercise: Kerberos attack strategies

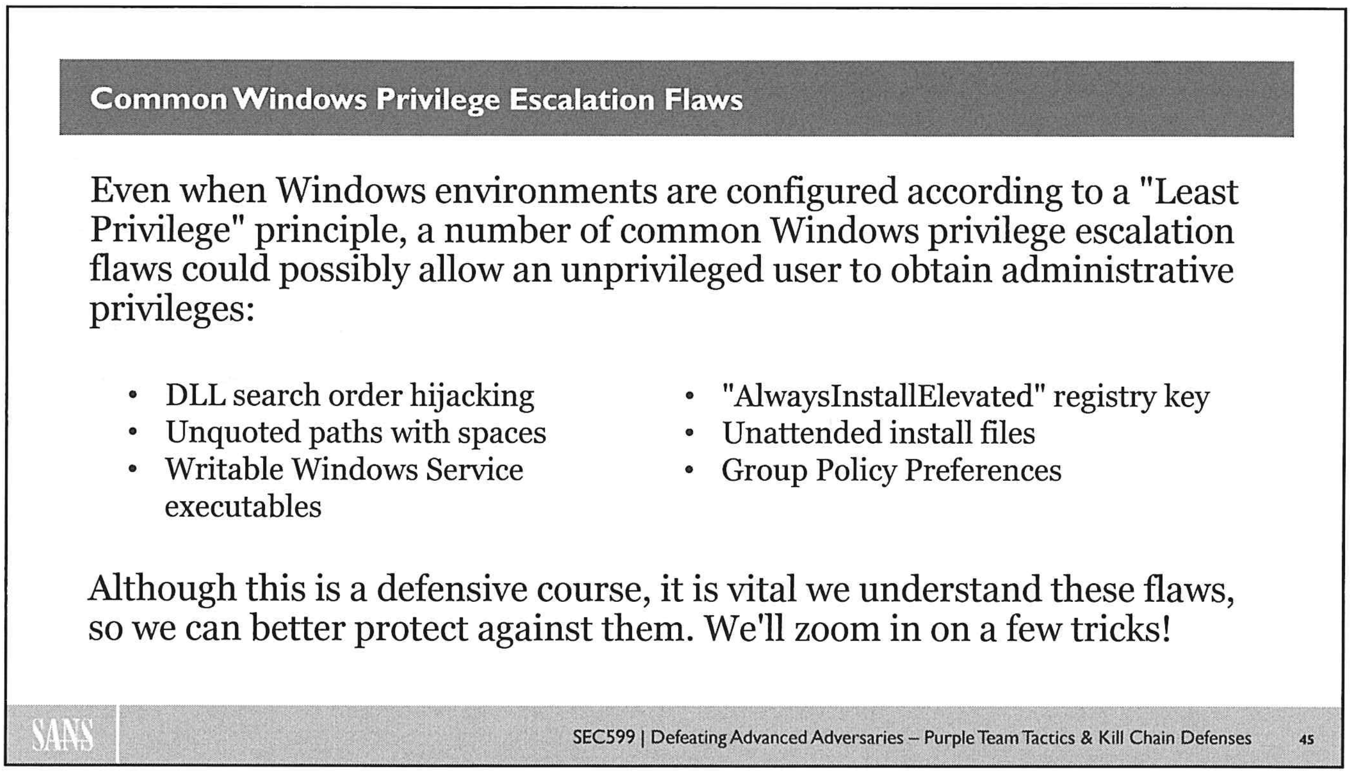
How can we detect lateral movement?
- Key logs to detect lateral movement in AD
- Deception – Tricking the adversary
- Exercise: Detecting lateral movement in AD

SANS

SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 44

This page intentionally left blank.

© 2019 Erik Van Buggenhout & Stephen Sims



Common Windows Privilege Escalation Flaws

Even when Windows environments are configured according to a "Least Privilege" principle, a number of common Windows privilege escalation flaws could possibly allow an unprivileged user to obtain administrative privileges:

- DLL search order hijacking
- Unquoted paths with spaces
- Writable Windows Service executables
- "AlwaysInstallElevated" registry key
- Unattended install files
- Group Policy Preferences

Although this is a defensive course, it is vital we understand these flaws, so we can better protect against them. We'll zoom in on a few tricks!

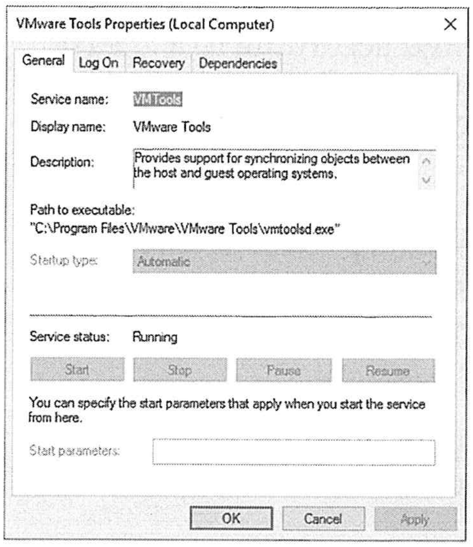
SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 45

Common Windows Privilege Escalation Flaws
Even when Windows environments are configured according to a "Least Privilege" principle and the number of administrative accounts is restricted; a number of common Windows privilege escalation flaws could possibly allow an unprivileged user to obtain administrative privileges. Local privilege escalation issues could be caused by outdated / unpatched software that exposes the core OS to vulnerabilities. More often, however, they are the result of a number of misconfigurations or mistakes that adversaries can abuse. The below is a non-exhaustive list of some commonly used attack strategies:

- DLL search order hijacking (which we discussed already in the persistence section of this material).
- Unquoted paths with spaces.
- Writable Windows Service executables.
- "AlwaysInstallElevated" registry key.
- Unattended install files.
- Group Policy Preferences (especially relevant on Windows 2008 domain environments).

Although this is a defensive course, it is vital we understand these flaws, so we can better protect against them. We will zoom in on a few of these tricks!

Unquoted Paths with Spaces (1)


VMware Tools Properties (Local Computer)

General | Log On | Recovery | Dependencies

Service name: VMTools
Display name: VMware Tools
Description: Provides support for synchronizing objects between the host and guest operating systems.
Path to executable:
"C:\Program Files\VMware\VMware Tools\vmtoolsd.exe"
Startup type: Automatic

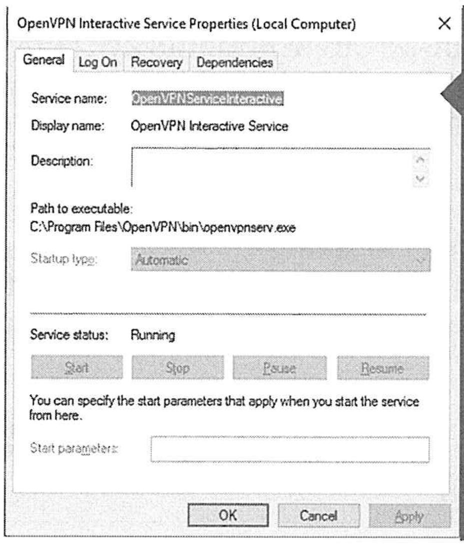
Service status: Running

Start | Stop | Pause | Resume

You can specify the start parameters that apply when you start the service from here.

Start parameters:

OK | Cancel | Apply


OpenVPN Interactive Service Properties (Local Computer)

General | Log On | Recovery | Dependencies

Service name: OpenVPNServiceInteractive
Display name: OpenVPN Interactive Service
Description:
Path to executable:
C:\Program Files\OpenVPN\bin\openvpnserv.exe
Startup type: Automatic

Service status: Running

Start | Stop | Pause | Resume

You can specify the start parameters that apply when you start the service from here.

Start parameters:

OK | Cancel | Apply

Both pictures are screenshots of two installed services in a Windows 10 environment.

Note the difference in the "Path to executable", which is surrounded using quotation marks in the example on the left.

What could go wrong in the example on the right?

Unquoted Paths with Spaces (1)
Unquoted paths with spaces can be a serious issue in Windows Services configurations. Let's analyze an example:

- The picture on the left is a screenshot of a recent VMWare Tools service installed on a Windows 10 host. Note the **Path to executable**, which is set to:

  "C:\Program Files\Vmware\Vmware Tools\vmtoolsd.exe"

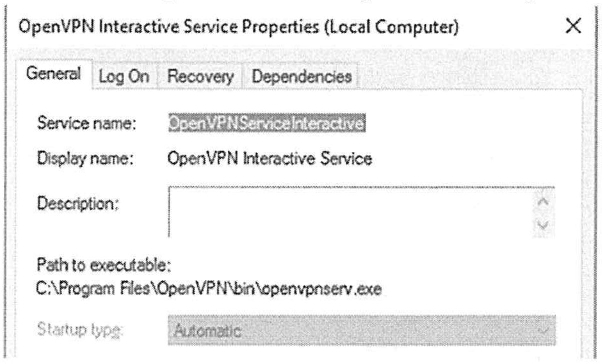
- The picture on the right is a screenshot of a recent OpenVPN installed on a Windows 10 host. Note the **Path to executable**, which is set to:

  C:\Program Files\OpenVPN\bin\openvpnserv.exe

The Vmware Tools service is clearly **NOT VULNERABLE** to the issue, as the executable path is properly delimited using quotes. The OpenVPN service, however, raises some questions… Can you think of what could go wrong?

Unquoted Paths with Spaces (2)

If the service path contains spaces and is not surrounded by quotation marks, then Windows has to guess where to find the service executable... Spaces on the CMD could either be part of the file path OR they could indicate command-line arguments!


OpenVPN Interactive Service Properties (Local Computer)

Service name: OpenVPNServiceInteractive
Display name: OpenVPN Interactive Service
Description:
Path to executable:
C:\Program Files\OpenVPN\bin\openvpnserv.exe
Startup type: Automatic

OPTION 1
File Path: C:\Program
Arguments: Files\OpenVPN\bin\openvpnserv.exe

OPTION 2
File Path: C:\Program Files\OpenVPN\bin\openvpnserv.exe
Arguments: <NONE>

If they can write to "C:\" adversaries could trick the Windows service controller to run C:\Program.exe (with elevated privileges) instead of openvpnserv.exe

```
wmic service get name,displayname,pathname,startmode |findstr /i "Auto" | findstr /i /v "C:\Windows\\" | findstr /i /v """
```

Unquoted Paths with Spaces (2)
If the service path contains spaces and is not surrounded by quotation marks, then Windows has to guess where to find the service executable... Spaces on the CMD could either be part of the file path OR they could indicate command-line arguments!

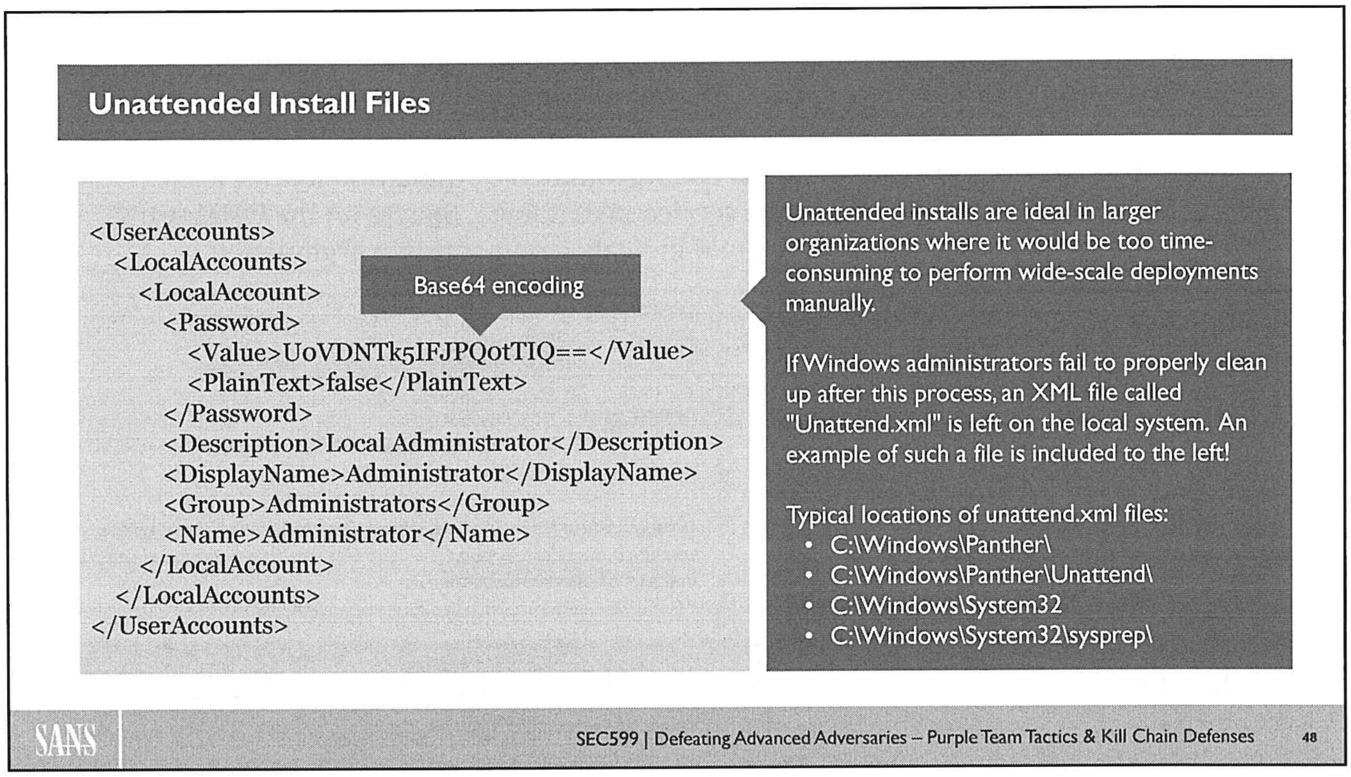
So, let's have a look at that OpenVPN service... Windows tries to start this service "automatically" (so on boot). But how does Windows try to start the service? It, of course, tries to load the "executable path". But, how is it interpreted? Given the space in the path name, there's an ambiguity here:

OPTION 1
Logic: Windows considers the space at the end of the filename and assumes everything that follows are arguments passed to that executable.
File Path: C:\Program
Arguments: Files\OpenVPN\bin\openvpnserv.exe

OPTION 2
Logic: The expected logic, Windows considers the space as part of the file path and finds the correct executable.
File Path: C:\Program Files\OpenVPN\bin\openvpnserv.exe
Arguments: <NONE>

Windows will first attempt option 1. If the adversary could thus write to "C:\", he/she could create a program called "Program.exe" that would subsequently be executed with the privileges of the OpenVPN Windows service. In this specific case, there is no privilege escalation as an unprivileged user cannot write to "C:\" either.

An excellent command line to find these types of services on our own machine is below (originally posted by Danial Compton):
wmic service get name,displayname,pathname,startmode |findstr /i "Auto" | findstr /i /v "C:\Windows\\" | findstr /i /v """



Unattended Install Files

```
<UserAccounts>
  <LocalAccounts>
    <LocalAccount>
      <Password>
        <Value>UoVDNTk5IFJPQotTIQ==</Value>
        <PlainText>false</PlainText>
      </Password>
      <Description>Local Administrator</Description>
      <DisplayName>Administrator</DisplayName>
      <Group>Administrators</Group>
      <Name>Administrator</Name>
    </LocalAccount>
  </LocalAccounts>
</UserAccounts>
```

Base64 encoding

Unattended installs are ideal in larger organizations where it would be too time-consuming to perform wide-scale deployments manually.

If Windows administrators fail to properly clean up after this process, an XML file called "Unattend.xml" is left on the local system. An example of such a file is included to the left!

Typical locations of unattend.xml files:

- C:\Windows\Panther\
- C:\Windows\Panther\Unattend\
- C:\Windows\System32
- C:\Windows\System32\sysprep\

Unattended Install Files

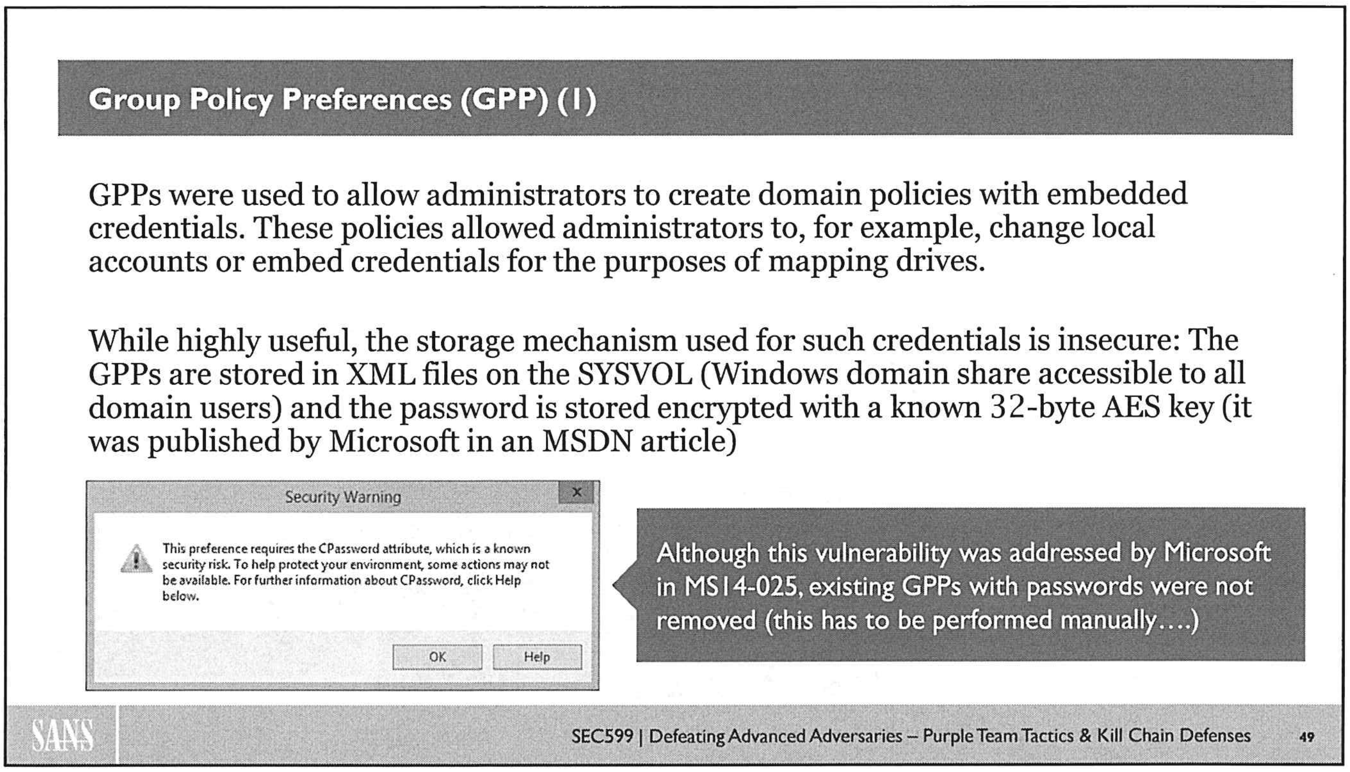
Unattended installs are often used in enterprise organizations where it would be too time-consuming to perform wide-scale deployments manually. If Windows administrators fail to properly clean up after this process, an XML file called "Unattend.xml" is left on the local system. An example of such a file is included to the left!

As you can see, it includes the password in a base64 encoded format, which means it can be very easily decoded. Now, where can you find these xml files? It depends… Some good candidates to check include:

- C:\Windows\Panther\
- C:\Windows\Panther\Unattend\
- C:\Windows\System32
- C:\Windows\System32\sysprep\

Keep an eye out on your environment and search for these files periodically… Upon discovery, they should be immediately deleted!

© 2019 Erik Van Buggenhout & Stephen Sims



Group Policy Preferences (GPP) (1)

GPPs were used to allow administrators to create domain policies with embedded credentials. These policies allowed administrators to, for example, change local accounts or embed credentials for the purposes of mapping drives.

While highly useful, the storage mechanism used for such credentials is insecure: The GPPs are stored in XML files on the SYSVOL (Windows domain share accessible to all domain users) and the password is stored encrypted with a known 32-byte AES key (it was published by Microsoft in an MSDN article)

Security Warning

This preference requires the CPassword attribute, which is a known security risk. To help protect your environment, some actions may not be available. For further information about CPassword, click Help below.

OK Help

Although this vulnerability was addressed by Microsoft in MS14-025, existing GPPs with passwords were not removed (this has to be performed manually....)

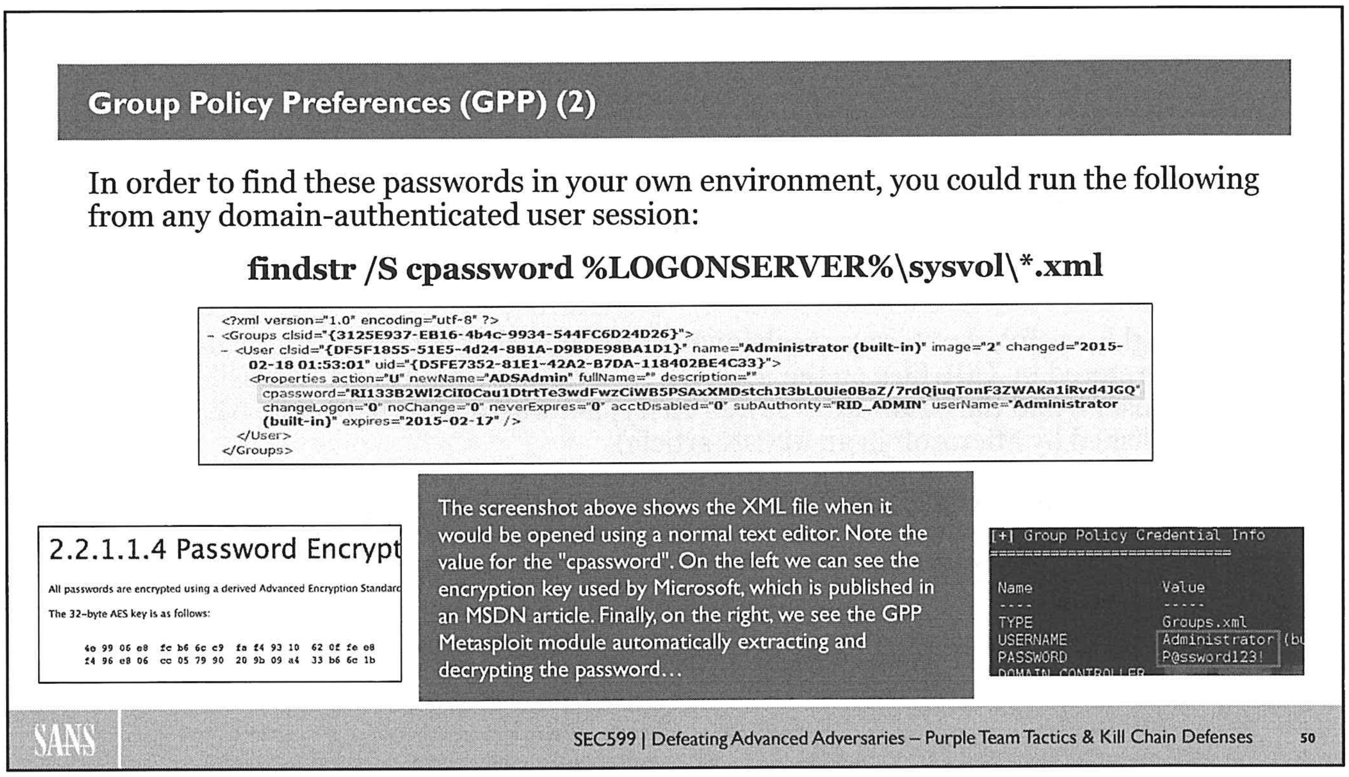
SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 49

Group Policy Preferences (GPP) (1)

GPPs were used to allow administrators to create domain policies with embedded credentials. These policies allowed administrators to, for example, change local accounts or embed credentials for the purposes of mapping drives.

While highly useful, the storage mechanism used for such credentials is insecure: The GPPs are stored in XML files on the SYSVOL share (Windows domain share accessible to all domain users) and the password is stored encrypted with a known 32-byte AES key (it was published by Microsoft in an MSDN article - https://msdn.microsoft.com/en-us/library/cc422924.aspx).

Although this vulnerability was addressed by Microsoft in MS14-025, existing GPPs with passwords were not removed (this has to be performed manually...)



Group Policy Preferences (GPP) (2)

In order to find these passwords in your own environment, you could run the following from any domain-authenticated user session:

findstr /S cpassword %LOGONSERVER%\sysvol\*.xml

```
<?xml version="1.0" encoding="utf-8" ?>
- <Groups clsid="{3125E937-EB16-4b4c-9934-544FC6D24D26}">
 - <User clsid="{DF5F1855-51E5-4d24-8B1A-D9BDE98BA1D1}" name="Administrator (built-in)" image="2" changed="2015-
     02-18 01:53:01" uid="{D5FE7352-81E1-42A2-B7DA-118402BE4C33}">
     <Properties action="U" newName="ADSAdmin" fullName="" description=""
       cpassword="RI133B2Wl2CiI0Cau1DtrtTe3wdFwzCiWB5PSAxXMDstchJt3bL0Uie0BaZ/7rdQjuqTonF3ZWAKa1iRvd4JGQ"
       changeLogon="0" noChange="0" neverExpires="0" acctDisabled="0" subAuthority="RID_ADMIN" userName="Administrator
       (built-in)" expires="2015-02-17" />
   </User>
 </Groups>
```

2.2.1.1.4 Password Encrypt

All passwords are encrypted using a derived Advanced Encryption Standard

The 32-byte AES key is as follows:

```
4e 99 06 e8  fc b6 6c c9  fa f4 93 10  62 0f fe e8
f4 96 e8 06  cc 05 79 90  20 9b 09 a4  33 b6 6c 1b
```

The screenshot above shows the XML file when it would be opened using a normal text editor. Note the value for the "cpassword". On the left we can see the encryption key used by Microsoft, which is published in an MSDN article. Finally, on the right, we see the GPP Metasploit module automatically extracting and decrypting the password...

```
[+] Group Policy Credential Info
================================

 Name                   Value
 ----                   -----
 TYPE                   Groups.xml
 USERNAME               Administrator (bu
 PASSWORD               P@ssword123!
 DOMAIN CONTROLLER
```

SANS

SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 50

Group Policy Preferences (GPP) (2)
In order to find these passwords in your own environment, you could run the following from any domain-authenticated user session:

findstr /S cpassword %LOGONSERVER%\sysvol\*.xml

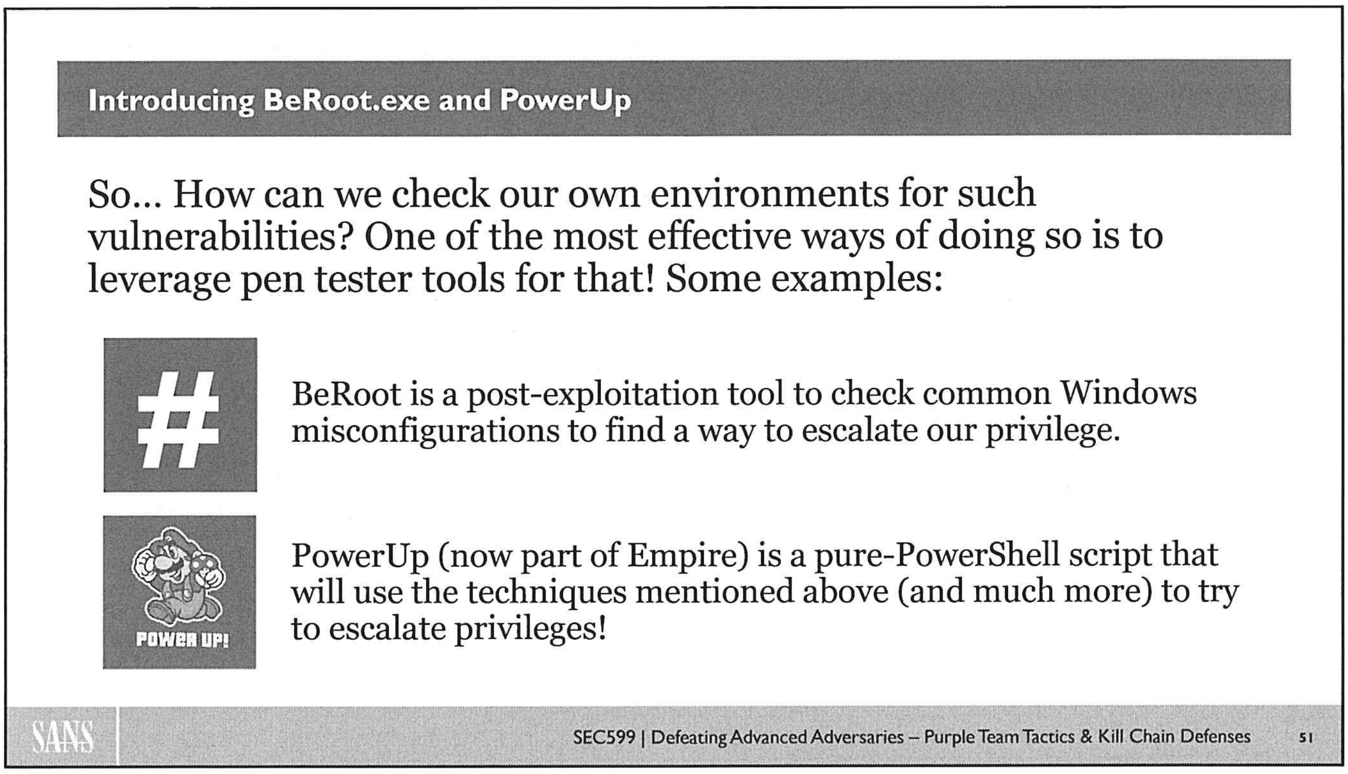
This one-line command will search for the string "cpassword" in any .xml file under the domain controller's publicly accessible sysvol network share.

The screenshots on the slide provide some additional context:

- The first screenshot shows an identified XML file with a cpassword value when it would be opened using a normal text editor. Note the value for the "cpassword".
- On the bottom left we can see the encryption key used by Microsoft, which is published in an MSDN article.
- Finally, on the bottom right, we see the GPP Metasploit module automatically extracting and decrypting the password...

Next to Metasploit modules, several PowerShell scripts exist that will do the same thing, which of course reduces the detection rate of such tools.

© 2019 Erik Van Buggenhout & Stephen Sims



Introducing BeRoot.exe and PowerUp

So… How can we check our own environments for such vulnerabilities? One of the most effective ways of doing so is to leverage pen tester tools for that! Some examples:

BeRoot is a post-exploitation tool to check common Windows misconfigurations to find a way to escalate our privilege.

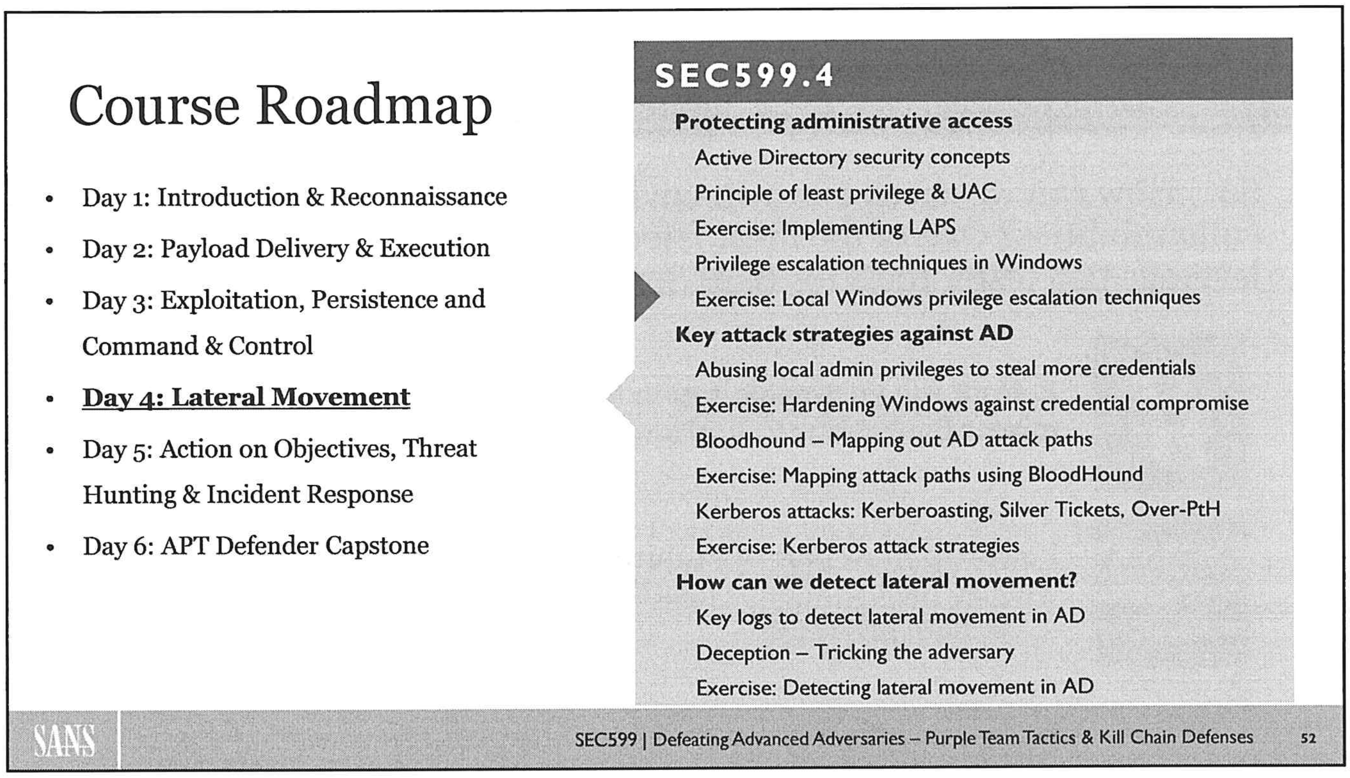
PowerUp (now part of Empire) is a pure-PowerShell script that will use the techniques mentioned above (and much more) to try to escalate privileges!

SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 51

Introducing BeRoot.exe and PowerUp
In order to better protect ourselves against these typical privilege escalation attacks, it's a good idea to use tooling that allows us to easily assess how vulnerable we are, so we can fix any identified vulnerabilities. Two tools are very easy to run and use:

- BeRoot is a post-exploitation tool to check common Windows misconfigurations. It can be downloaded as a stand-alone binary. Upon running, it will attempt any privilege escalation techniques and will return a result.

- PowerUp is a pure-PowerShell script that will use the techniques mentioned above (and much more) to try to escalate privileges. PowerUp used to be a stand-alone PowerShell script, but it's now been included in the Empire PowerShell post-exploitation framework.



Course Roadmap

- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- **Day 4: Lateral Movement**
- Day 5: Action on Objectives, Threat Hunting & Incident Response
- Day 6: APT Defender Capstone

SEC599.4

Protecting administrative access

Active Directory security concepts

Principle of least privilege & UAC

Exercise: Implementing LAPS

Privilege escalation techniques in Windows

Exercise: Local Windows privilege escalation techniques

Key attack strategies against AD

Abusing local admin privileges to steal more credentials

Exercise: Hardening Windows against credential compromise

Bloodhound – Mapping out AD attack paths

Exercise: Mapping attack paths using BloodHound

Kerberos attacks: Kerberoasting, Silver Tickets, Over-PtH

Exercise: Kerberos attack strategies

How can we detect lateral movement?

Key logs to detect lateral movement in AD

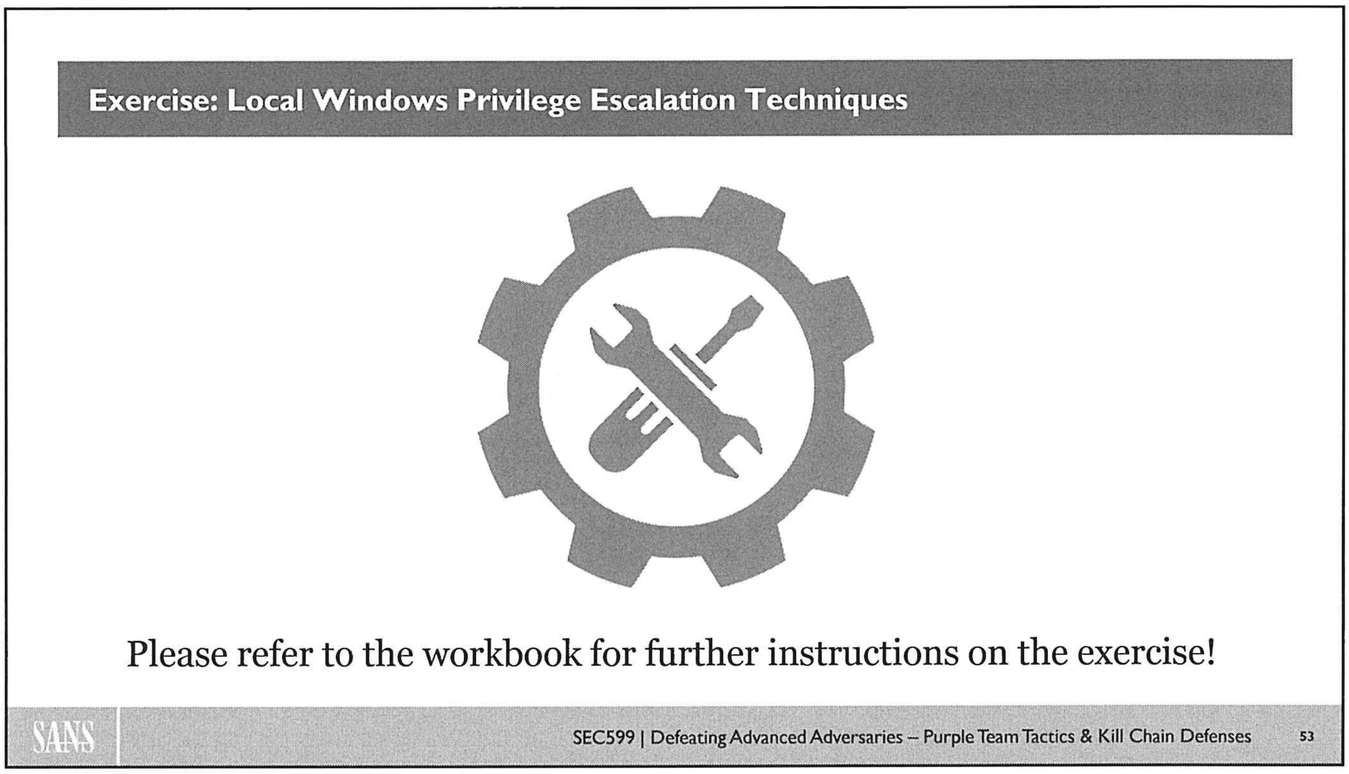
Deception – Tricking the adversary

Exercise: Detecting lateral movement in AD

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 52

This page intentionally left blank.

© 2019 Erik Van Buggenhout & Stephen Sims

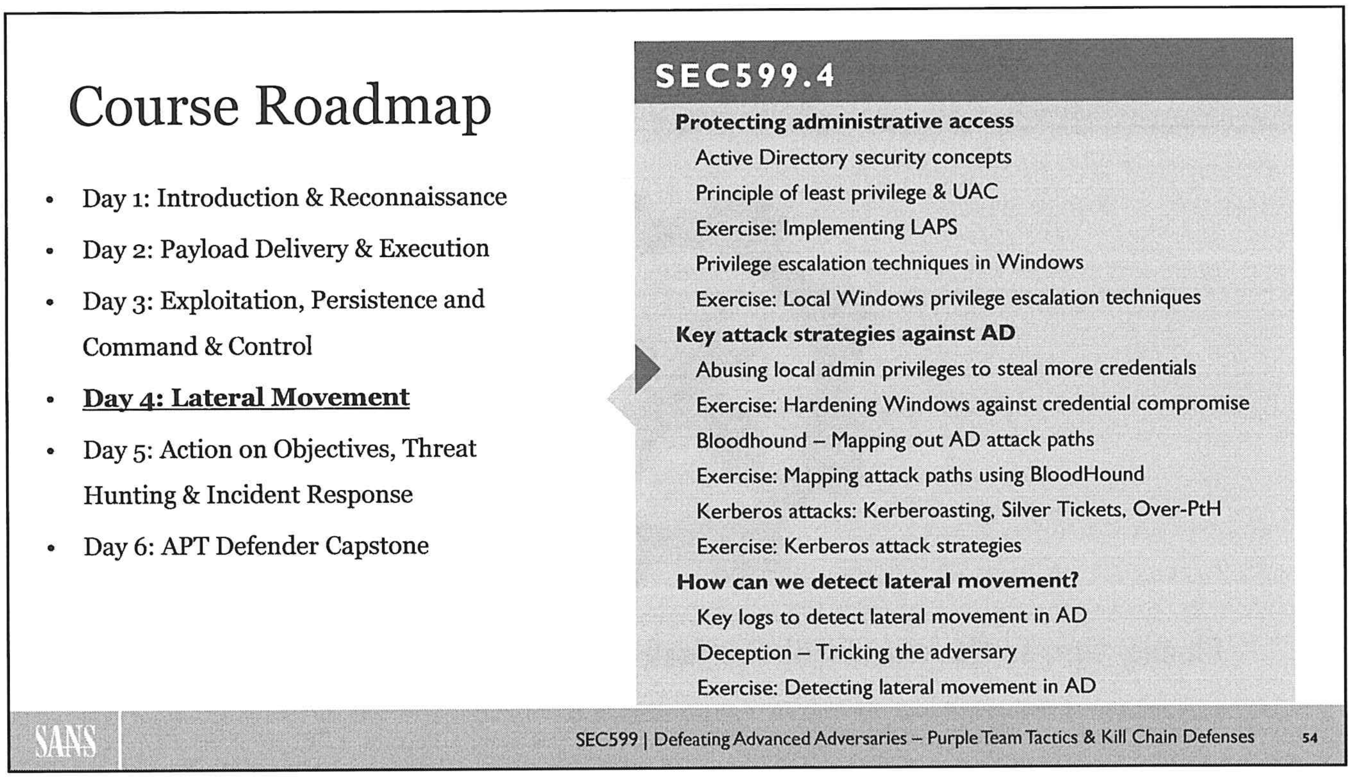


Exercise: Local Windows Privilege Escalation Techniques

Please refer to the workbook for further instructions on the exercise!

SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 53

This page intentionally left blank.



Course Roadmap

- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- **Day 4: Lateral Movement**
- Day 5: Action on Objectives, Threat Hunting & Incident Response
- Day 6: APT Defender Capstone

SEC599.4

Protecting administrative access

Active Directory security concepts

Principle of least privilege & UAC

Exercise: Implementing LAPS

Privilege escalation techniques in Windows

Exercise: Local Windows privilege escalation techniques

Key attack strategies against AD

Abusing local admin privileges to steal more credentials

Exercise: Hardening Windows against credential compromise

Bloodhound – Mapping out AD attack paths

Exercise: Mapping attack paths using BloodHound

Kerberos attacks: Kerberoasting, Silver Tickets, Over-PtH

Exercise: Kerberos attack strategies

How can we detect lateral movement?

Key logs to detect lateral movement in AD

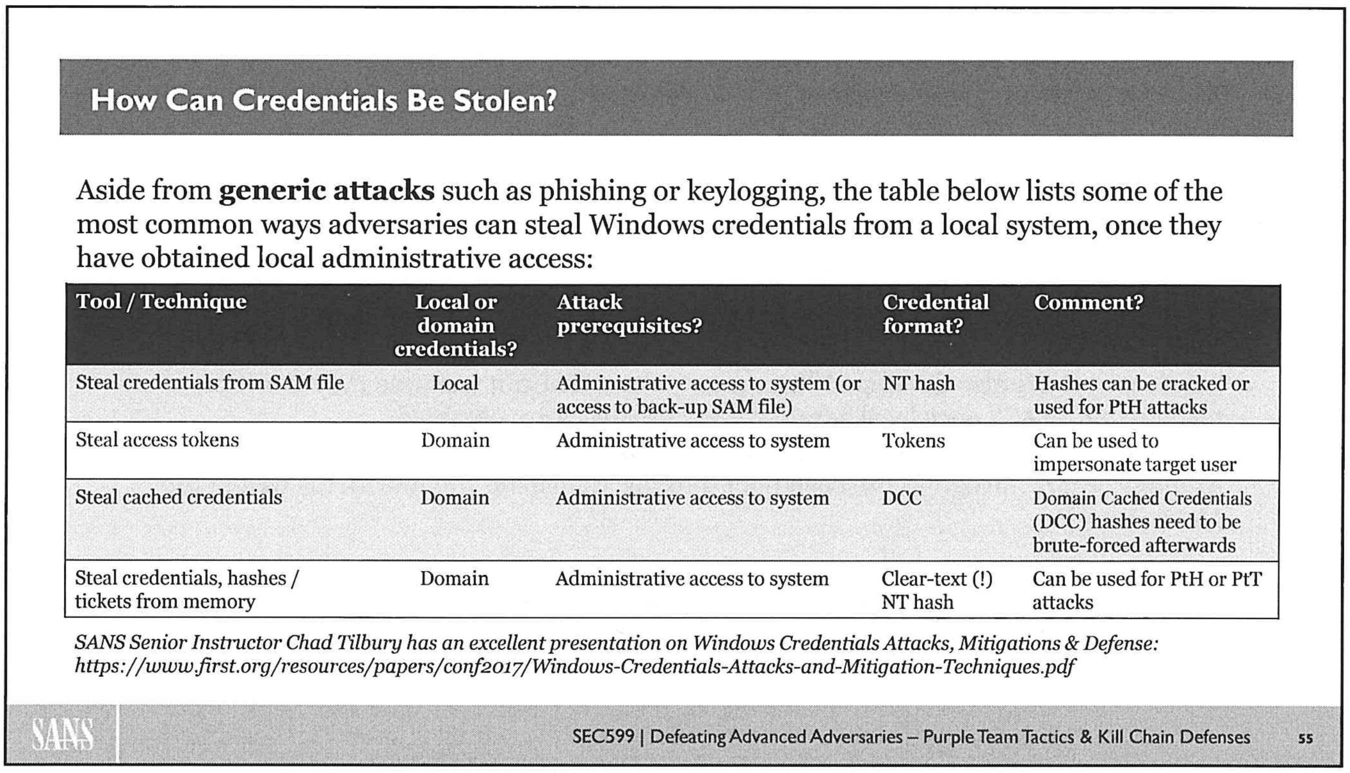
Deception – Tricking the adversary

Exercise: Detecting lateral movement in AD

SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 54

This page intentionally left blank.

© 2019 Erik Van Buggenhout & Stephen Sims



How Can Credentials Be Stolen?

Aside from **generic attacks** such as phishing or keylogging, the table below lists some of the most common ways adversaries can steal Windows credentials from a local system, once they have obtained local administrative access:

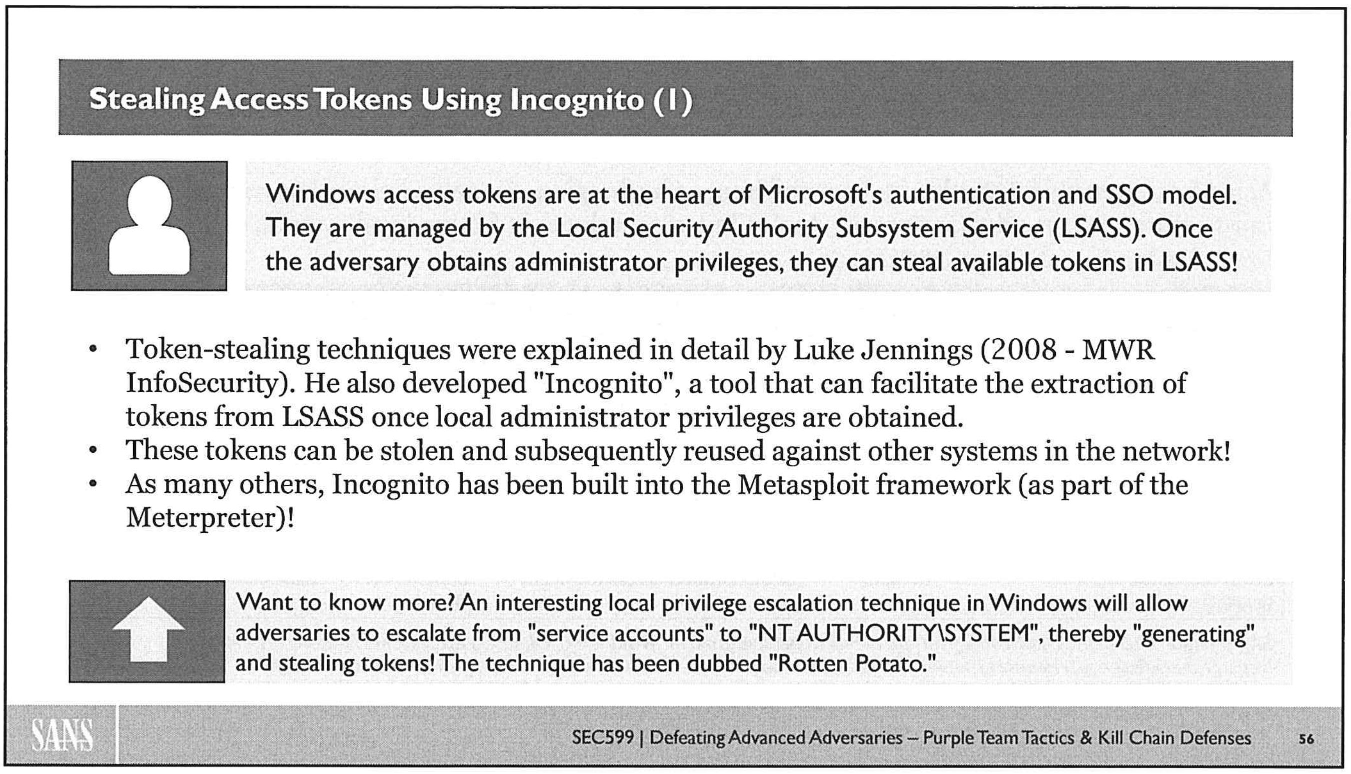
Tool / Technique	Local or domain credentials?	Attack prerequisites?	Credential format?	Comment?
Steal credentials from SAM file	Local	Administrative access to system (or access to back-up SAM file)	NT hash	Hashes can be cracked or used for PtH attacks
Steal access tokens	Domain	Administrative access to system	Tokens	Can be used to impersonate target user
Steal cached credentials	Domain	Administrative access to system	DCC	Domain Cached Credentials (DCC) hashes need to be brute-forced afterwards
Steal credentials, hashes / tickets from memory	Domain	Administrative access to system	Clear-text (!) NT hash	Can be used for PtH or PtT attacks

SANS Senior Instructor Chad Tilbury has an excellent presentation on Windows Credentials Attacks, Mitigations & Defense: https://www.first.org/resources/papers/conf2017/Windows-Credentials-Attacks-and-Mitigation-Techniques.pdf

How Can Credentials Be Stolen?
Aside from generic attacks such as phishing or keylogging, the table below lists some of the most common ways adversaries can steal Windows credentials from a local system. Note that all of these require prior local Administrative access to the system:

- Adversaries can steal cached credentials in Domain Cached Credentials (DCC) format, which are used by Windows to support authentication to the domain when connectivity to the domain controller is not available.
- Adversaries can steal hashes / tickets that are stored in the LSASS memory. Depending on the Windows version and configuration, this could include cleartext credentials or just the NT hash (which can be used in a Pass-the-Hash attack).
- Attackers could steal access tokens, which can be used to impersonate users with higher privileges.
- Finally, adversaries can steal credentials from the local SAM file, which would of course provide local credentials, which is less relevant in an attack where adversaries have already obtained administrative access to the local system.

Let's zoom in on these attack techniques...



Stealing Access Tokens Using Incognito (1)

Windows access tokens are at the heart of Microsoft's authentication and SSO model. They are managed by the Local Security Authority Subsystem Service (LSASS). Once the adversary obtains administrator privileges, they can steal available tokens in LSASS!

- Token-stealing techniques were explained in detail by Luke Jennings (2008 - MWR InfoSecurity). He also developed "Incognito", a tool that can facilitate the extraction of tokens from LSASS once local administrator privileges are obtained.
- These tokens can be stolen and subsequently reused against other systems in the network!
- As many others, Incognito has been built into the Metasploit framework (as part of the Meterpreter)!

Want to know more? An interesting local privilege escalation technique in Windows will allow adversaries to escalate from "service accounts" to "NT AUTHORITY\SYSTEM", thereby "generating" and stealing tokens! The technique has been dubbed "Rotten Potato."

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 56

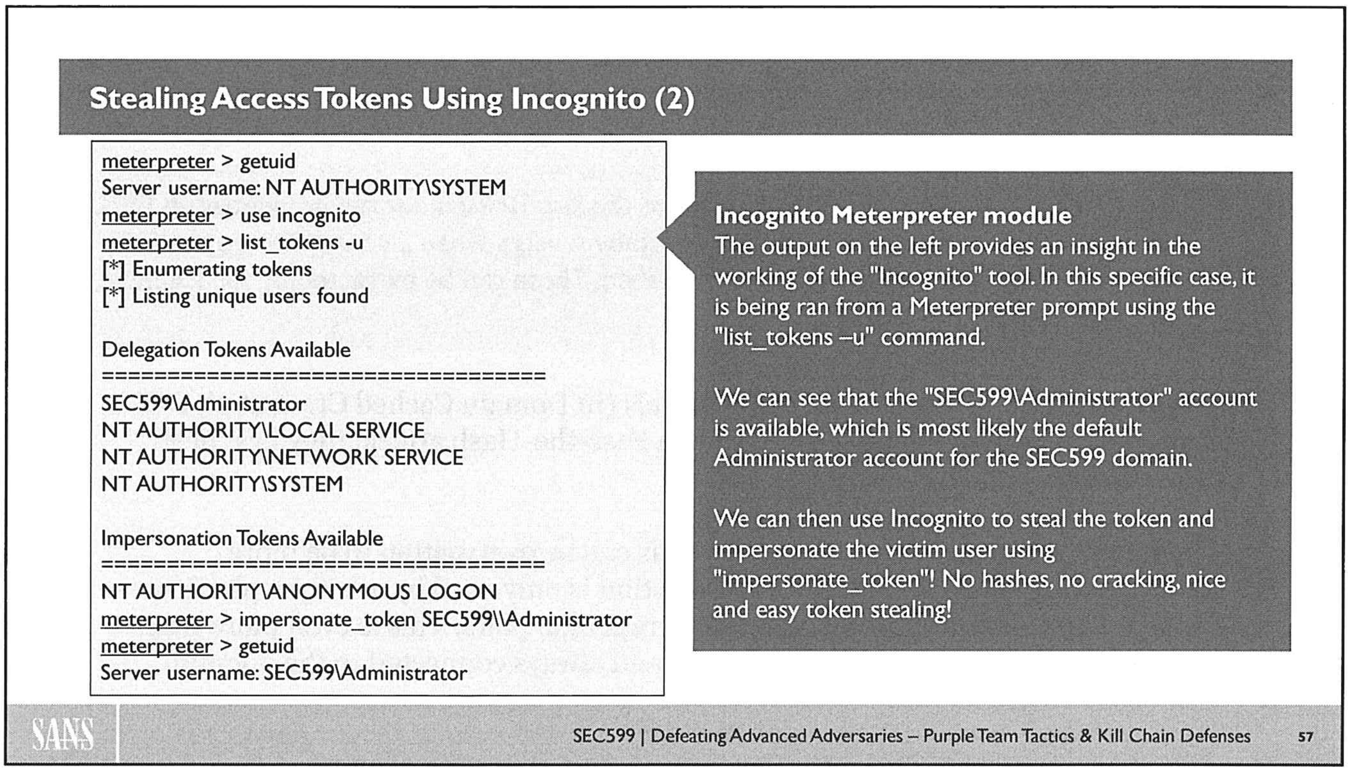
Stealing Access Tokens Using Incognito (1)
Windows access tokens are at the heart of Microsoft's authentication and SSO model. Please refer to the previous sections of this course for additional information on how Windows Access Tokens are used by Microsoft. In short, tokens are managed by the Local Security Authority Subsystem Service (LSASS). Once the adversary obtains administrator privileges, they can steal available tokens in LSASS!

- Luke Jennings (MWR InfoSecurity) published an in-depth analysis of token-stealing techniques in 2008. He also developed "Incognito", a tool that can facilitate the extraction of tokens from LSASS once local administrator privileges are obtained.
- These tokens can be stolen and subsequently reused against other systems in the network!
- As many others, Incognito has been built in in the Metasploit framework (as part of the Meterpreter)!

Want to know more? An interesting local privilege escalation technique in Windows will allow adversaries to escalate from "service accounts" to "NT AUTHORITY\SYSTEM", thereby "generating" and stealing tokens! The technique has been dubbed "Rotten Potato." For additional information on the "Rotten Potato" attack, please refer to the following article:

https://foxglovesecurity.com/2016/09/26/rotten-potato-privilege-escalation-from-service-accounts-to-system/

© 2019 Erik Van Buggenhout & Stephen Sims


Stealing Access Tokens Using Incognito (2)

```
meterpreter > getuid
Server username: NT AUTHORITY\SYSTEM
meterpreter > use incognito
meterpreter > list_tokens -u
[*] Enumerating tokens
[*] Listing unique users found

Delegation Tokens Available
=================================
SEC599\Administrator
NT AUTHORITY\LOCAL SERVICE
NT AUTHORITY\NETWORK SERVICE
NT AUTHORITY\SYSTEM

Impersonation Tokens Available
=================================
NT AUTHORITY\ANONYMOUS LOGON
meterpreter > impersonate_token SEC599\\Administrator
meterpreter > getuid
Server username: SEC599\Administrator
```

Incognito Meterpreter module
The output on the left provides an insight in the working of the "Incognito" tool. In this specific case, it is being ran from a Meterpreter prompt using the "list_tokens –u" command.

We can see that the "SEC599\Administrator" account is available, which is most likely the default Administrator account for the SEC599 domain.

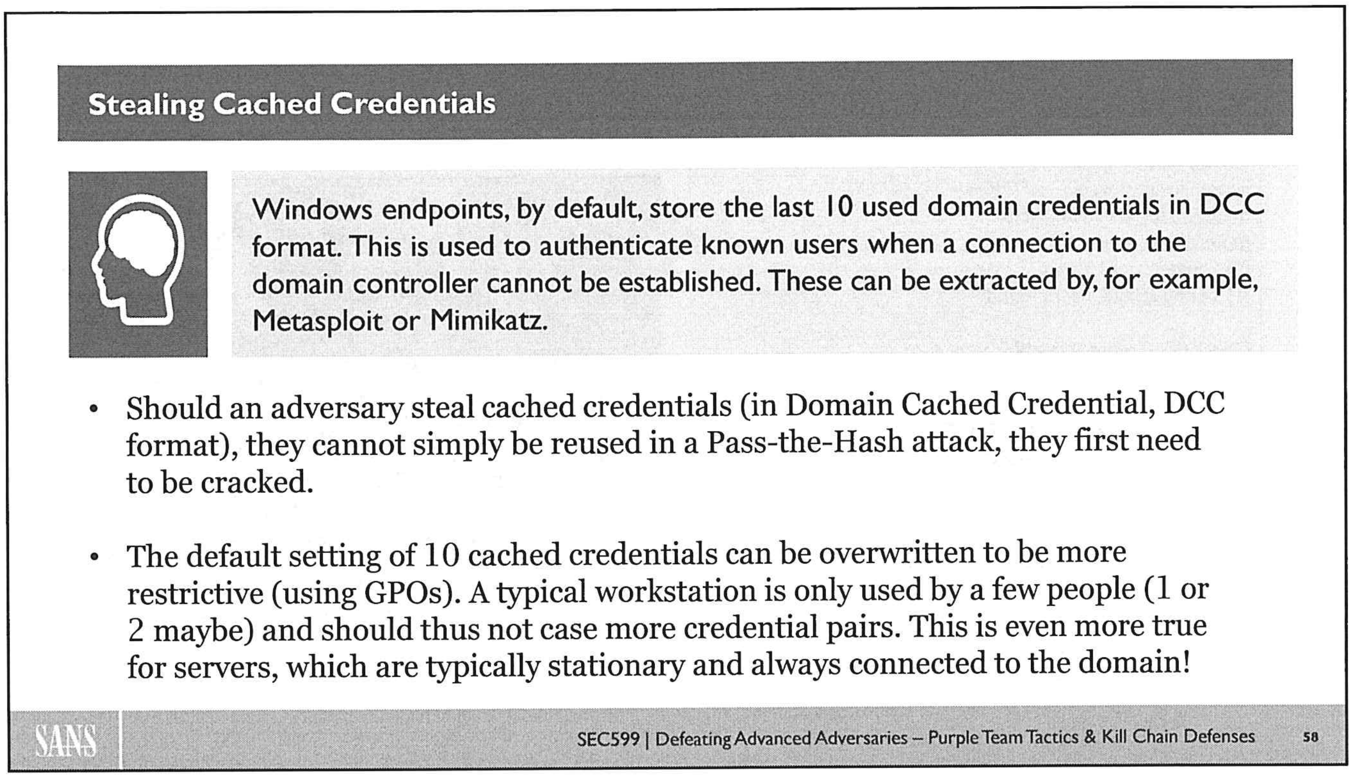
We can then use Incognito to steal the token and impersonate the victim user using "impersonate_token"! No hashes, no cracking, nice and easy token stealing!

SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 57

Stealing Access Tokens Using Incognito (2)
The output on the slide provides an insight into the working of the "Incognito" tool. What we can see is the following:

- The adversary checks his current privileges on the system using the "getuid" command.
- The adversary loads the incognito module in the meterpreter.
- The available access tokens are listed using "list_tokens -u".
- The "SEC599\Administrator" token is found and impersonated using "impersonate_token".
- The adversary confirms his stolen privileges using the "getuid" command.

No hashes, no cracking, nice and easy token stealing!



Stealing Cached Credentials

Windows endpoints, by default, store the last 10 used domain credentials in DCC format. This is used to authenticate known users when a connection to the domain controller cannot be established. These can be extracted by, for example, Metasploit or Mimikatz.

- Should an adversary steal cached credentials (in Domain Cached Credential, DCC format), they cannot simply be reused in a Pass-the-Hash attack, they first need to be cracked.
- The default setting of 10 cached credentials can be overwritten to be more restrictive (using GPOs). A typical workstation is only used by a few people (1 or 2 maybe) and should thus not case more credential pairs. This is even more true for servers, which are typically stationary and always connected to the domain!

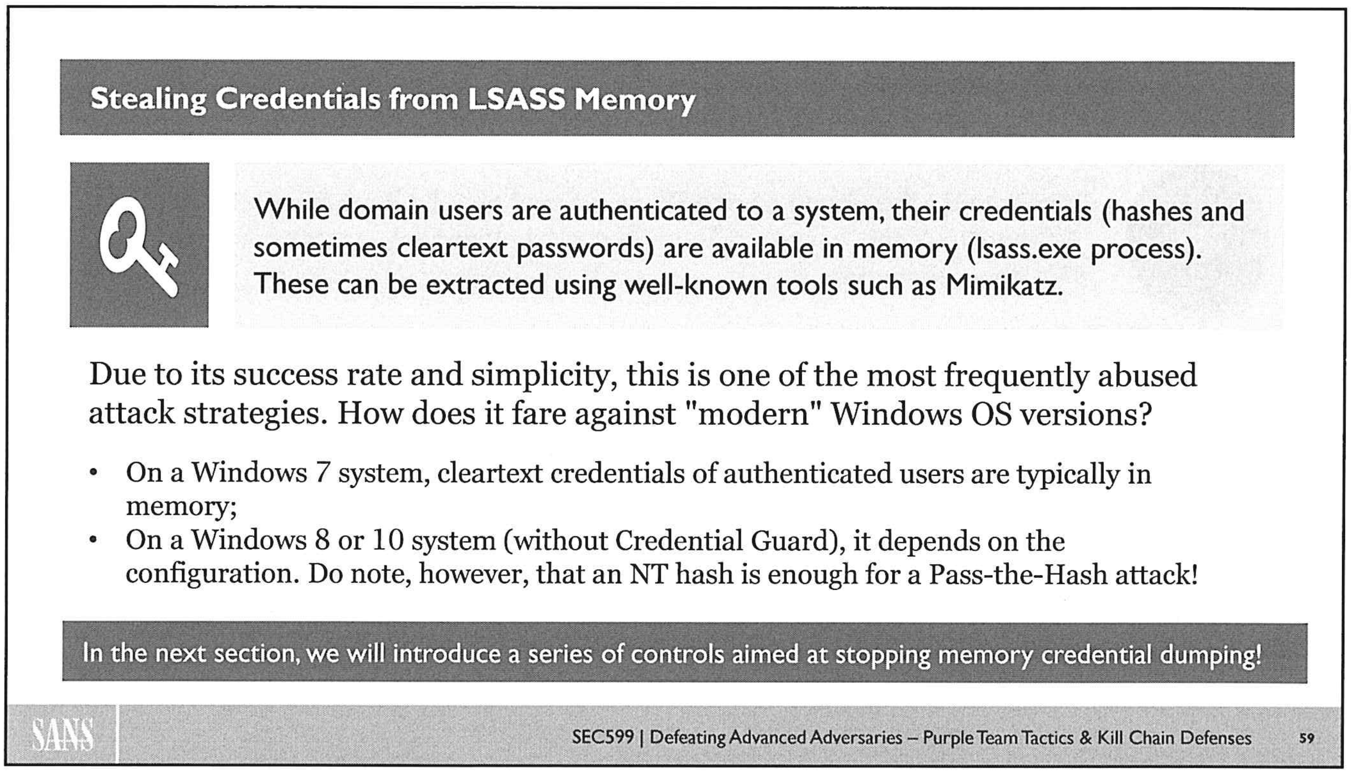
SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 58

Stealing Cached Credentials
Windows endpoints, by default, store the last 10 used domain credentials in DCC format. This is used to authenticate known users when a connection to the domain controller cannot be established. These can be extracted by for example Metasploit or Mimikatz.

There are a few items to note, however:

- Should an adversary steal cached credentials (in Domain Cached Credential, DCC format), they cannot simply be reused in a Pass-the-Hash attack, they first need to be cracked.
- The default setting of 10 cached credentials can be overwritten to be more restrictive (using GPO's). A typical workstation is only used by a few people (1 or 2 maybe) and should thus not case more credential pairs. This is even more true for servers, which are typically stationary and always connected to the domain!

© 2019 Erik Van Buggenhout & Stephen Sims



Stealing Credentials from LSASS Memory

While domain users are authenticated to a system, their credentials (hashes and sometimes cleartext passwords) are available in memory (lsass.exe process). These can be extracted using well-known tools such as Mimikatz.

Due to its success rate and simplicity, this is one of the most frequently abused attack strategies. How does it fare against "modern" Windows OS versions?

- On a Windows 7 system, cleartext credentials of authenticated users are typically in memory;
- On a Windows 8 or 10 system (without Credential Guard), it depends on the configuration. Do note, however, that an NT hash is enough for a Pass-the-Hash attack!

In the next section, we will introduce a series of controls aimed at stopping memory credential dumping!

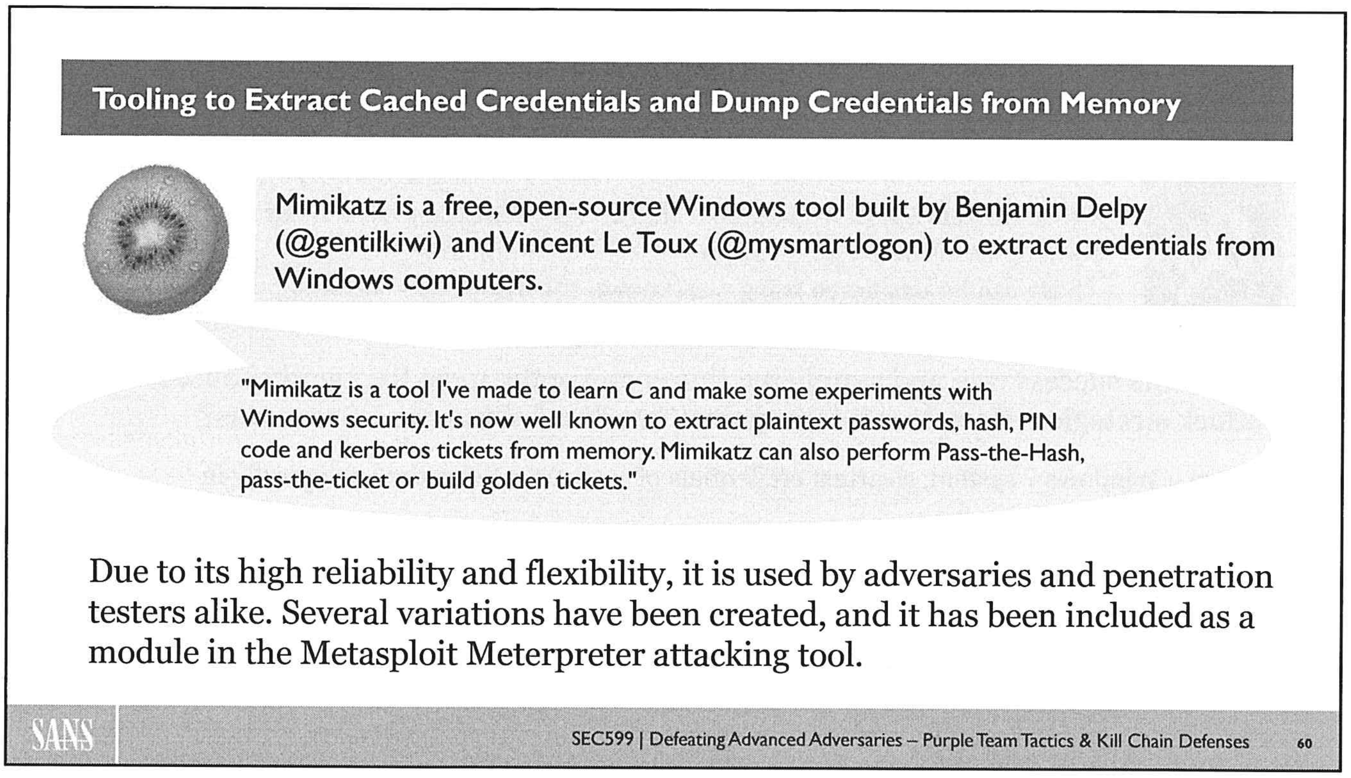
SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 59

Stealing Credentials from LSASS Memory
While domain users are authenticated to a system, their credentials (hashes and sometimes cleartext passwords) are available in memory (lsass.exe process). These can be extracted using well-known tools such as Mimikatz. Due to its success rate and simplicity, this is one of the most frequently abused attack strategies. LSASS memory credential dumping attacks are some of the most well-known attacks out there!

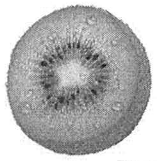
How does this attack strategy fare against "modern" Windows OS versions?

- On a Windows 7 system, cleartext credentials of authenticated users are typically in memory.
- On a Windows 8 or 10 system (without Credential Guard), it depends on the configuration. Do note, however, that an NT hash is enough for a Pass-the-Hash attack!

"What kind of controls?" In the upcoming section, we will introduce different controls, their internals, strengths and limitations!



Tooling to Extract Cached Credentials and Dump Credentials from Memory



Mimikatz is a free, open-source Windows tool built by Benjamin Delpy (@gentilkiwi) and Vincent Le Toux (@mysmartlogon) to extract credentials from Windows computers.

"Mimikatz is a tool I've made to learn C and make some experiments with Windows security. It's now well known to extract plaintext passwords, hash, PIN code and kerberos tickets from memory. Mimikatz can also perform Pass-the-Hash, pass-the-ticket or build golden tickets."

Due to its high reliability and flexibility, it is used by adversaries and penetration testers alike. Several variations have been created, and it has been included as a module in the Metasploit Meterpreter attacking tool.

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 60

Tooling to Extract Cached Credentials and Dump Credentials from Memory
Mimikatz is a tool that has many features and functions, for example, extracting hashes from the LSA process lsass.exe. It is a free, open-source Windows tool, developed by Benjamin Delpy (Twitter @gentilkiwi). It has many features:

- Extracting hashes
- Extracting passwords
- Extracting tickets
- Executing Pass-the-Hash attacks
- Executing Pass-the-Ticket attacks
- Generating golden tickets
- …

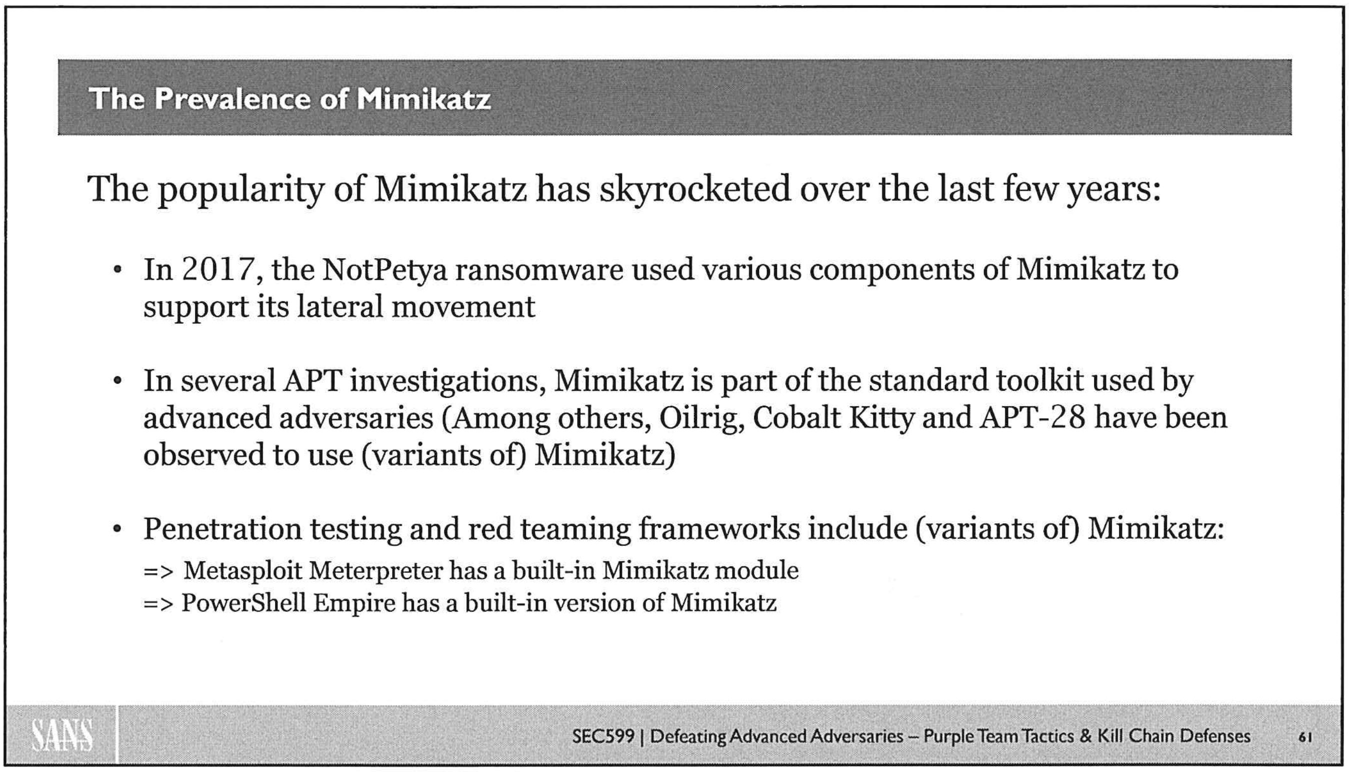
Several of these features will be explained later. Because of all these features, and constant updates with new features and support for new Windows versions, Mimikatz has become the most popular credential tool used by red teams and adversaries.

Mimikatz has 3 components (in 32-bit and 64-bit versions):

1. Mimikatz.exe: This is the executable, and the console that interacts with the user.
2. Mimilib.dll: This is the dll.
3. Mimidrv.sys: This is the kernel driver, necessary for features that require access or modifications to kernel data.

Because it is very powerful and open source, it has been transformed by hackers and malware authors for various purposes. A lot of antivirus programs detect the mimikatz.exe executable. Because this poses a problem to pen testers, file-less versions have been developed that launch directly into memory from various scripting platforms, like PowerShell.

© 2019 Erik Van Buggenhout & Stephen Sims



The Prevalence of Mimikatz

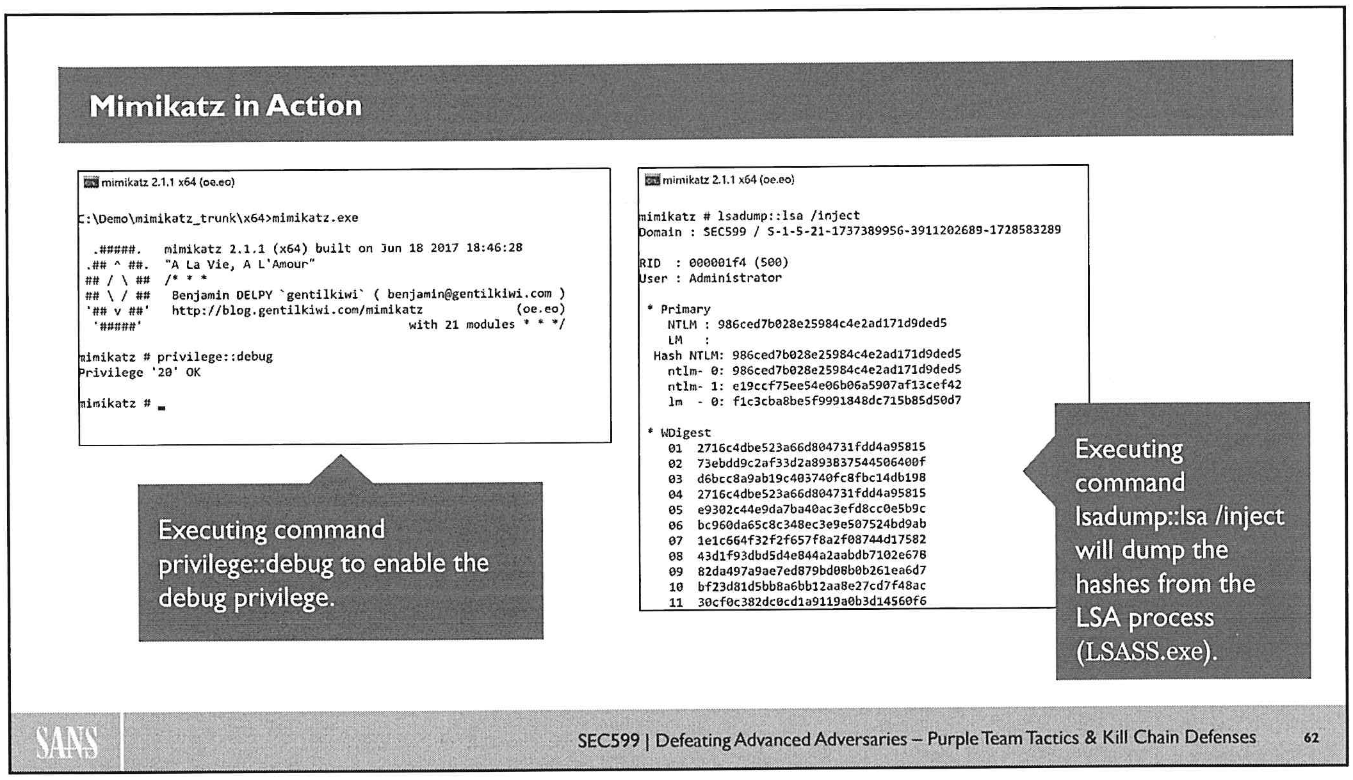
The popularity of Mimikatz has skyrocketed over the last few years:

- In 2017, the NotPetya ransomware used various components of Mimikatz to support its lateral movement
- In several APT investigations, Mimikatz is part of the standard toolkit used by advanced adversaries (Among others, Oilrig, Cobalt Kitty and APT-28 have been observed to use (variants of) Mimikatz)
- Penetration testing and red teaming frameworks include (variants of) Mimikatz:
  => Metasploit Meterpreter has a built-in Mimikatz module
  => PowerShell Empire has a built-in version of Mimikatz

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 61

The Prevalence of Mimikatz
Mimikatz has reached a "legendary" status amongst penetration testers and cyber security experts in general. Since its original release, its popularity has skyrocketed:

- In 2017, the NotPetya ransomware used various components of Mimikatz to supports its lateral movement. Once the ransomware had obtained access to a system with administrative privileges, it used Mimikatz to dump additional credentials in an attempt to further spread in the network.
- In several high-profile APT investigations, Mimikatz is part of the standard toolkit used by advanced adversaries (Among others, Oilrig, Cobalt Kitty and APT-28 have been observed to use (variants of) Mimikatz). Why reinvent the wheel, right? ☺
- Most penetration testing and red teaming frameworks include (variants of) Mimikatz: The Metasploit meterpreter has a built-in Mimikatz module (labeled Kiwi). Furthermore, PowerShell Empire has an "Invoke-Mimikatz" module, which is fully implemented in PowerShell!



Mimikatz in Action

mimikatz 2.1.1 x64 (oe.eo)

```
C:\Demo\mimikatz_trunk\x64>mimikatz.exe

  .#####.   mimikatz 2.1.1 (x64) built on Jun 18 2017 18:46:28
 .## ^ ##.  "A La Vie, A L'Amour"
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                     with 21 modules * * */

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # _
```

Executing command privilege::debug to enable the debug privilege.

mimikatz 2.1.1 x64 (oe.eo)

```
mimikatz # lsadump::lsa /inject
Domain : SEC599 / S-1-5-21-1737389956-3911202689-1728583289

RID  : 000001f4 (500)
User : Administrator

 * Primary
    NTLM : 986ced7b028e25984c4e2ad171d9ded5
    LM   :
  Hash NTLM: 986ced7b028e25984c4e2ad171d9ded5
    ntlm- 0: 986ced7b028e25984c4e2ad171d9ded5
    ntlm- 1: e19ccf75ee54e06b06a5907af13cef42
    lm  - 0: f1c3cba8be5f9991848dc715b85d50d7

 * WDigest
    01  2716c4dbe523a66d804731fdd4a95815
    02  73ebdd9c2af33d2a893837544506400f
    03  d6bcc8a9ab19c403740fc8fbc14db198
    04  2716c4dbe523a66d804731fdd4a95815
    05  e9302c44e9da7ba40ac3efd8cc0e5b9c
    06  bc960da65c8c348ec3e9e507524bd9ab
    07  1e1c664f32f2f657f8a2f08744d17582
    08  43d1f93dbd5d4e844a2aabdb7102e678
    09  82da497a9ae7ed879bd08b0b261ea6d7
    10  bf23d81d5bb8a6bb12aa8e27cd7f48ac
    11  30cf0c382dc0cd1a9119a0b3d14560f6
```

Executing command lsadump::lsa /inject will dump the hashes from the LSA process (LSASS.exe).

SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 62

Mimikatz in Action

When mimikatz is launched, a banner is displayed, and the user can start to type commands. Commands are grouped into modules. A command has to be prefixed with the module name followed by :: (except for the standard module). In the example above, command privilege::debug is executed. privilege is the name of the module, and debug is the name of the command.

A normal user can access all the processes (including their memory) that run under their user account. They cannot access the memory of processes running under other users, like system, for example. An administrator can access all the processes, including processes running under other accounts. This is possible because administrators have the debug privilege. Having this privilege is a mandatory requirement to access the memory of system processes, for example.

Besides having the privilege, the privilege also has to be "enabled" before one can use this privilege. Many tools (like debuggers) do this automatically, but with mimikatz, it is optional. The command privilege::debug has to be executed to enable the debug privilege. Module lsadump contains several commands to extract information from the Local Security Authority process. The lsa command will extract hashes and other secrets from the LSA process. It requires admin rights, and the debug privilege must be enabled.

There are two methods to extract hashes from the LSA process. Option /patch will modify Windows code to be able to dump hashes, and option /inject will inject extra code into the LSA process to dump hashes.

In the screenshot above, we see the use of Mimikatz's lsadump::lsa command with option /inject on a domain controller. This command will dump hashes for all users in an Active Directory domain managed by this domain controller.

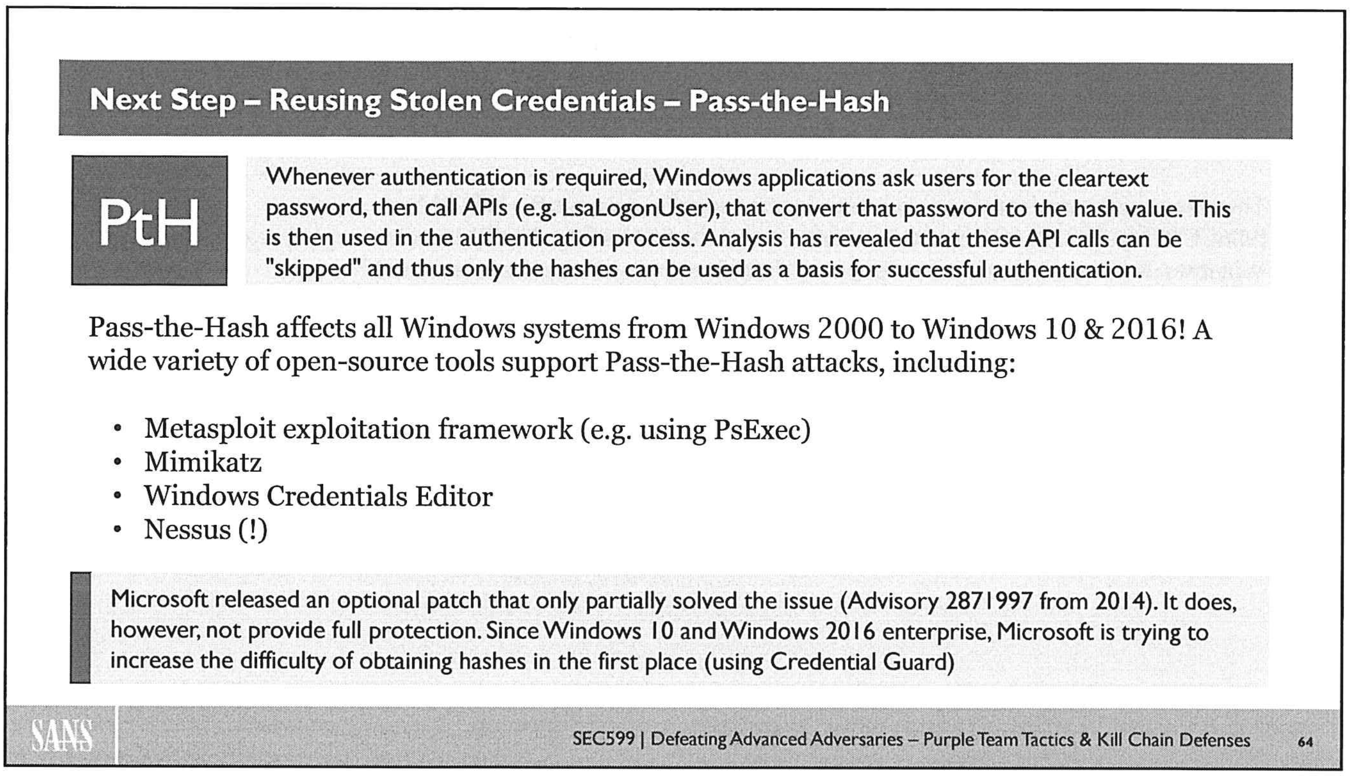
The first line gives us the domain: SEC599, together with the SID of the domain. The Security ID of the domain is the string that starts with S-1-5-21- …

© 2019 Erik Van Buggenhout & Stephen Sims

After that, Mimikatz dumps hashes for each user. The RID is the Relative ID and is appended to a base SID. In this example, it is 500 because the account is the Administrator account (local Administrator accounts and Domain Administrator accounts have RID 500). The SID of the Administrator account is the concatenation of the domain SID and the users' RID: S-1-5-21-...-500.

Then the primary credentials are dumped. In this example, we have the NTLM hash (986CED7B028E25984C4E2AD171D9DED5) and the LM hash (). The LM hash is empty because this is a Windows 2016 server domain controller: By default, storing LM hashes is turned off.

The following NTLM hashes are the password history hashes (used to enforce password rotation).



Next Step – Reusing Stolen Credentials – Pass-the-Hash

PtH

Whenever authentication is required, Windows applications ask users for the cleartext password, then call APIs (e.g. LsaLogonUser), that convert that password to the hash value. This is then used in the authentication process. Analysis has revealed that these API calls can be "skipped" and thus only the hashes can be used as a basis for successful authentication.

Pass-the-Hash affects all Windows systems from Windows 2000 to Windows 10 & 2016! A wide variety of open-source tools support Pass-the-Hash attacks, including:

- Metasploit exploitation framework (e.g. using PsExec)
- Mimikatz
- Windows Credentials Editor
- Nessus (!)

Microsoft released an optional patch that only partially solved the issue (Advisory 2871997 from 2014). It does, however, not provide full protection. Since Windows 10 and Windows 2016 enterprise, Microsoft is trying to increase the difficulty of obtaining hashes in the first place (using Credential Guard)

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 64

Next Step – Reusing Stolen Credentials – Pass-the-Hash
If we go back to the explanation of NTLM authentication (challenge/response), we remember that it is actually the NTLM hash of the password that is used to calculate the response, and not the password itself.

This means that if we can obtain the hash of a password (and not the password itself), we still must be able to authenticate. A pass the hash attack is exactly this: A stolen hash (example extracted from Active Directory database) is used to authenticate.

This cannot be done just using built-in Windows utilities or commands; there is no command (for example net use) that will take a hash as an argument instead of a password.

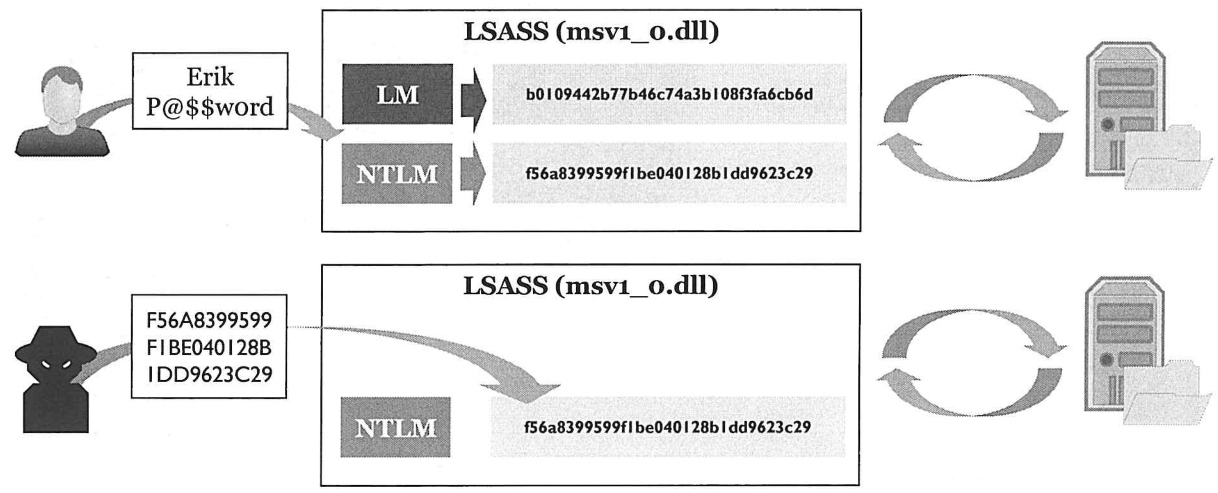
But there are adversary tools available that allow the use of a hash to authenticate (for example Windows Credential Editor and Mimikatz). These fall into two categories: tools that implement the NTLM authentication algorithm and take a hash as input, and tools that inject a hash into Windows memory for use by built-in Windows NTLM authentication. The advantage for an adversary to use a Pass-the-Hash attack is that the password is not required (e.g. time-consuming password cracking is not necessary).

Protecting against Pass-the-Hash attacks is virtually impossible: The hash is used in NTLM authentication and is the only secret necessary to calculate the response. It can only be protected against by preventing the use of NTLM authentication, or by making sure hashes cannot be obtained. Kerberos authentication with tickets is the authentication mechanism that replaces NTLM. Unfortunately, there are many cases where Kerberos authentication cannot work and then Windows falls back to NTLM authentication. It is possible to disable NTLM authentication via the registry so that only Kerberos authentication can be used, but in our experience, this is not a viable option for corporate networks. Corporate infrastructure has too many "legacy" systems and applications that require NTLM authentication.

Microsoft released an optional patch that only partially solved the issue (Advisory 2871997 from 2014). It does, however, not provide full protection, as it only protects against PtH attacks focused on local accounts (so not domain-level accounts) AND not the default RID-500 local administrator account.

© 2019 Erik Van Buggenhout & Stephen Sims

Next Step – Reusing Stolen Credentials – Pass-the-Hash – Details


LSASS (msv1_0.dll)
Erik
P@$$word
LM
b0109442b77b46c74a3b108f3fa6cb6d
NTLM
f56a8399599f1be040128b1dd9623c29
LSASS (msv1_0.dll)
F56A8399599
F1BE040128B
1DD9623C29
NTLM
f56a8399599f1be040128b1dd9623c29

Illustrations inspired by "Abusing Microsoft Kerberos - Sorry you guys don't get it," Benjamin Delpy (Blackhat USA 2014)

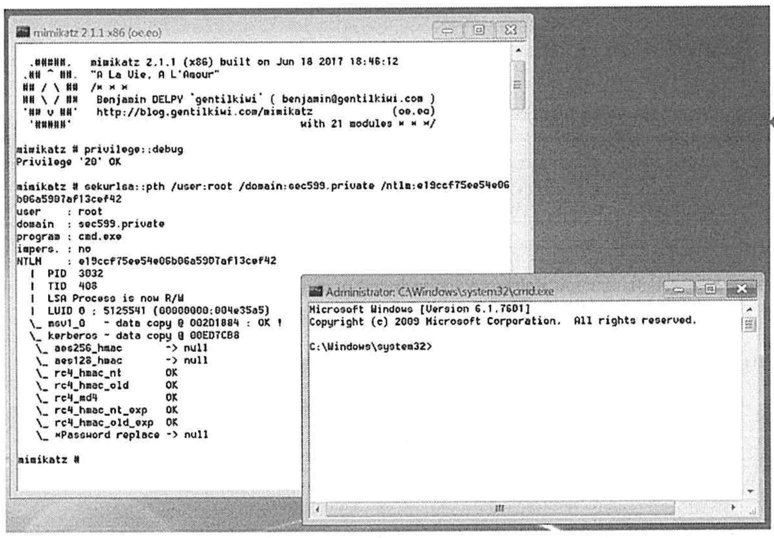
Next Step – Reusing Stolen Credentials – Pass-the-Hash – Details
So how does Pass-the-Hash work in detail? LSA is the Local Security Authority, which is a "protected subsystem" of the Operating System. It treats all aspects of the local system security and provides multiple authentication services (security packages). Lsass.exe (Local Security Authority Sub System) boots up when the machine starts and is thus one of the core processes of Windows.

Windows Single Sign On (SSO) is implemented in LSASS. LSASS supports Kerberos (through kerberos.dll), NTLM (msv1_0.dll) and digest authentication (wdigest.dll). Once the user is authenticated, a certain form of his credentials is stored in LSASS.

A user signs on with the entry of their username and password. After this, all files and functions he/she has cleared for are accessible without having to enter credentials anymore. This concept is referred to as Single Sign On and is implemented in Windows with LSASS. LSASS supports Kerberos (kerberos.dll), NTLM (msv1_0.dll) or Digest Authentication (wdigest.dll). After a user's authentication, the credentials are stored in the memory of the system. This is done so that the security packages can access it. Depending on the package, the password is stored as a hash value, encrypted or even in plaintext.

The trick with Pass-the-Hash is that an adversary can immediately store the NTLM hash of a user's password in LSASS, without knowing the actual password value! The illustrations created in the slide above were inspired by Benjamin Delpy's "Abusing Microsoft Kerberos - Sorry you guys don't get it", which was presented at Blackhat USA 2014.

Next Step – Reusing Stolen Credentials – Pass-the-Hash - Example



```
mimikatz 2.1.1 x86 (oe.eo)

  .#####.   mimikatz 2.1.1 (x86) built on Jun 18 2017 18:46:12
 .## ^ ##.  "A La Vie, A L'Amour"
 ## / \ ##  /* * *
 ## \ / ##   Benjamin DELPY `gentilkiwi` ( benjamin@gentilkiwi.com )
 '## v ##'   http://blog.gentilkiwi.com/mimikatz             (oe.eo)
  '#####'                                  with 21 modules * * */

mimikatz # privilege::debug
Privilege '20' OK

mimikatz # sekurlsa::pth /user:root /domain:sec599.private /ntlm:e19ccf75ee54e06
b06a5907af13cef42
user    : root
domain  : sec599.private
program : cmd.exe
impers. : no
NTLM    : e19ccf75ee54e06b06a5907af13cef42
  |  PID  3032
  |  TID  408
  |  LSA Process is now R/W
  |  LUID 0 ; 5125541 (00000000:004e35a5)
  \_ msv1_0   - data copy @ 002D1884 : OK !
  \_ kerberos - data copy @ 00ED7CB8
   \_ aes256_hmac       -> null
   \_ aes128_hmac       -> null
   \_ rc4_hmac_nt       OK
   \_ rc4_hmac_old      OK
   \_ rc4_md4           OK
   \_ rc4_hmac_nt_exp   OK
   \_ rc4_hmac_old_exp  OK
   \_ *Password replace -> null

mimikatz #
```

```
Administrator: C:\Windows\system32\cmd.exe

Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation.  All rights reserved.

C:\Windows\system32>
```

PtH with Mimikatz
In the example on the left, Mimikatz is used to perform a PtH attack against our target system.

In the first step, Mimikatz is set up with debug privileges;

Afterwards, a previously stolen NTLM hash for user "root" is injected and used to spawn a cmd.exe window.

Next Step – Reusing Stolen Credentials – Pass-the-Hash – Example
Mimikatz is a tool that allows users to execute Pass-the-Hash attacks by injecting hashes into Windows memory.

In the example above, Mimikatz is executed as administrator and the debug privilege is enabled (privilege::debug command). This is necessary because Mimikatz will write into the LSA process to inject the hash. This cannot be done without administrative rights.

The Mimikatz command to start a Pass-the-Hash attack is sekurlsa::pth. It has 3 mandatory arguments: The username (root in our example), the domain name (sec599.private in our example) and the NTLM hash (E19CCF75EE54E06B06A5907AF13CEF42 in our example, which is the NTLM hash of P@ssw0rd).

With this information, Mimikatz will launch a command-line process (cmd.exe) while opening the LSA process memory to inject the hash we provided for the security tokens used by the cmd.exe program.

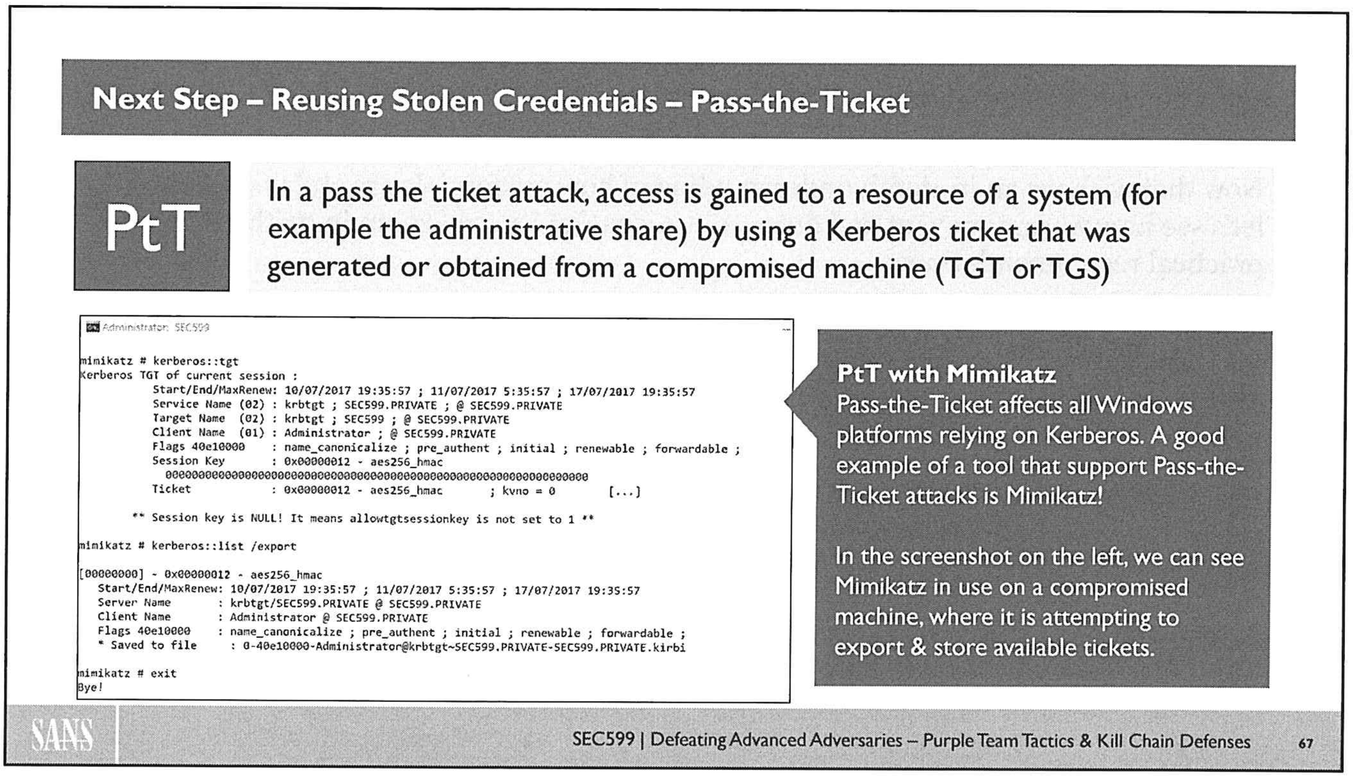
This cmd.exe program is running with the injected credentials, and if we would use a "net use *" command (without providing credentials) to connect to a remote share, the NTLM challenge/response would be executed with the NTLM hash we injected, and thus execute a Pass-the-Hash attack.

Programs started by this cmd.exe process will also inherit the injected credentials.

The sekurlsa::pth command can also be used to start other programs than cmd.exe.

Credential Guard will protect against Pass-the-Hash attacks with tools that inject the hash into memory because reading/writing the credentials in memory is no longer possible (cfr. next chapter). But of course, this requires that all your Windows machines run Windows 10/2016 Enterprise.

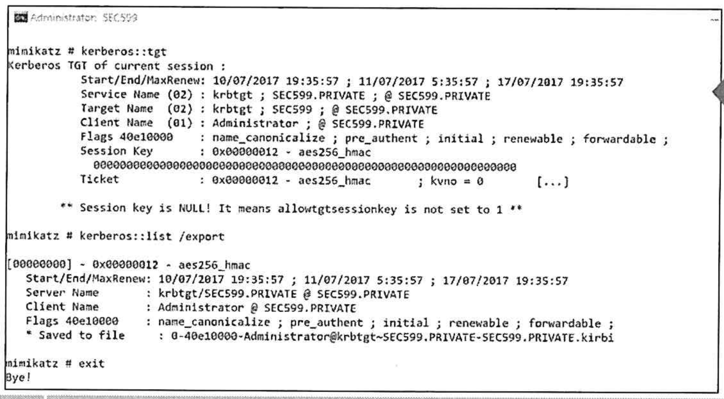
© 2019 Erik Van Buggenhout & Stephen Sims



Next Step – Reusing Stolen Credentials – Pass-the-Ticket

PtT

In a pass the ticket attack, access is gained to a resource of a system (for example the administrative share) by using a Kerberos ticket that was generated or obtained from a compromised machine (TGT or TGS)



```
Administrator: SEC599

mimikatz # kerberos::tgt
Kerberos TGT of current session :
           Start/End/MaxRenew: 10/07/2017 19:35:57 ; 11/07/2017 5:35:57 ; 17/07/2017 19:35:57
           Service Name (02) : krbtgt ; SEC599.PRIVATE ; @ SEC599.PRIVATE
           Target Name  (02) : krbtgt ; SEC599 ; @ SEC599.PRIVATE
           Client Name  (01) : Administrator ; @ SEC599.PRIVATE
           Flags 40e10000    : name_canonicalize ; pre_authent ; initial ; renewable ; forwardable ;
           Session Key       : 0x00000012 - aes256_hmac
             0000000000000000000000000000000000000000000000000000000000000000
           Ticket            : 0x00000012 - aes256_hmac       ; kvno = 0        [...]

        ** Session key is NULL! It means allowtgtsessionkey is not set to 1 **

mimikatz # kerberos::list /export

[00000000] - 0x00000012 - aes256_hmac
   Start/End/MaxRenew: 10/07/2017 19:35:57 ; 11/07/2017 5:35:57 ; 17/07/2017 19:35:57
   Server Name       : krbtgt/SEC599.PRIVATE @ SEC599.PRIVATE
   Client Name       : Administrator @ SEC599.PRIVATE
   Flags 40e10000    : name_canonicalize ; pre_authent ; initial ; renewable ; forwardable ;
   * Saved to file     : 0-40e10000-Administrator@krbtgt~SEC599.PRIVATE-SEC599.PRIVATE.kirbi

mimikatz # exit
Bye!
```

PtT with Mimikatz
Pass-the-Ticket affects all Windows platforms relying on Kerberos. A good example of a tool that support Pass-the-Ticket attacks is Mimikatz!

In the screenshot on the left, we can see Mimikatz in use on a compromised machine, where it is attempting to export & store available tickets.

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 67

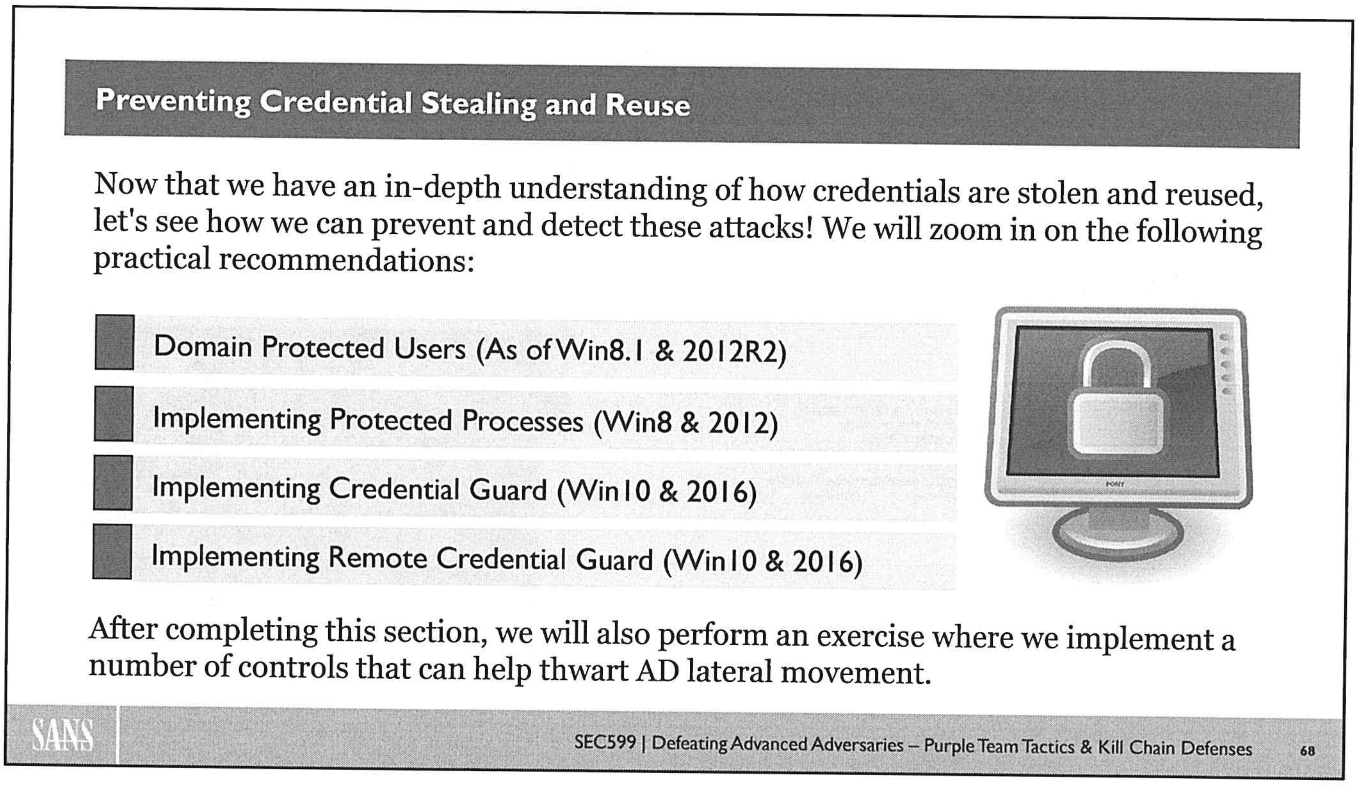
Next Step – Reusing Stolen Credentials – Pass-the-Ticket

Since tickets can be used to obtain access to a service or request other tickets, it should be no surprise that adversaries found out ways to steal and reuse tickets. Such an attack is called a pass-the-ticket attack. Adversaries obtain existing tickets (preferably ticket-granting-tickets from administrative accounts) by extracting them from the memory of compromised machines, and then use them to gain access to other machines. There are open-source attack tools available to extract tickets and execute pass-the-ticket attacks.

Abusing a ticket implies that we are abusing Kerberos authentication and authorization: The advantage for an adversary to use an attack with a ticket is that he/she does not require the password or the hash of the compromised account. Just a ticket.

With Mimikatz, we can display the TGT by using command kerberos::tgt. As you can see from the output, the service name is krbtgt for domain sec599.private (hence it is a TGT) and the client name is administrator at domain sec599.private. So, this is a very valuable ticket to steal: It is the TGT of the Domain Administrator. This ticket will provide access to all resources of the domain. From the start and end time, we can see that this ticket is valid for 10 hours. To obtain this ticket, the adversaries need to have (local) administrative access to a machine where the Domain Administrator is logged on. This ticket can be exported to a file, for example. For this we issue the command "kerberos::list /export": this will export all tickets to a file (one for each ticket) in the current directory.

We cannot select the name for the tickets; it is Mimikatz that decides the name based on metadata like the username. Tickets generated by Mimikatz use the extension .kirbi, but any extension can be used for a pass-the-ticket attack.



Preventing Credential Stealing and Reuse

Now that we have an in-depth understanding of how credentials are stolen and reused, let's see how we can prevent and detect these attacks! We will zoom in on the following practical recommendations:

- Domain Protected Users (As of Win8.1 & 2012R2)
- Implementing Protected Processes (Win8 & 2012)
- Implementing Credential Guard (Win10 & 2016)
- Implementing Remote Credential Guard (Win10 & 2016)

After completing this section, we will also perform an exercise where we implement a number of controls that can help thwart AD lateral movement.

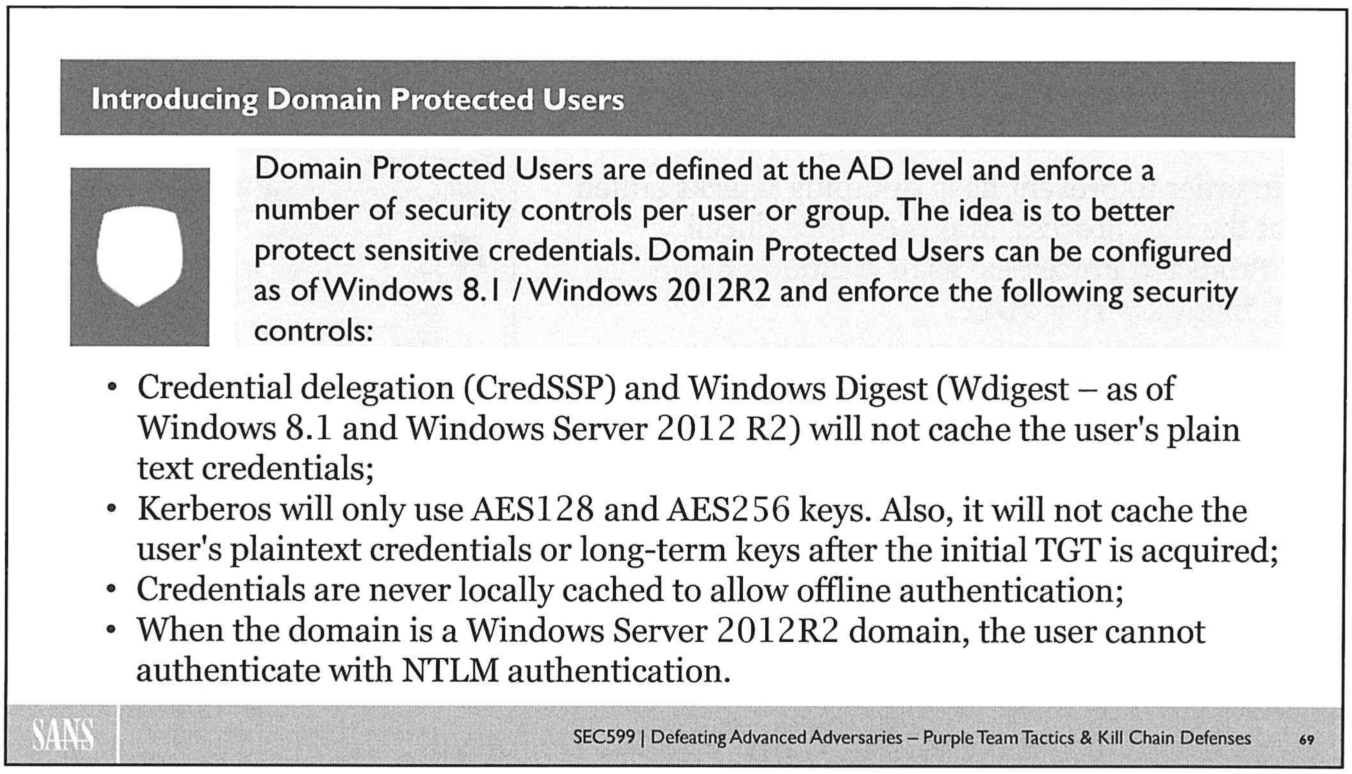
SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 68

Preventing Credential Stealing and Reuse
Now that we have an in-depth understanding of how credentials are stolen and reused, let's see how we can prevent and detect these attacks! We will zoom in on the following practical recommendations:

- Domain Protected Users can be defined on AD level to enforce a number of security controls on a selection of users (e.g. privileged users).
- Protected Processes are a control available as of Windows 8 and 2012. They attempt to prevent the dumping of credentials from the LSASS process, but it can be bypassed by tools such as Mimikatz.
- Credential Guard is an excellent new feature as of Windows 10 that will isolate the LSASS process in an attempt to protect it from "hash dumping" attacks.
- Remote Credential Guard is aimed at protecting your credentials when a Remote Desktop session is set up.

We will zoom in on these controls!

© 2019 Erik Van Buggenhout & Stephen Sims



Introducing Domain Protected Users

Domain Protected Users are defined at the AD level and enforce a number of security controls per user or group. The idea is to better protect sensitive credentials. Domain Protected Users can be configured as of Windows 8.1 / Windows 2012R2 and enforce the following security controls:

- Credential delegation (CredSSP) and Windows Digest (Wdigest – as of Windows 8.1 and Windows Server 2012 R2) will not cache the user's plain text credentials;
- Kerberos will only use AES128 and AES256 keys. Also, it will not cache the user's plaintext credentials or long-term keys after the initial TGT is acquired;
- Credentials are never locally cached to allow offline authentication;
- When the domain is a Windows Server 2012R2 domain, the user cannot authenticate with NTLM authentication.

SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 69

Introducing Domain Protected Users
Domain Protected Users are defined at the AD level and enforce a number of security controls per user or group. The idea is to better protect sensitive credentials. Domain Protected Users can be configured as of Windows 8.1 / Windows 2012R2 (parts of the functionality can be installed and enabled on Windows 7) and enforce the following security controls:

- Credential delegation (CredSSP) and Windows Digest (Wdigest – as of Windows 8.1 and Windows Server 2012 R2) will not cache the user's plaintext credentials in LSASS. This control aims to defeat credential dumping from the LSASS process memory.
- Kerberos will no longer create DES or RC4 keys, it will only use AES128 and AES256. Also, it will not cache the user's plaintext credentials or long-term keys after the initial TGT is acquired.
- Credentials are never locally cached to allow offline authentication.
- When the domain is a Windows Server 2012R2 domain, the user cannot authenticate with NTLM authentication.

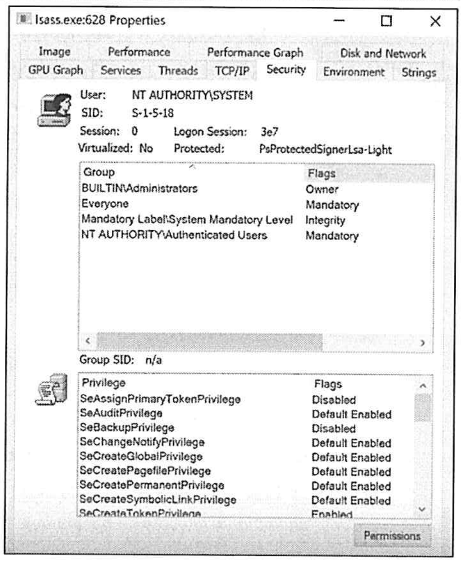
Although not fully bulletproof, these controls provide an interesting additional security layer! Additional information can be found on the following Microsoft resource page:

https://docs.microsoft.com/en-us/windows-server/security/credentials-protection-and-management/protected-users-security-group

Introducing Protected Processes

In order to prevent hash dumping attacks aimed at the LSA process, Microsoft introduced "Protected Processes" as of Windows 8 and Windows Server 2012.

- Protected Processes were first introduced in Windows Vista for DRM (Digital Rights Management) purposes, but were adapted for "security purposes" in Windows 8
- The screenshot on the right provides an example of the lsass.exe process running as a "protected process"
- Protected Processes are implemented in the Kernel software and can thus be defeated...



```
User:        NT AUTHORITY\SYSTEM
SID:         S-1-5-18
Session:     0        Logon Session:   3e7
Virtualized: No       Protected:       PsProtectedSignerLsa-Light

Group                                      Flags
BUILTIN\Administrators                     Owner
Everyone                                   Mandatory
Mandatory Label\System Mandatory Level     Integrity
NT AUTHORITY\Authenticated Users           Mandatory

Group SID:  n/a

Privilege                                  Flags
SeAssignPrimaryTokenPrivilege              Disabled
SeAuditPrivilege                           Default Enabled
SeBackupPrivilege                          Disabled
SeChangeNotifyPrivilege                    Default Enabled
SeCreateGlobalPrivilege                    Default Enabled
SeCreatePagefilePrivilege                  Default Enabled
SeCreatePermanentPrivilege                 Default Enabled
SeCreateSymbolicLinkPrivilege              Default Enabled
SeCreateTokenPrivilege                     Enabled
```

Introducing Protected Processes
The next two protection methods we will discuss not only apply to domain controllers, but to all Windows machines. The Local Security Authority is the process that handles the credentials. As we have seen in Active Directory attacks, many powerful tools are available to adversaries to extract credentials from the lsass.exe process.

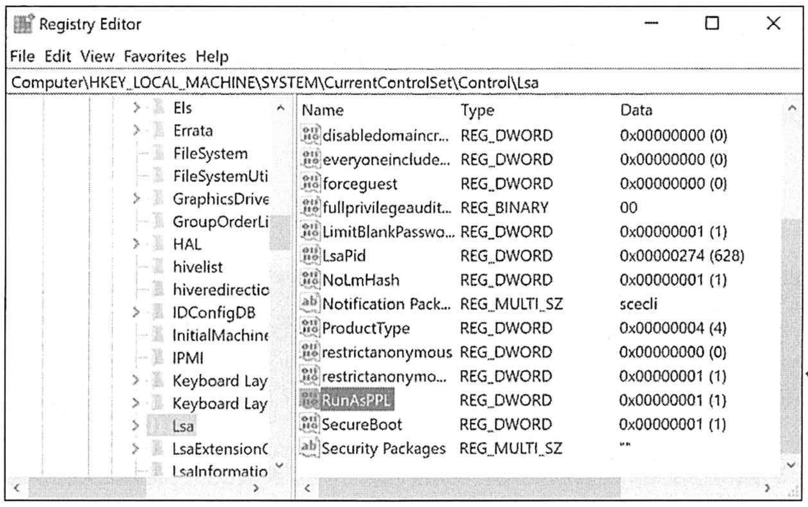
With Windows Server 2012 (and Windows 8), the LSA can be configured to have its lsass.exe process hosted as a protected process. Under Windows, with the correct privileges, accounts can access the memory of any Windows process. This is not the case with Protected Processes. Protected Processes (and their memory) cannot be accessed by other processes, regardless of the account they run with. Protected Processes were introduced with Windows Vista for DRM purposes (to make media players), but Protected Processes were repurposed for security when Windows 8 was introduced.

By running the lsass.exe process as a protected process, tools like Mimikatz cannot access the process to extract credentials. In the screenshot above, we can see that the lsass.exe process running on this machine is protected: Process Explorer's security tab indicates that the process is protected (PsProtectedSignerLsa-Light).

Protected Processes are implemented in the Kernel software and can thus be defeated... Mimikatz has a function to remove the protection from Protected Processes: This "converts" the process into a normal process. To do this, Mimikatz requires its kernel driver to be installed. Installation of this kernel driver can, however, be detected and responded to.

© 2019 Erik Van Buggenhout & Stephen Sims

How to Configure a Process to Be "Protected"?



```
Registry Editor
File  Edit  View  Favorites  Help
Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa

Els                 Name                  Type           Data
Errata              disabledomaincr...    REG_DWORD      0x00000000 (0)
FileSystem          everyoneinclude...    REG_DWORD      0x00000000 (0)
FileSystemUti       forceguest            REG_DWORD      0x00000000 (0)
GraphicsDrive       fullprivilegeaudit... REG_BINARY     00
GroupOrderLi        LimitBlankPasswo...   REG_DWORD      0x00000001 (1)
HAL                 LsaPid                REG_DWORD      0x00000274 (628)
hivelist            NoLmHash              REG_DWORD      0x00000001 (1)
hiveredirectio      Notification Pack...  REG_MULTI_SZ   scecli
IDConfigDB          ProductType           REG_DWORD      0x00000004 (4)
InitialMachine      restrictanonymous     REG_DWORD      0x00000000 (0)
IPMI                restrictanonymo...    REG_DWORD      0x00000001 (1)
Keyboard Lay        RunAsPPL              REG_DWORD      0x00000001 (1)
Keyboard Lay        SecureBoot            REG_DWORD      0x00000001 (1)
Lsa                 Security Packages     REG_MULTI_SZ   ""
LsaExtensionC
LsaInformatio
```

A new registry key with a DWORD name of RunAsPPL, and value 1 should be created to run LSA as a protected process.

How to Configure a Process to Be "Protected"?

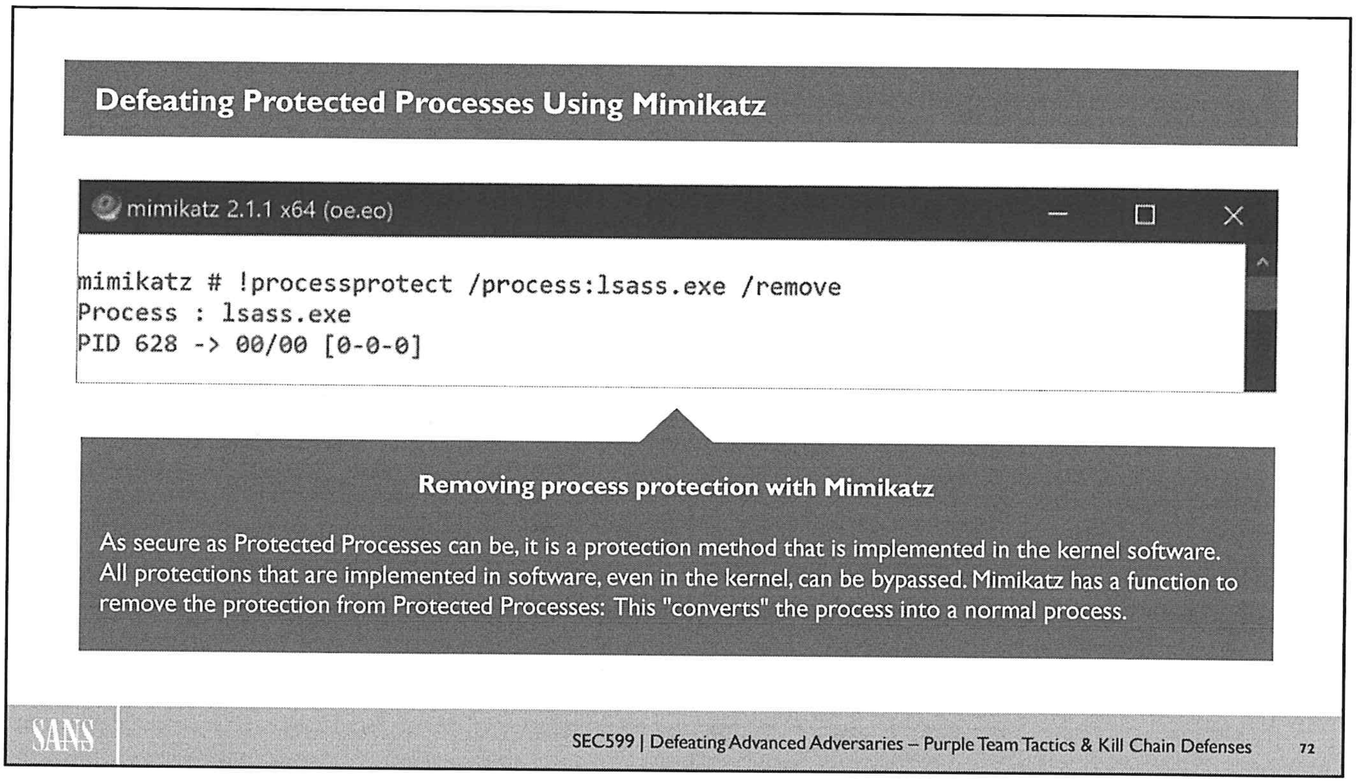
To configure Windows to run the lsass.exe process as a protected process, a change has to be made. By default, Windows will not run the lsass.exe process as a protected process.

A registry value has to be created and set: Under the registry key HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa, a double word value (DWORD) has to be created. The name of this DWORD is RunAsPPL, and its value should be set to 1.

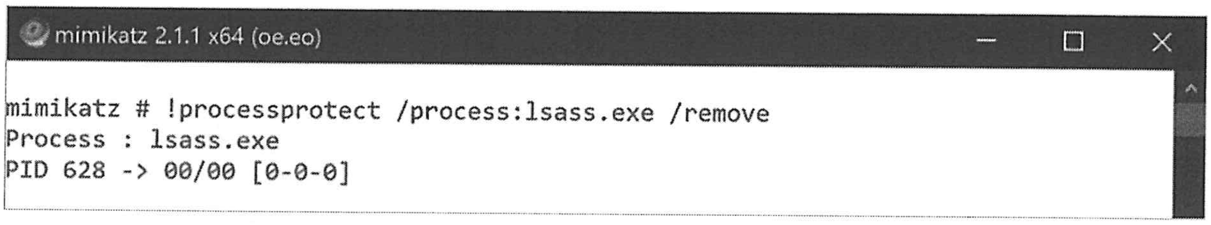
Then, Windows needs to be rebooted, after which the LSA process will run as a protected process.

This requires at least Windows 2012 (or Windows 8).

© 2019 Erik Van Buggenhout & Stephen Sims



Defeating Protected Processes Using Mimikatz



```
mimikatz # !processprotect /process:lsass.exe /remove
Process : lsass.exe
PID 628 -> 00/00 [0-0-0]
```

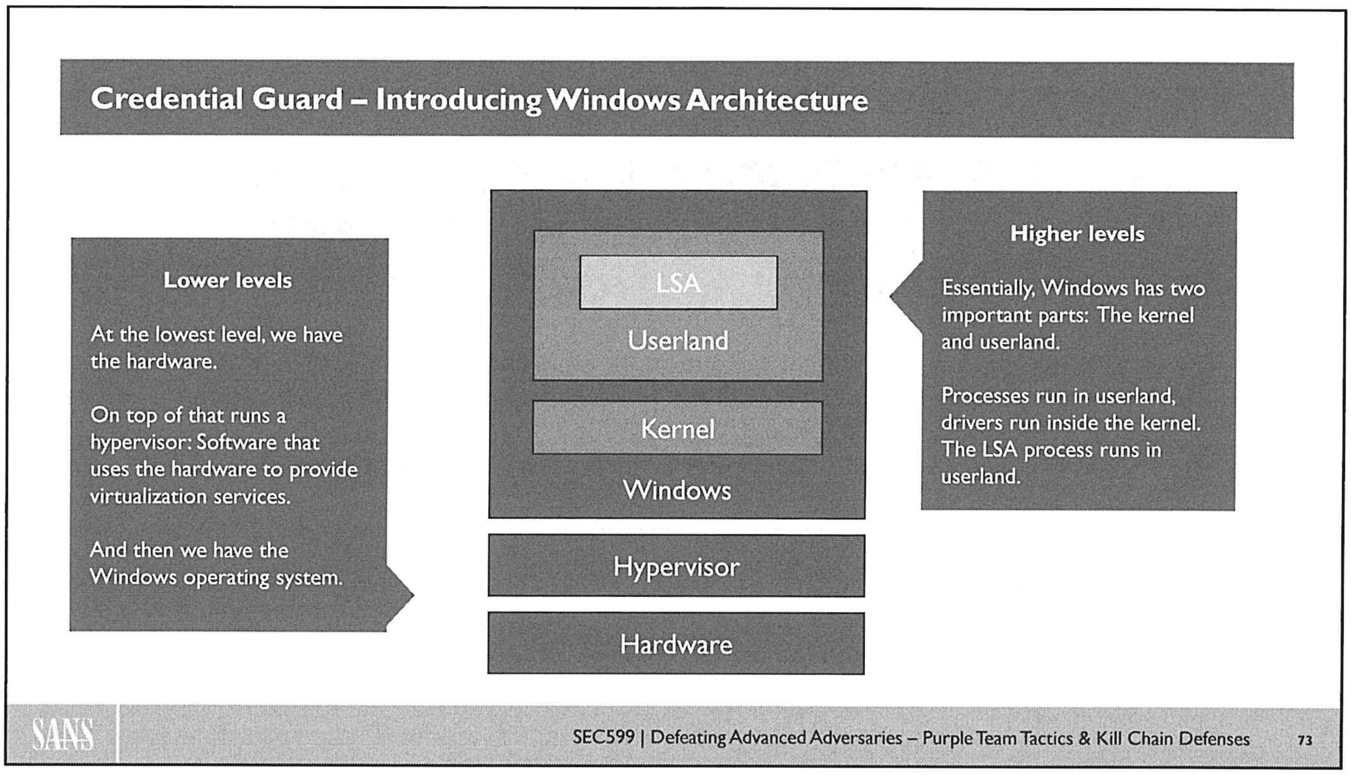
Removing process protection with Mimikatz

As secure as Protected Processes can be, it is a protection method that is implemented in the kernel software. All protections that are implemented in software, even in the kernel, can be bypassed. Mimikatz has a function to remove the protection from Protected Processes: This "converts" the process into a normal process.

Defeating Protected Processes Using Mimikatz
As secure as Protected Processes can be, it is a protection method that is implemented in the kernel software. All protections that are implemented in software, even in the kernel, can be bypassed. Mimikatz has a function to remove the protection from Protected Processes: This "converts" the process into a normal process. To do this, mimikatz requires its kernel driver to be installed.

This can be done with the command !+
The next step is to use the kernel command processprotect to remove the protection of the lsass.exe process: !processprotect /process:lsass.exe /remove.

When this is done, mimikatz (or other tools) can be used to extract the credentials, as the memory of the LSA process is no longer protected.



Credential Guard – Introducing Windows Architecture

Lower levels

At the lowest level, we have the hardware.

On top of that runs a hypervisor: Software that uses the hardware to provide virtualization services.

And then we have the Windows operating system.

Higher levels

Essentially, Windows has two important parts: The kernel and userland.

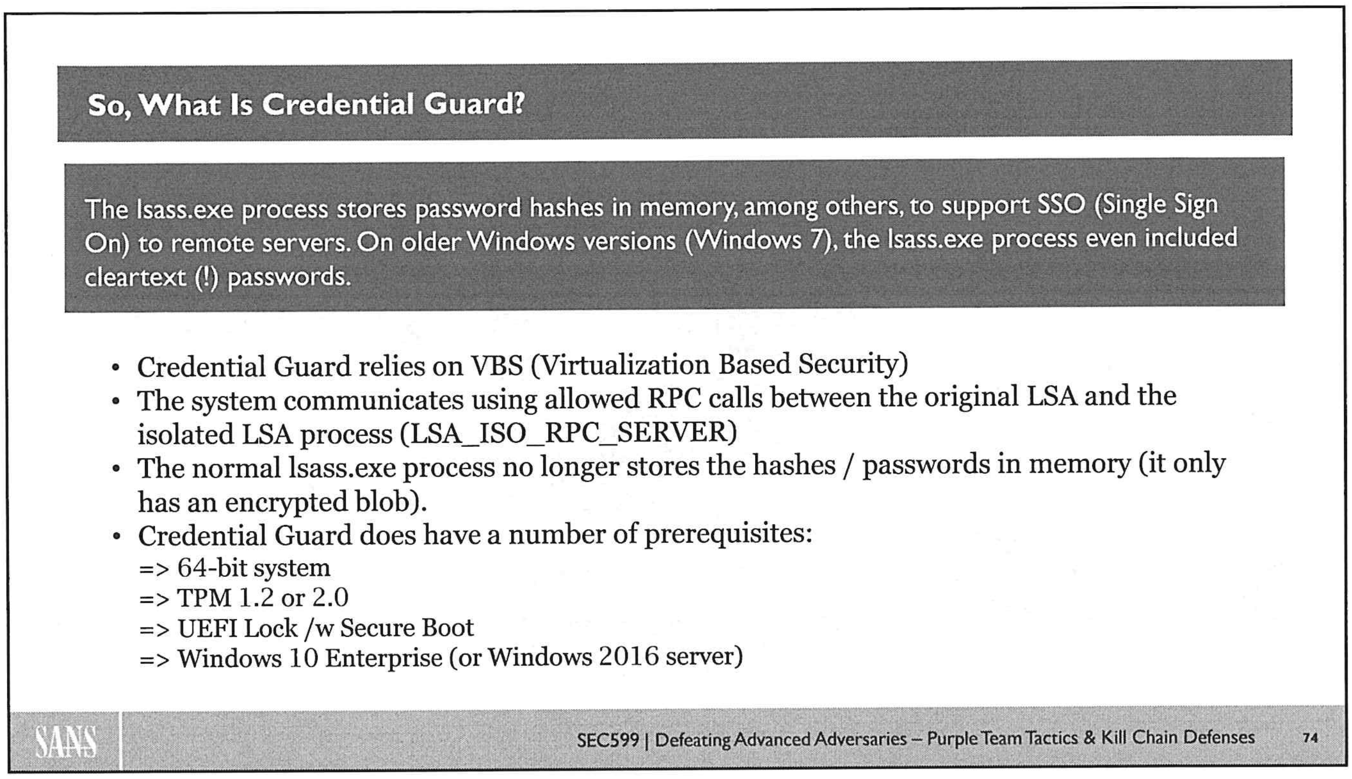
Processes run in userland, drivers run inside the kernel. The LSA process runs in userland.

Credential Guard – Introducing Windows Architecture
As a pure software solution like Protected Processes can be bypassed, Microsoft developed a hardware-based solution: Credential Guard. In order to understand Credential Guard, we first need to understand the normal architecture of Windows.

The simplified diagram above describes this architecture.

At the lowest level, we have the hardware. On top of that runs a hypervisor: Software that uses the hardware to provide virtualization services. And then we have the Windows operating system.

Essentially, Windows has two important parts: The kernel and userland. Processes run in userland; drivers run inside the kernel. The LSA process runs in userland.



So, What Is Credential Guard?

The lsass.exe process stores password hashes in memory, among others, to support SSO (Single Sign On) to remote servers. On older Windows versions (Windows 7), the lsass.exe process even included cleartext (!) passwords.

- Credential Guard relies on VBS (Virtualization Based Security)
- The system communicates using allowed RPC calls between the original LSA and the isolated LSA process (LSA_ISO_RPC_SERVER)
- The normal lsass.exe process no longer stores the hashes / passwords in memory (it only has an encrypted blob).
- Credential Guard does have a number of prerequisites:
  => 64-bit system
  => TPM 1.2 or 2.0
  => UEFI Lock /w Secure Boot
  => Windows 10 Enterprise (or Windows 2016 server)

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 74

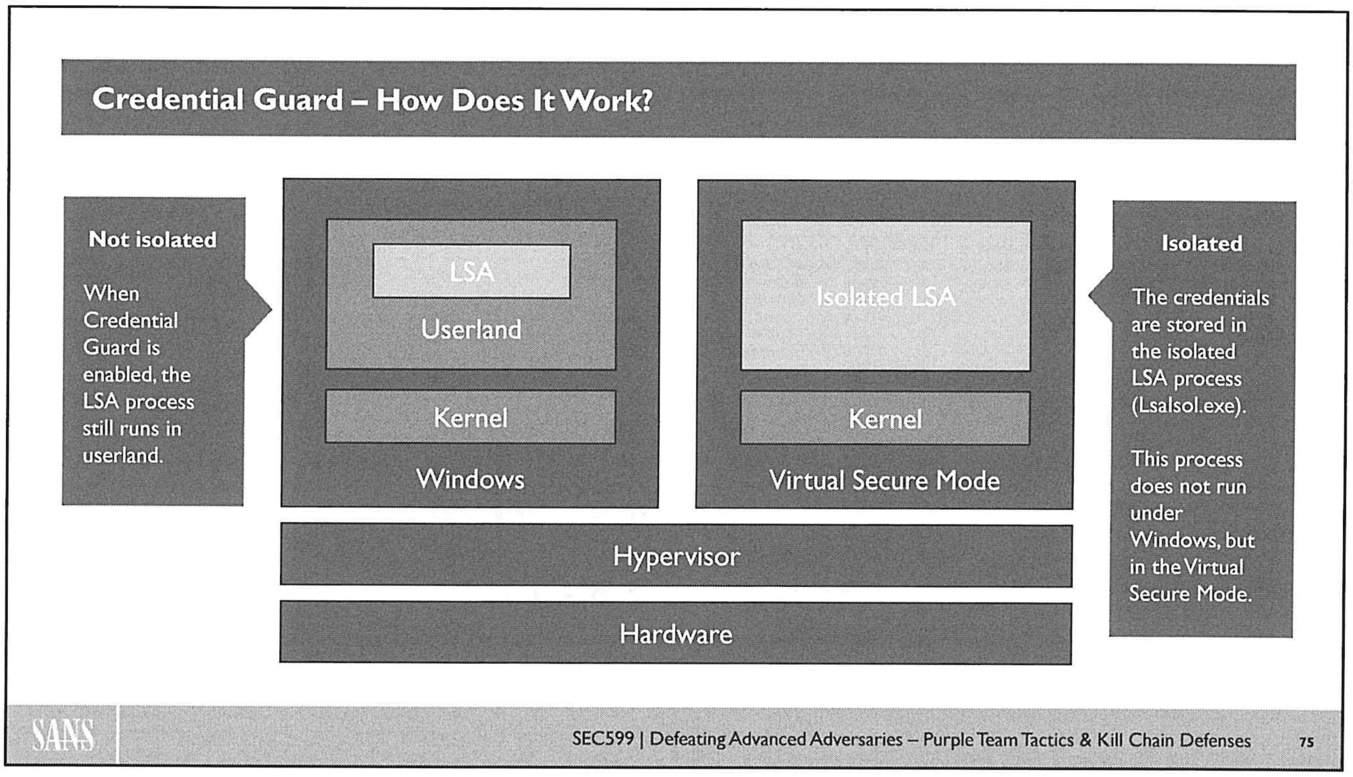
So, What Is Credential Guard?
As we already discussed a few times, the lsass.exe process stores password hashes in memory, amongst others to support SSO (Single Sign On) to remote servers. On older Windows versions (Windows 7), the lsass.exe process even included cleartext (!) passwords (often due to CredSSP or Wdigest).

Credential Guard is a control that relies on VBS (Virtualization Based Security). Virtualization-based security uses hardware virtualization features to create and isolate a secure region of memory from the normal operating system. Windows can use this "virtual secure mode" to host a number of security solutions.

When Credential Guard is enabled, the isolated LSASS process holds the credentials (IsoLsa.exe) and runs in the isolated / secure region. The original lsass.exe process is able to communicate with the isolated lsass process using so-called "trustlets". The original lsass.exe process only holds an encrypted blob of the credentials.

A highly effective control, Credential Guard does come with a number of prerequisites:
- A 64-bit system.
- Trusted Platform Module 1.2 or 2.0 .
- UEFI Lock with secure boot (no legacy boot).
- Windows 10 Enterprise (or Windows 2016 server).

Due to these prerequisites, the full roll-out of Credential Guard in enterprise organizations is expected to be slow!



Credential Guard – How Does It Work?

Not isolated

When Credential Guard is enabled, the LSA process still runs in userland.

Isolated

The credentials are stored in the isolated LSA process (LsaIsol.exe).

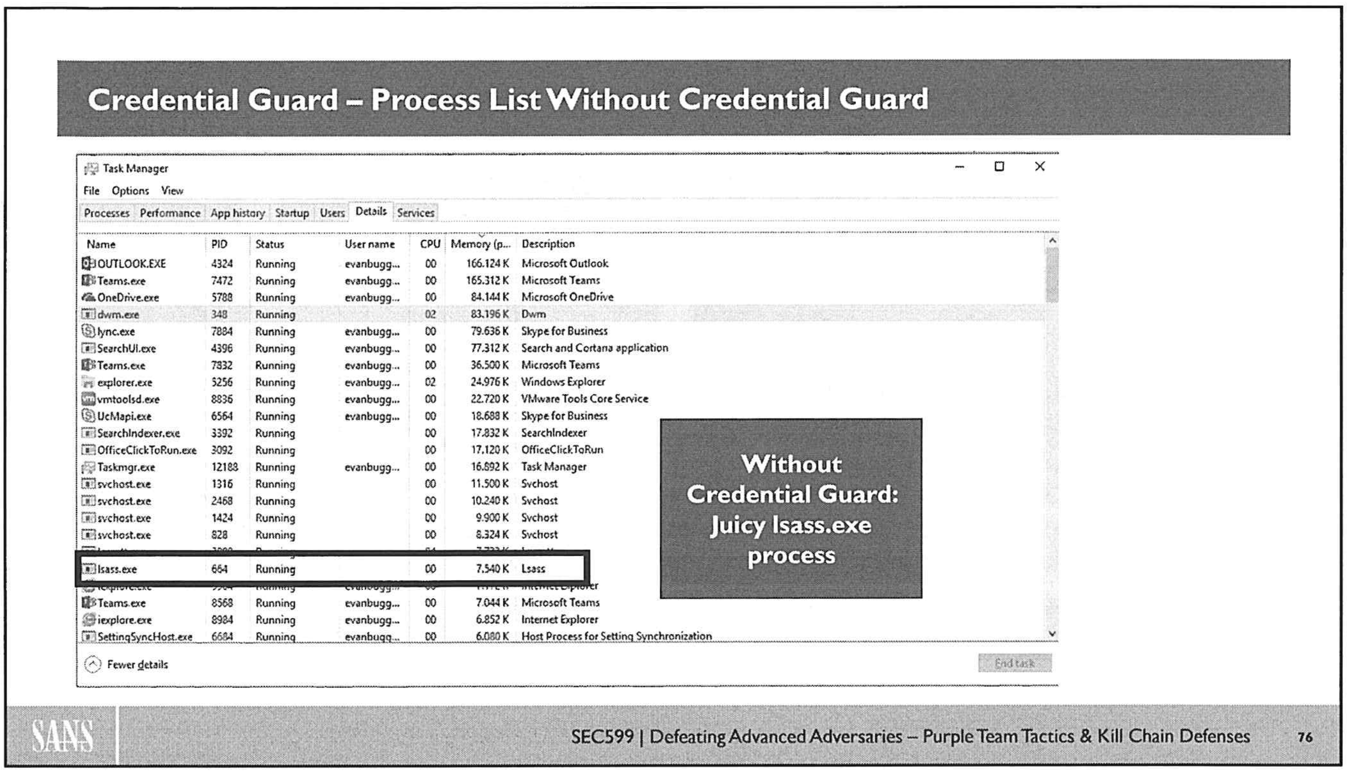
This process does not run under Windows, but in the Virtual Secure Mode.

Credential Guard – How Does It Work?
Credential Guard was introduced with Windows Server 2016 and Windows 10, enterprise editions. It requires modern CPUs that provide virtualization functionality.

When Credential Guard is enabled, Windows still runs on top of the hypervisor and the hardware, and the LSA process still runs in userland. The difference, however, is that the credentials are no longer stored inside this LSA process (lsass.exe).

With Credential Guard, the credentials are stored in the Isolated LSA process (LsaIso.exe). This process does not run under Windows but in the Virtual Secure Mode. This is a separate, virtualized environment that is separated from the other environments (like Windows) via hardware.

It is impossible for processes in the Windows environment to access processes in the Virtual Secure Mode environment, even by manipulating kernel data structures. All operations that require credentials, like checking NTML hashes, is done by the Isolated LSA upon request of the LSA. The credentials never leave the Isolated LSA.



Credential Guard – Process List Without Credential Guard

Task Manager

File Options View

Processes | Performance | App history | Startup | Users | Details | Services

Name	PID	Status	User name	CPU	Memory (p...	Description
OUTLOOK.EXE	4324	Running	evanbugg...	00	166.124 K	Microsoft Outlook
Teams.exe	7472	Running	evanbugg...	00	165.312 K	Microsoft Teams
OneDrive.exe	5788	Running	evanbugg...	00	84.144 K	Microsoft OneDrive
dwm.exe	348	Running		02	83.196 K	Dwm
lync.exe	7884	Running	evanbugg...	00	79.636 K	Skype for Business
SearchUI.exe	4396	Running	evanbugg...	00	77.312 K	Search and Cortana application
Teams.exe	7832	Running	evanbugg...	00	36.500 K	Microsoft Teams
explorer.exe	5256	Running	evanbugg...	02	24.976 K	Windows Explorer
vmtoolsd.exe	8836	Running	evanbugg...	00	22.720 K	VMware Tools Core Service
UcMapi.exe	6564	Running	evanbugg...	00	18.688 K	Skype for Business
SearchIndexer.exe	3392	Running		00	17.832 K	SearchIndexer
OfficeClickToRun.exe	3092	Running		00	17.120 K	OfficeClickToRun
Taskmgr.exe	12188	Running	evanbugg...	00	16.892 K	Task Manager
svchost.exe	1316	Running		00	11.500 K	Svchost
svchost.exe	2468	Running		00	10.240 K	Svchost
svchost.exe	1424	Running		00	9.900 K	Svchost
svchost.exe	828	Running		00	8.324 K	Svchost
[illegible]	[illegible]	[illegible]		[illegible]	[illegible]	[illegible]
lsass.exe	664	Running		00	7.540 K	Lsass
iexplore.exe	[illegible]	Running	evanbugg...	[illegible]	[illegible]	Internet Explorer
Teams.exe	8568	Running	evanbugg...	00	7.044 K	Microsoft Teams
iexplore.exe	8984	Running	evanbugg...	00	6.852 K	Internet Explorer
SettingSyncHost.exe	6684	Running	evanbugg...	00	6.080 K	Host Process for Setting Synchronization

Fewer details

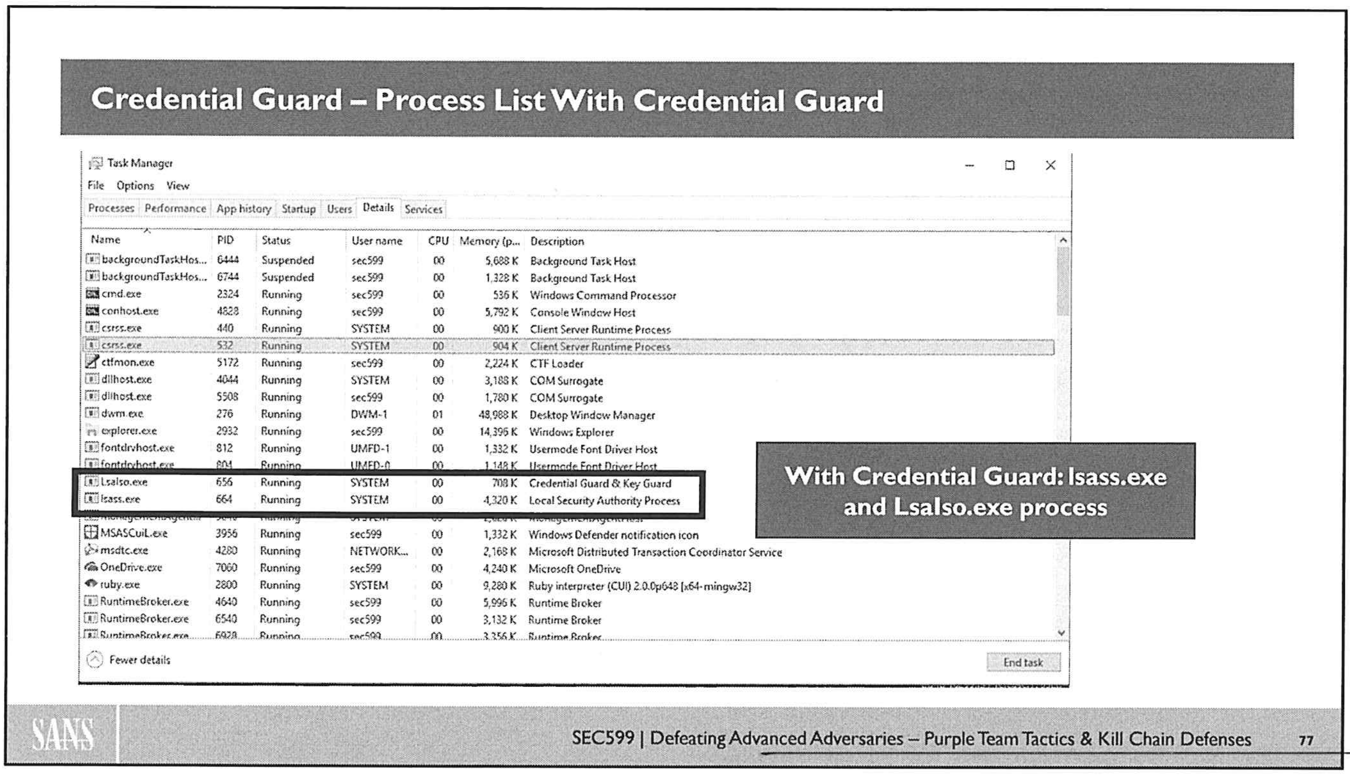
End task

Without Credential Guard: Juicy lsass.exe process

Credential Guard – Process List Without Credential Guard

The above screenshot shows a juicy lsass.exe process, which holds credentials in memory and is thus a feasible target for Mimikatz!

© 2019 Erik Van Buggenhout & Stephen Sims



Credential Guard – Process List With Credential Guard

Task Manager

File Options View

Processes | Performance | App history | Startup | Users | Details | Services

Name	PID	Status	User name	CPU	Memory (p...	Description
backgroundTaskHos...	6444	Suspended	sec599	00	5,688 K	Background Task Host
backgroundTaskHos...	6744	Suspended	sec599	00	1,328 K	Background Task Host
cmd.exe	2324	Running	sec599	00	536 K	Windows Command Processor
conhost.exe	4828	Running	sec599	00	5,792 K	Console Window Host
csrss.exe	440	Running	SYSTEM	00	900 K	Client Server Runtime Process
csrss.exe	532	Running	SYSTEM	00	904 K	Client Server Runtime Process
ctfmon.exe	5172	Running	sec599	00	2,224 K	CTF Loader
dllhost.exe	4044	Running	SYSTEM	00	3,188 K	COM Surrogate
dllhost.exe	5508	Running	sec599	00	1,780 K	COM Surrogate
dwm.exe	276	Running	DWM-1	01	48,988 K	Desktop Window Manager
explorer.exe	2932	Running	sec599	00	14,396 K	Windows Explorer
fontdrvhost.exe	812	Running	UMFD-1	00	1,332 K	Usermode Font Driver Host
fontdrvhost.exe	804	Running	UMFD-0	00	1,148 K	Usermode Font Driver Host
LsaIso.exe	656	Running	SYSTEM	00	708 K	Credential Guard & Key Guard
lsass.exe	664	Running	SYSTEM	00	4,320 K	Local Security Authority Process
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
MSASCuiL.exe	3956	Running	sec599	00	1,332 K	Windows Defender notification icon
msdtc.exe	4280	Running	NETWORK...	00	2,168 K	Microsoft Distributed Transaction Coordinator Service
OneDrive.exe	7060	Running	sec599	00	4,240 K	Microsoft OneDrive
ruby.exe	2800	Running	SYSTEM	00	9,280 K	Ruby interpreter (CUI) 2.0.0p648 [x64-mingw32]
RuntimeBroker.exe	4640	Running	sec599	00	5,996 K	Runtime Broker
RuntimeBroker.exe	6540	Running	sec599	00	3,132 K	Runtime Broker
RuntimeBroker.exe	6928	Running	sec599	00	3,356 K	Runtime Broker

Fewer details

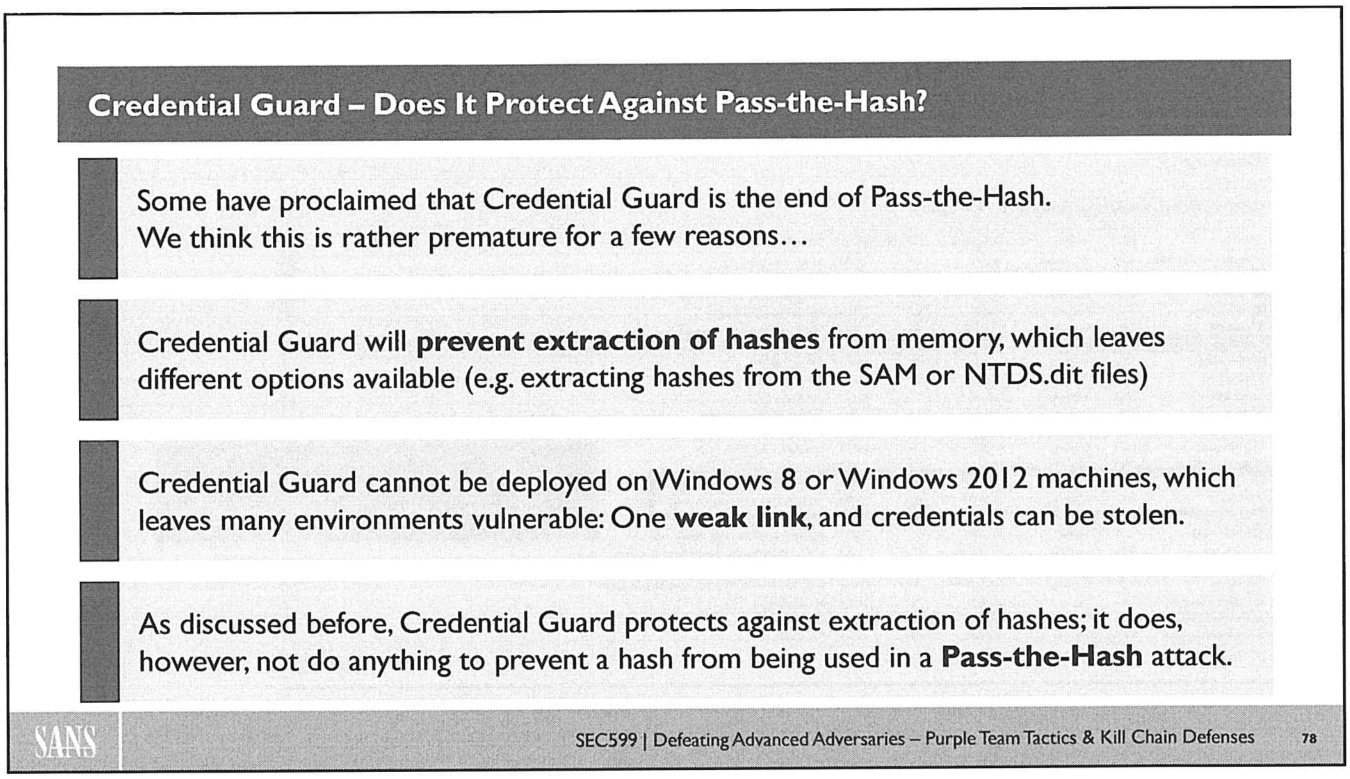
End task

With Credential Guard: lsass.exe and LsaIso.exe process

SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 77

Credential Guard – Process List with Credential Guard

The above screenshot shows the original lsass.exe process and the LsaIso.exe process. In the process description, we can see Credential Guard being mentioned!



Credential Guard – Does It Protect Against Pass-the-Hash?

Some have proclaimed that Credential Guard is the end of Pass-the-Hash.
We think this is rather premature for a few reasons…

Credential Guard will **prevent extraction of hashes** from memory, which leaves different options available (e.g. extracting hashes from the SAM or NTDS.dit files)

Credential Guard cannot be deployed on Windows 8 or Windows 2012 machines, which leaves many environments vulnerable: One **weak link**, and credentials can be stolen.

As discussed before, Credential Guard protects against extraction of hashes; it does, however, not do anything to prevent a hash from being used in a **Pass-the-Hash** attack.

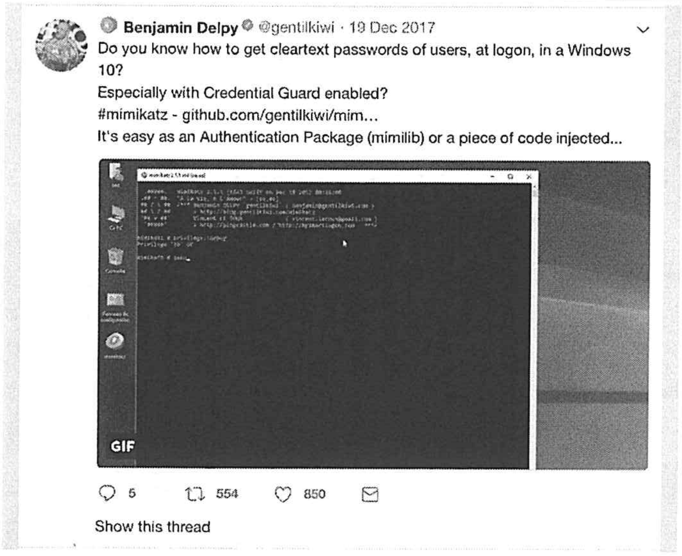
SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 78

Credential Guard – Does It Protect Against Pass-the-Hash?
Credential guard: The end of pass the hash? Some have proclaimed that Credential Guard is the end of Pass-the-Hash. We think this is rather premature for a few reasons:

- First of all, Credential Guard will prevent extraction of hashes from memory, which leaves different options available (e.g. extracting hashes from the SAM or NTDS.dit files).
- Secondly, Credential Guard is only available for Windows 10 and Windows Server 2016, which leaves a potentially large group of vulnerable systems in your corporate environment. One Windows 8 or Windows Server 2012 machine and credentials can still be stolen from that machine.
- Credential Guard does not prevent using a hash for Pass-the-Hash attacks.
- As discussed before, Credential Guard protects against extraction of hashes; it does, however, not do anything to prevent a hash from being used in a Pass-the-Hash attack.

© 2019 Erik Van Buggenhout & Stephen Sims

Credential Guard – Another Interesting Attack Strategy!


Benjamin Delpy @gentilkiwi · 19 Dec 2017
Do you know how to get cleartext passwords of users, at logon, in a Windows 10?
Especially with Credential Guard enabled?
#mimikatz - github.com/gentilkiwi/mim...
It's easy as an Authentication Package (mimilib) or a piece of code injected...

GIF

5 554 850

Show this thread

Highly effective attack strategy developed by @gentilkiwi

Doesn't dump credentials from LSASS memory, requires live authentication

Injects an authentication package that intercepts credentials on logon!

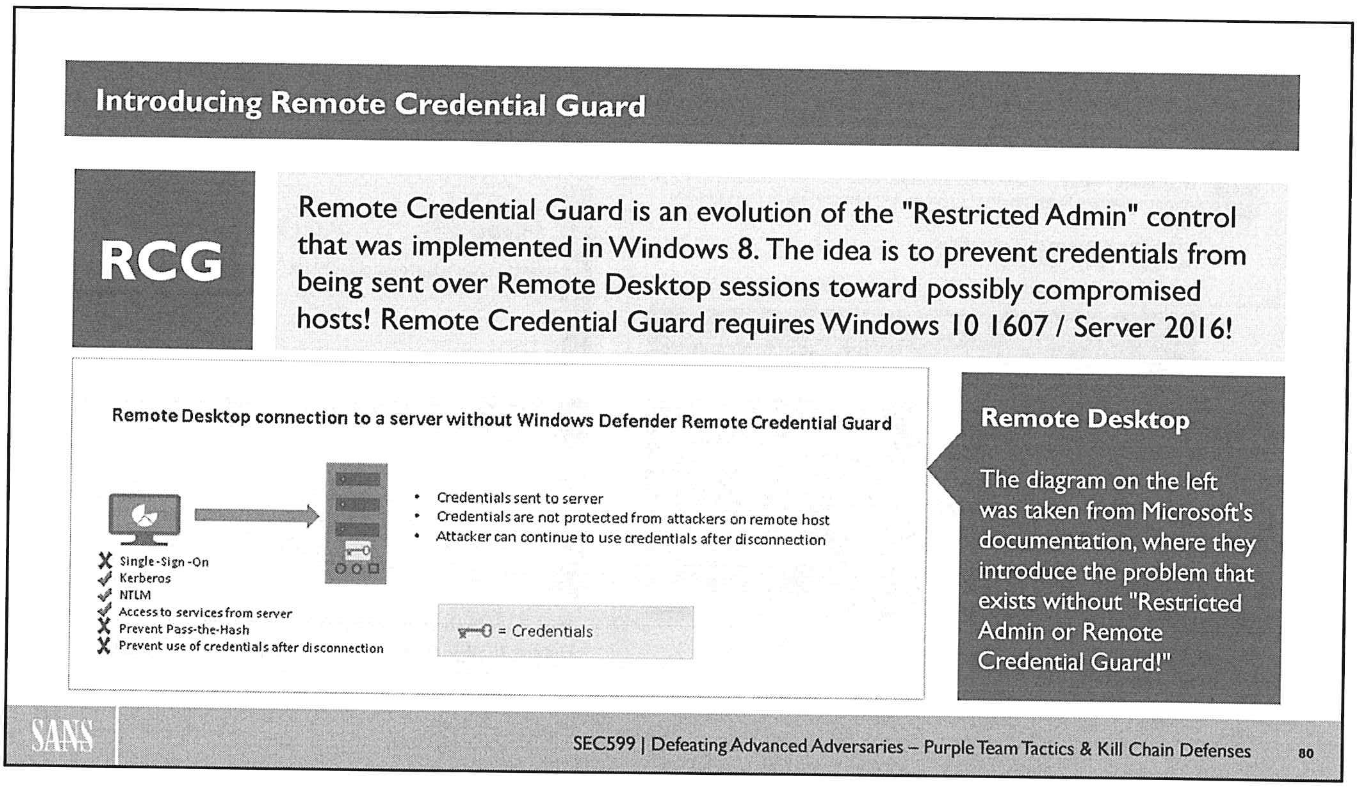
For a detailed write-up, check out https://blog.nviso.be/2018/01/09/windows-credential-guard-mimikatz

Credential Guard – Another Interesting Attack Strategy!
Benjamin Delpy (the author of Mimikatz) sent out an interesting tweet in December 2017, where he teased followers on how he could steal cleartext passwords of users in Windows 10 on systems with Credential Guard enabled.

As an initial reflex, one might think this means he has found a way to defeat Credential Guard. His clever attack strategy, however, was already present in Mimikatz for several months and was not a "new" attack. Furthermore, he does not defeat Credential Guard, as he just leverages another attack strategy to get access to credentials: He injects an authentication package in the Windows memory that will capture credentials when they are entered by the user at logon (think of it as a keylogger).

An interesting write-up about the attack can be found at https://blog.nviso.be/2018/01/09/windows-credential-guard-mimikatz.

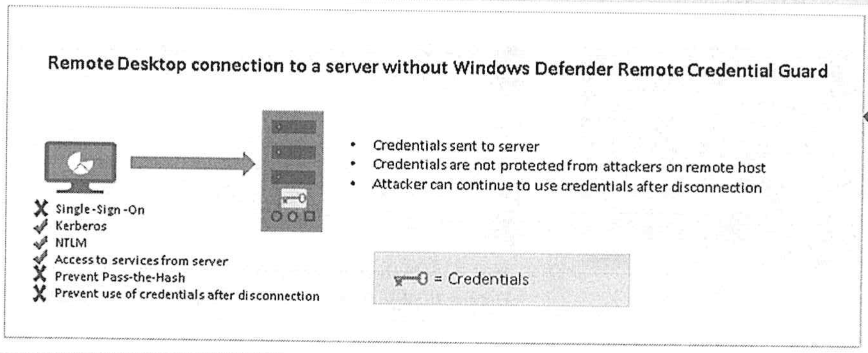
© 2019 Erik Van Buggenhout & Stephen Sims



Introducing Remote Credential Guard

RCG

Remote Credential Guard is an evolution of the "Restricted Admin" control that was implemented in Windows 8. The idea is to prevent credentials from being sent over Remote Desktop sessions toward possibly compromised hosts! Remote Credential Guard requires Windows 10 1607 / Server 2016!


Remote Desktop connection to a server without Windows Defender Remote Credential Guard
✗ Single-Sign-On
✓ Kerberos
✓ NTLM
✓ Access to services from server
✗ Prevent Pass-the-Hash
✗ Prevent use of credentials after disconnection
• Credentials sent to server
• Credentials are not protected from attackers on remote host
• Attacker can continue to use credentials after disconnection
= Credentials

Remote Desktop

The diagram on the left was taken from Microsoft's documentation, where they introduce the problem that exists without "Restricted Admin or Remote Credential Guard!"

SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 80

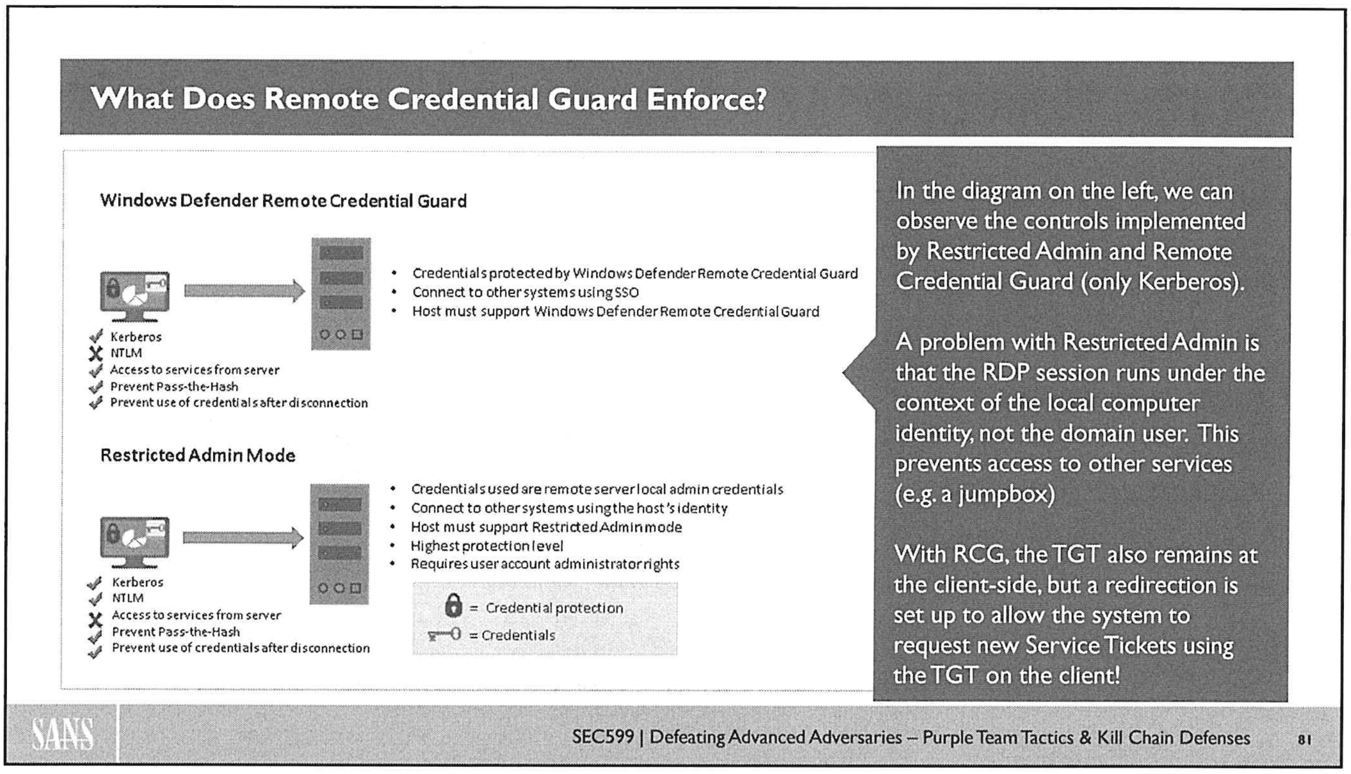
Introducing Remote Credential Guard

Remote Credential Guard is an evolution of the "Restricted Admin" control that was implemented in Windows 8. The idea is to prevent credentials from being sent over Remote Desktop sessions toward possibly compromised hosts! Remote Credential Guard requires Windows 10 1607 / Server 2016!

So, what is the problem? In the diagram on the slide (which was taken from Microsoft's documentation), we can observe the issues and risks associated with normal Remote Desktop connectivity: When an administrator connects to a device, his credentials are sent to the server, where they are possibly at risk of being stolen by an adversary!

Restricted Admin was a prior security control proposed by Microsoft, but it never really "took off" as it had some inherent weaknesses (it prevented SSO access and could thus, for example, not be implemented on jump boxes). More information can be found on Microsoft documentation here: https://docs.microsoft.com/en-us/windows/security/identity-protection/remote-credential-guard.

© 2019 Erik Van Buggenhout & Stephen Sims



What Does Remote Credential Guard Enforce?

Windows Defender Remote Credential Guard

- Credentials protected by Windows Defender Remote Credential Guard
- Connect to other systems using SSO
- Host must support Windows Defender Remote Credential Guard

✓ Kerberos
✗ NTLM
✓ Access to services from server
✓ Prevent Pass-the-Hash
✓ Prevent use of credentials after disconnection

Restricted Admin Mode

- Credentials used are remote server local admin credentials
- Connect to other systems using the host's identity
- Host must support Restricted Admin mode
- Highest protection level
- Requires user account administrator rights

✓ Kerberos
✓ NTLM
✗ Access to services from server
✓ Prevent Pass-the-Hash
✓ Prevent use of credentials after disconnection

🔒 = Credential protection
🔑 = Credentials

In the diagram on the left, we can observe the controls implemented by Restricted Admin and Remote Credential Guard (only Kerberos).

A problem with Restricted Admin is that the RDP session runs under the context of the local computer identity, not the domain user. This prevents access to other services (e.g. a jumpbox)

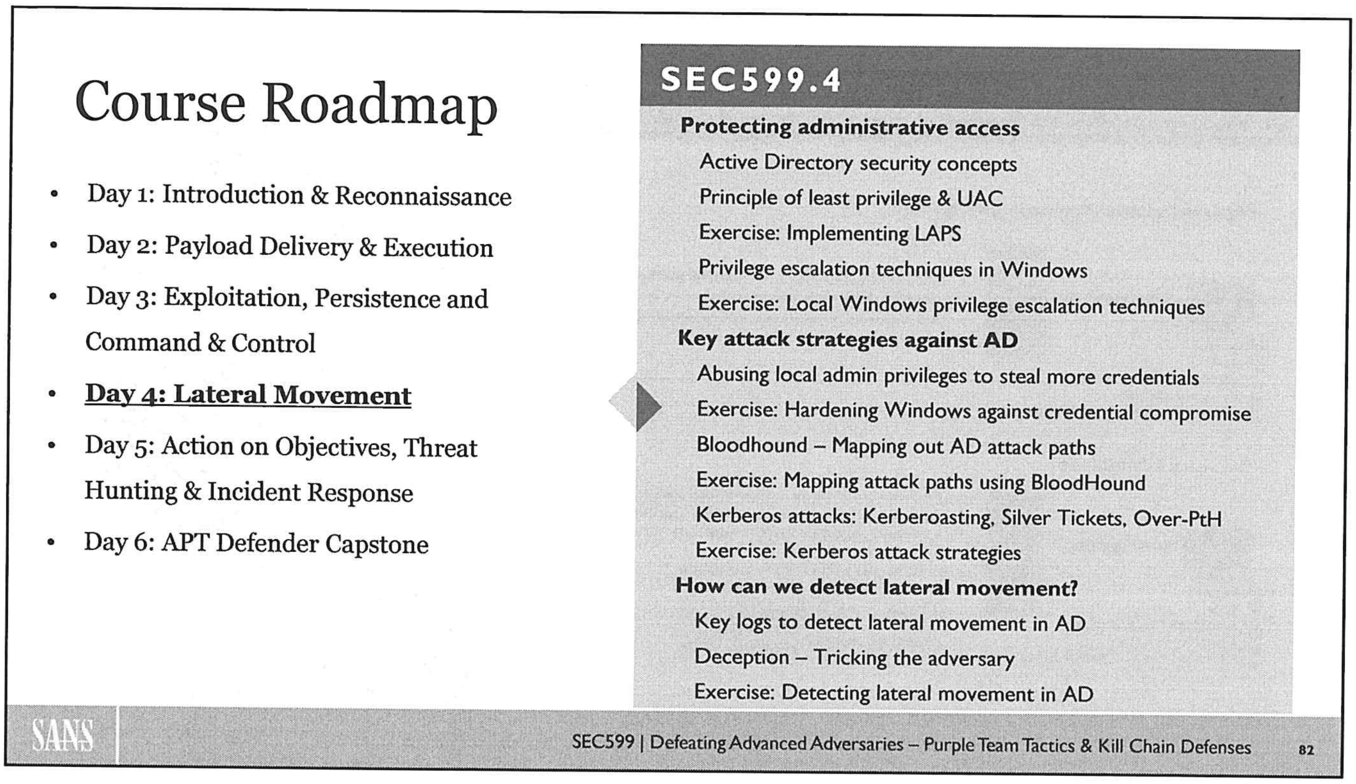
With RCG, the TGT also remains at the client-side, but a redirection is set up to allow the system to request new Service Tickets using the TGT on the client!

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 81

What Does Remote Credential Guard Enforce?
In the diagram on the left, we can observe the controls implemented by Restricted Admin and Remote Credential Guard (only Kerberos).

A problem with Restricted Admin is that the RDP session runs under the context of the local computer identity, not the domain user. This prevents access to other services (e.g. a jumpbox). With RCG, the TGT also remains at the client-side, but a redirection is set up to allow the system to request new Service Tickets using the TGT on the client! Service Tickets are, however, available on the server that is connected to. Worst-case, an adversary could steal Service Tickets (which only have a limited lifetime, though, and only allow access to a specific service).

It's worthwhile mentioning that Microsoft still actively recommends using "Restricted Admin" for the scenario where helpdesk users connect to workstations to perform local support operations. The main reason here being that in this case, the limited time-window available to an adversary is also not present, as the remote system only uses local admin credentials, which would be already compromised in an attack scenario.



Course Roadmap

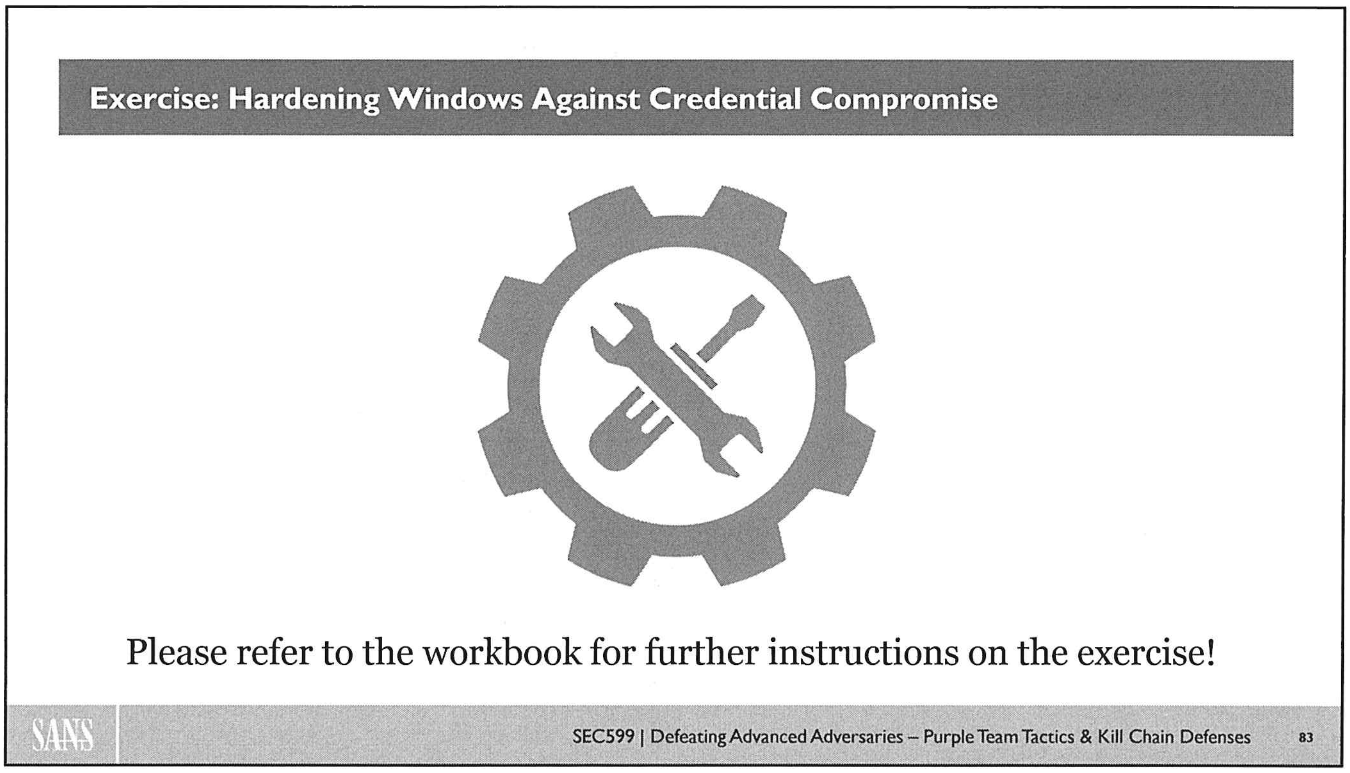
- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- **Day 4: Lateral Movement**
- Day 5: Action on Objectives, Threat Hunting & Incident Response
- Day 6: APT Defender Capstone

SEC599.4

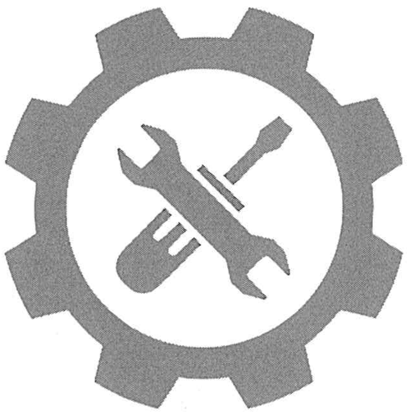
SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 82

This page intentionally left blank.

© 2019 Erik Van Buggenhout & Stephen Sims



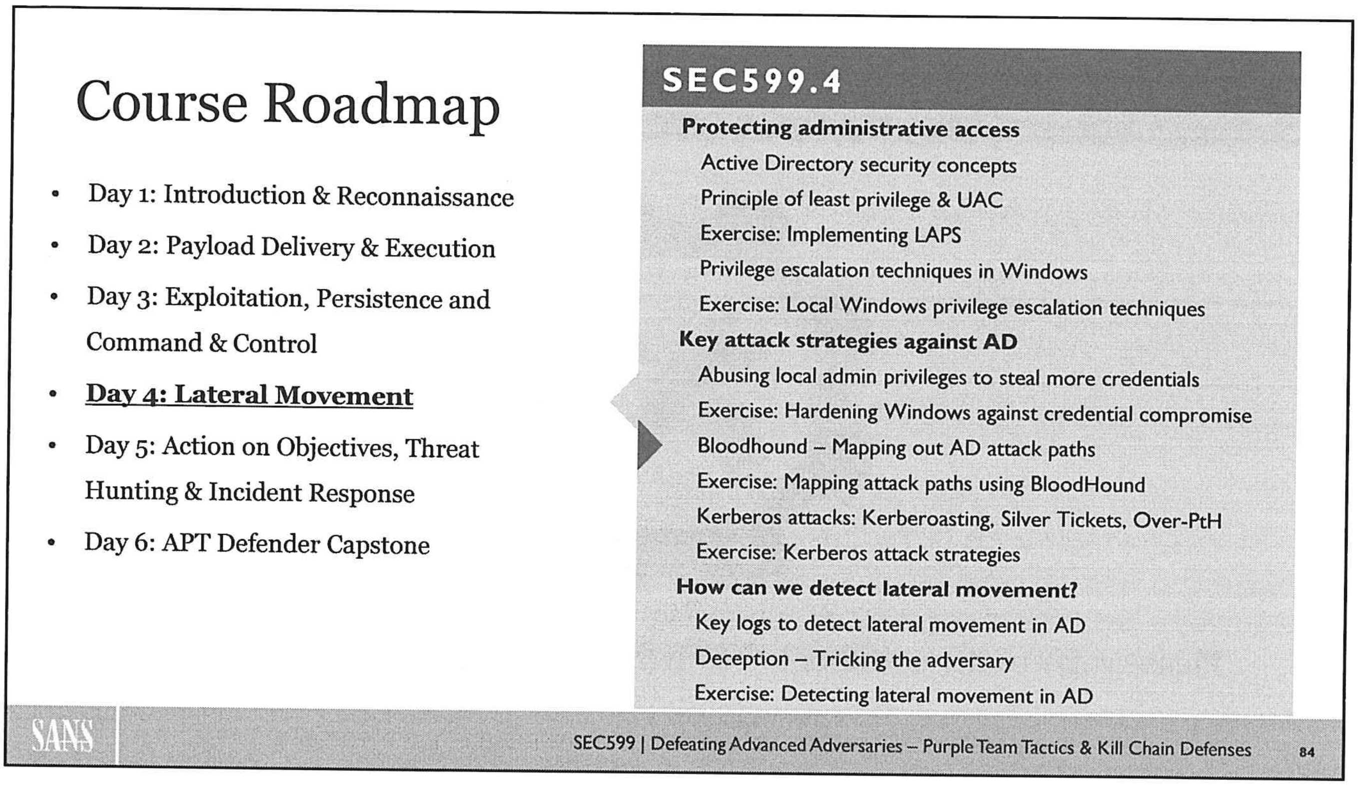
Exercise: Hardening Windows Against Credential Compromise



Please refer to the workbook for further instructions on the exercise!

This page intentionally left blank.

© 2019 Erik Van Buggenhout & Stephen Sims



Course Roadmap

- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- **Day 4: Lateral Movement**
- Day 5: Action on Objectives, Threat Hunting & Incident Response
- Day 6: APT Defender Capstone

SEC599.4

Protecting administrative access
- Active Directory security concepts
- Principle of least privilege & UAC
- Exercise: Implementing LAPS
- Privilege escalation techniques in Windows
- Exercise: Local Windows privilege escalation techniques

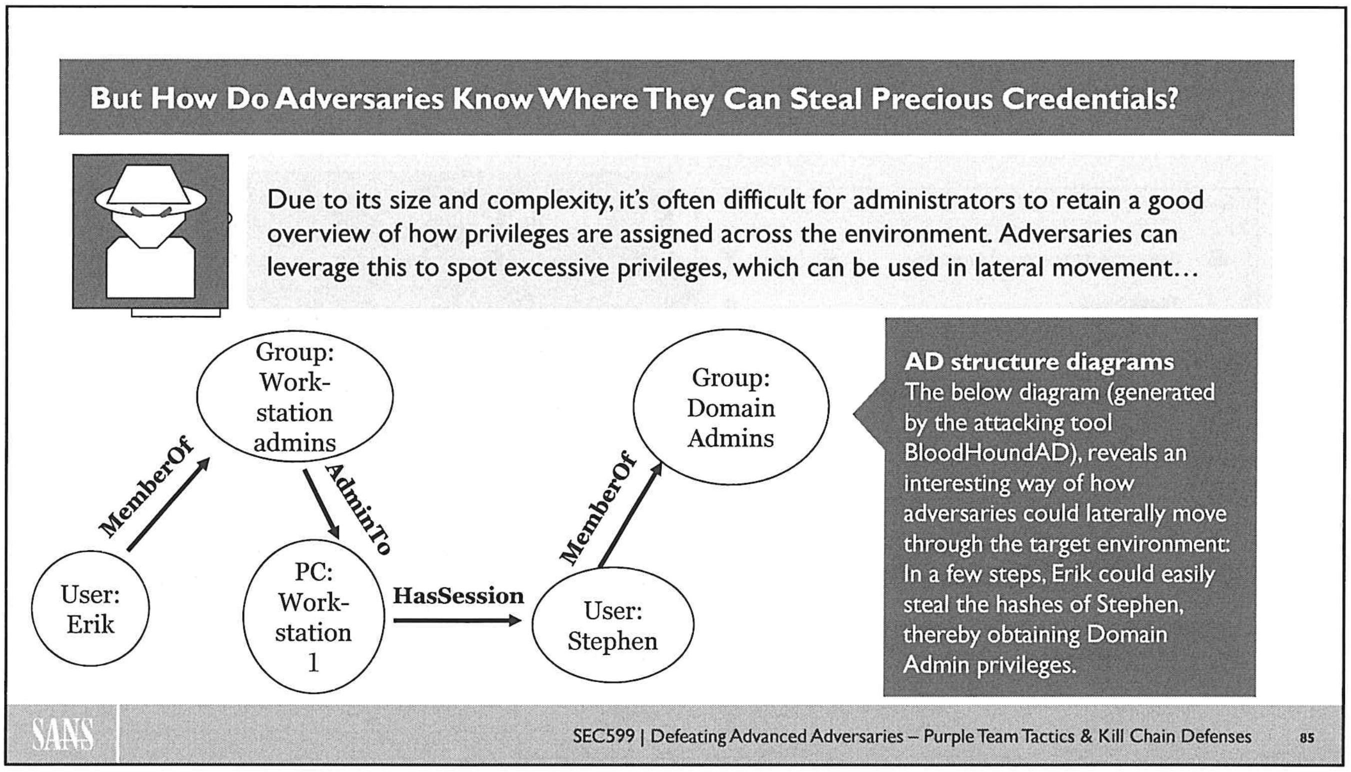
Key attack strategies against AD
- Abusing local admin privileges to steal more credentials
- Exercise: Hardening Windows against credential compromise
- Bloodhound – Mapping out AD attack paths
- Exercise: Mapping attack paths using BloodHound
- Kerberos attacks: Kerberoasting, Silver Tickets, Over-PtH
- Exercise: Kerberos attack strategies

How can we detect lateral movement?
- Key logs to detect lateral movement in AD
- Deception – Tricking the adversary
- Exercise: Detecting lateral movement in AD

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 84

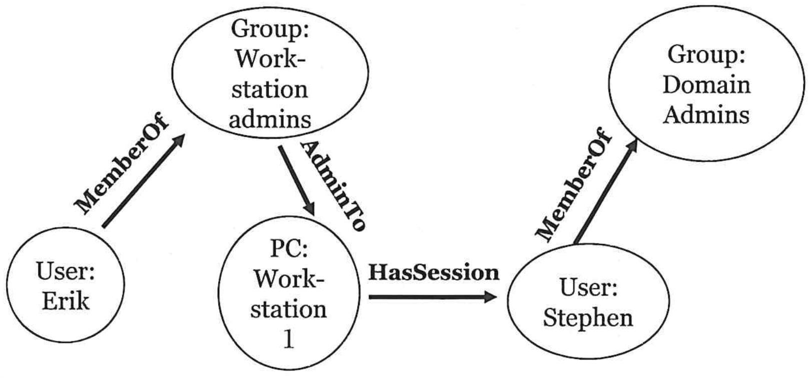
This page intentionally left blank.

© 2019 Erik Van Buggenhout & Stephen Sims



But How Do Adversaries Know Where They Can Steal Precious Credentials?

Due to its size and complexity, it's often difficult for administrators to retain a good overview of how privileges are assigned across the environment. Adversaries can leverage this to spot excessive privileges, which can be used in lateral movement...



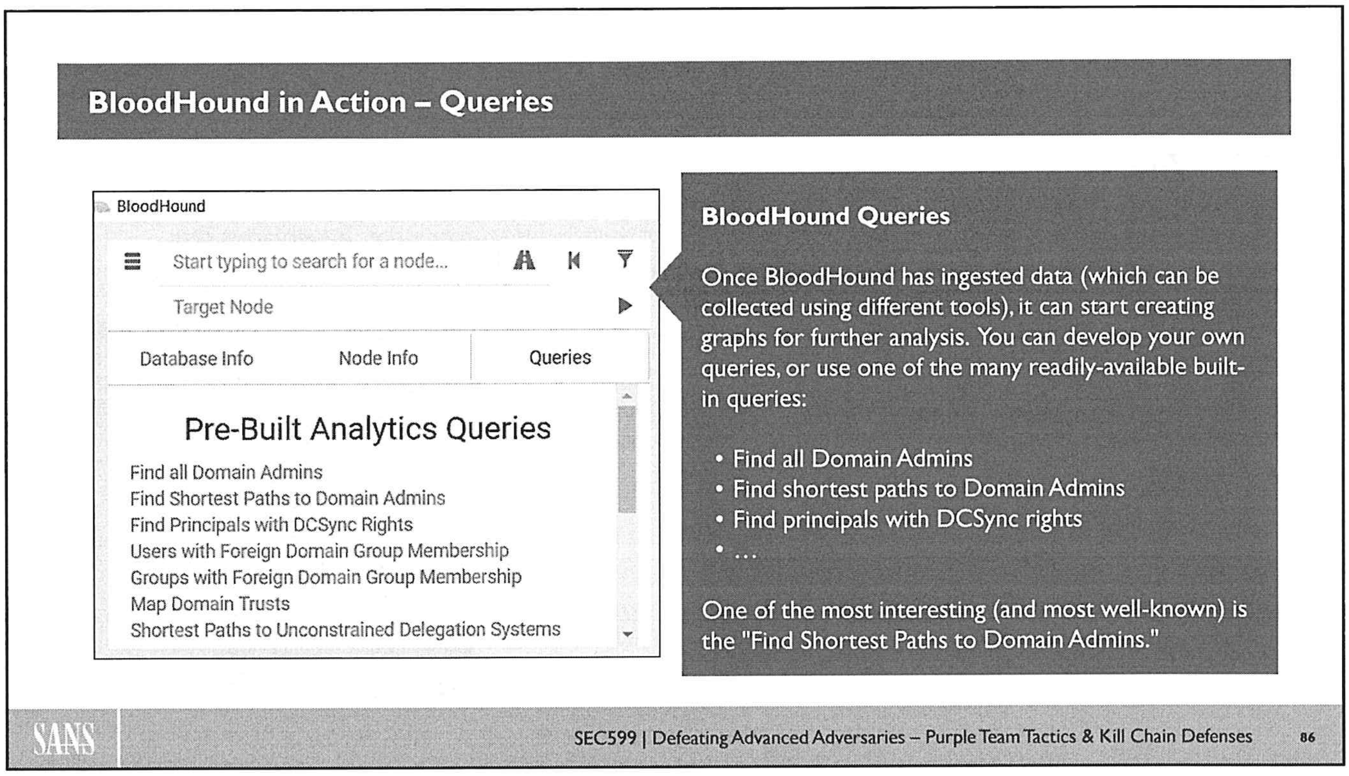
AD structure diagrams
The below diagram (generated by the attacking tool BloodHoundAD), reveals an interesting way of how adversaries could laterally move through the target environment: In a few steps, Erik could easily steal the hashes of Stephen, thereby obtaining Domain Admin privileges.

SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 85

But How Do Adversaries Know Where They Can Steal Precious Credentials?
Due to its size and complexity, it's often difficult for administrators to retain a good overview of how privileges are assigned across the environment. Adversaries can leverage this to spot excessive privileges, which can be used in lateral movement...

Once (limited) administrator privileges are obtained (e.g. on all workstations but not on servers), adversaries can start hopping from one system to the other in an attempt to steal credentials from different hops, thereby escalating privileges as they go along. An example would be a Domain Administrator that is authenticated to one of the workstations under the control of the adversary. The adversary could go to this workstation and dump the credentials from memory using Mimikatz.

A tool that facilitates this attack is BloodHoundAD, which generates a diagram of active sessions and relationships in Active Directory. On the slide above, we can see an example of such a diagram: In a few steps, Erik could easily steal the hashes of Stephen, thereby obtaining Domain Admin privileges.



BloodHound in Action – Queries

BloodHound

Start typing to search for a node...

Target Node

Database Info | Node Info | Queries

Pre-Built Analytics Queries

Find all Domain Admins
Find Shortest Paths to Domain Admins
Find Principals with DCSync Rights
Users with Foreign Domain Group Membership
Groups with Foreign Domain Group Membership
Map Domain Trusts
Shortest Paths to Unconstrained Delegation Systems

BloodHound Queries

Once BloodHound has ingested data (which can be collected using different tools), it can start creating graphs for further analysis. You can develop your own queries, or use one of the many readily-available built-in queries:

- Find all Domain Admins
- Find shortest paths to Domain Admins
- Find principals with DCSync rights
- …

One of the most interesting (and most well-known) is the "Find Shortest Paths to Domain Admins."

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 86

BloodHound in Action – Queries
Once BloodHound has ingested data (which can be collected using different tools), it can start creating graphs for further analysis. BloodHound provides multiple tools for data collection (called ingestors). A good example is Sharphound, which is a C# ingestor, the primary ingestor to run on Windows systems.

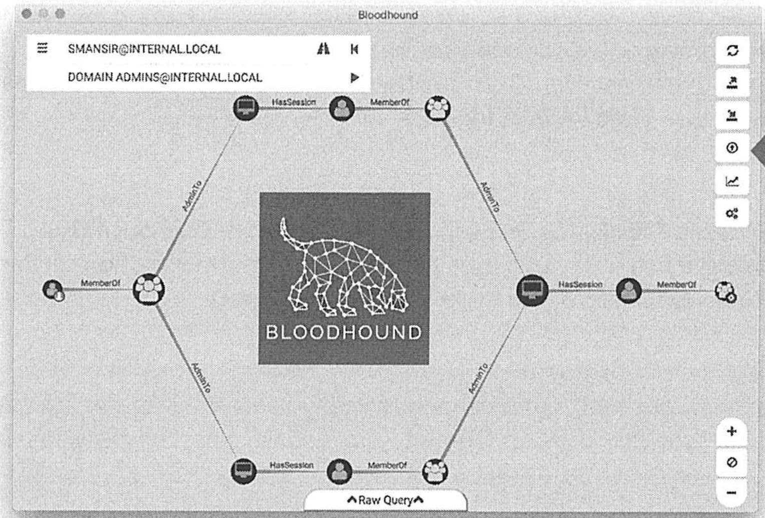
You can develop your own queries or use one of the many readily available built-in queries. Some examples of available queries include:

- Find all Domain Admins.
- Find Shortest Paths to Domain Admins.
- Find Principals with DCSync Rights.
- …

One of the most interesting (and most well-known) is the "Find Shortest Paths to Domain Admins"!

© 2019 Erik Van Buggenhout & Stephen Sims

BloodHound in Action – Graph Interface



BloodHound Graph Interface

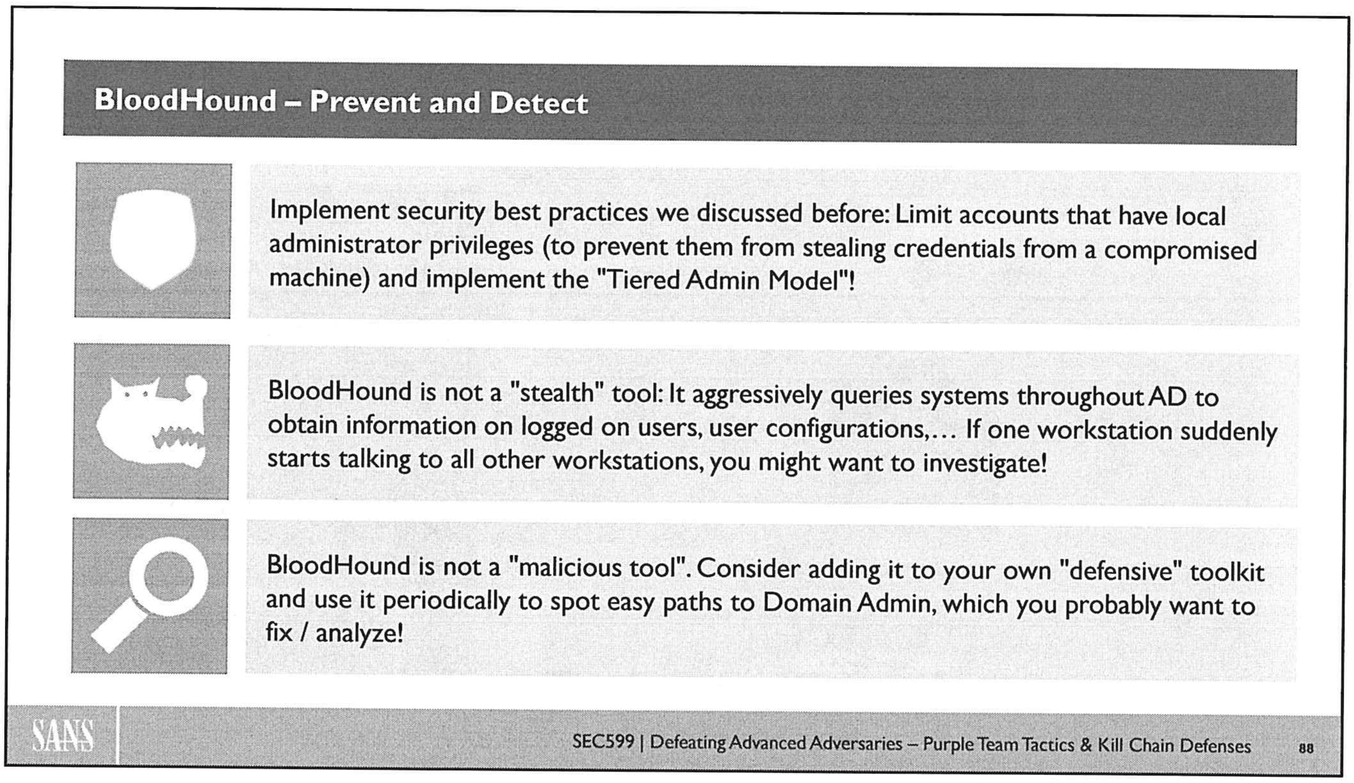
In the graph to the left, we can see the result of one of the queries.

In this specific case, a user attempted to find the shortest path to getting Domain Administrator. It appears there are two available paths, which are clearly described.

A real-life graph of an enterprise environment will likely look more complex!

BloodHound in Action – Graph Interface
In the graph on the slide, we can see the result of one of the queries. BloodHound uses a neo4j database to store all of the information and provides it in a visual web interface for analysis. In this specific case, a user attempted to find the shortest path to getting Domain Administrator. It appears there are two available paths, which can be clearly seen in the diagram.

As a next step, the attacker would now authenticate to one of the first machines in the graph and attempt to steal credentials of the next user in the attack path. Note that in this specific case, there are only 2 computer hops; a real-life graph of an enterprise environment will likely look more complex!



BloodHound – Prevent and Detect

Implement security best practices we discussed before: Limit accounts that have local administrator privileges (to prevent them from stealing credentials from a compromised machine) and implement the "Tiered Admin Model"!

BloodHound is not a "stealth" tool: It aggressively queries systems throughout AD to obtain information on logged on users, user configurations,... If one workstation suddenly starts talking to all other workstations, you might want to investigate!

BloodHound is not a "malicious tool". Consider adding it to your own "defensive" toolkit and use it periodically to spot easy paths to Domain Admin, which you probably want to fix / analyze!

SANS

SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 88

BloodHound – Prevent and Detect
What can we do to "protect" our environments from BloodHound? There's a few items to consider:

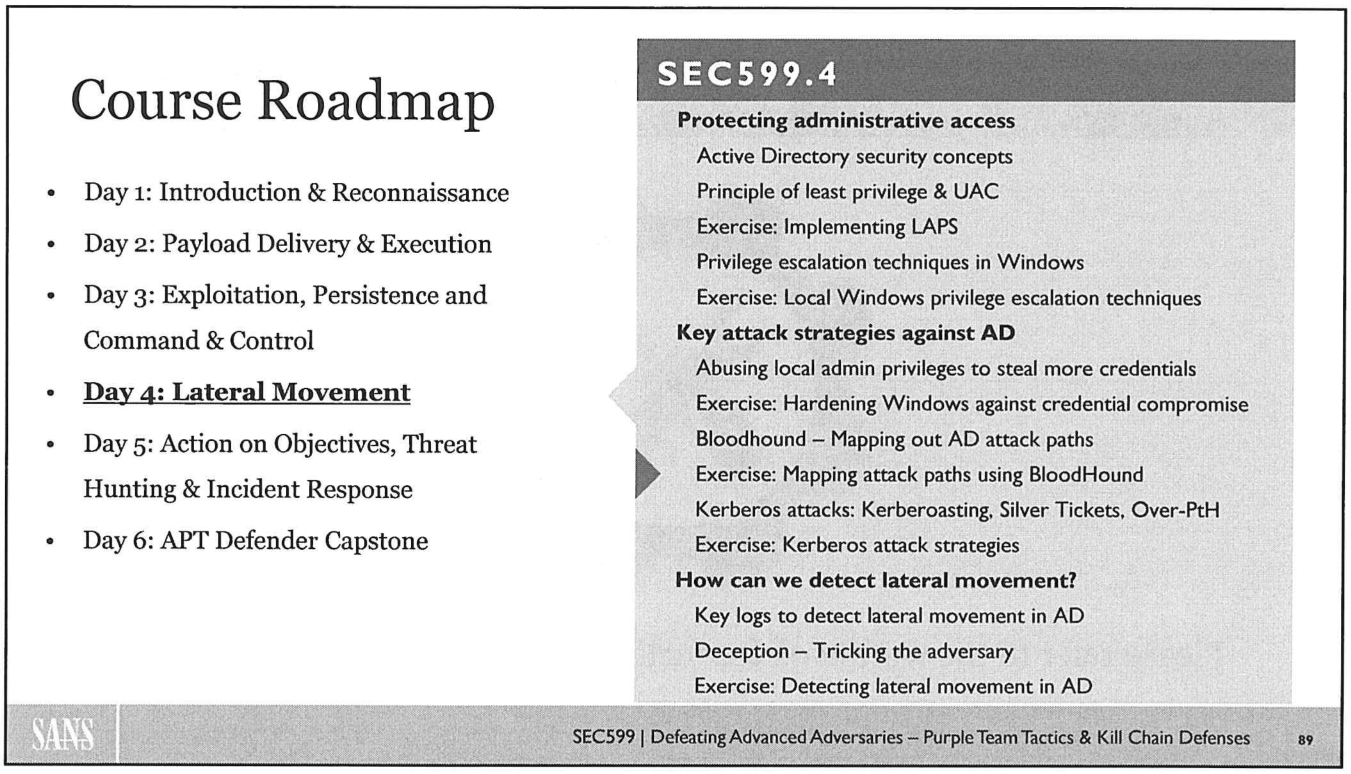
- As a first, fundamental recommendation, we advise implementation of the security best practices we discussed before: Limit accounts that have local administrator privileges (to prevent them from stealing credentials from a compromised machine) and implement the "Tiered Admin Model"! This will severely reduce the number of possible attack paths created by BloodHound.

- It's worth mentioning that BloodHound is not a "stealth" tool: It aggressively queries systems throughout AD to obtain information on logged-on users, user configurations,... If one workstation suddenly starts talking to all other workstations, you might want to investigate! This would, of course, require visibility on internal traffic flows!

- Finally, it's important to note that BloodHound is not a "malicious tool". Consider adding it to your own "defensive" toolkit and use it periodically to spot easy paths to Domain Admin, which you probably want to fix / analyze!

We will discuss deception later in the course. An example of deception is generating "fake" attack paths for BloodHound, thereby making the adversary lose time!

© 2019 Erik Van Buggenhout & Stephen Sims



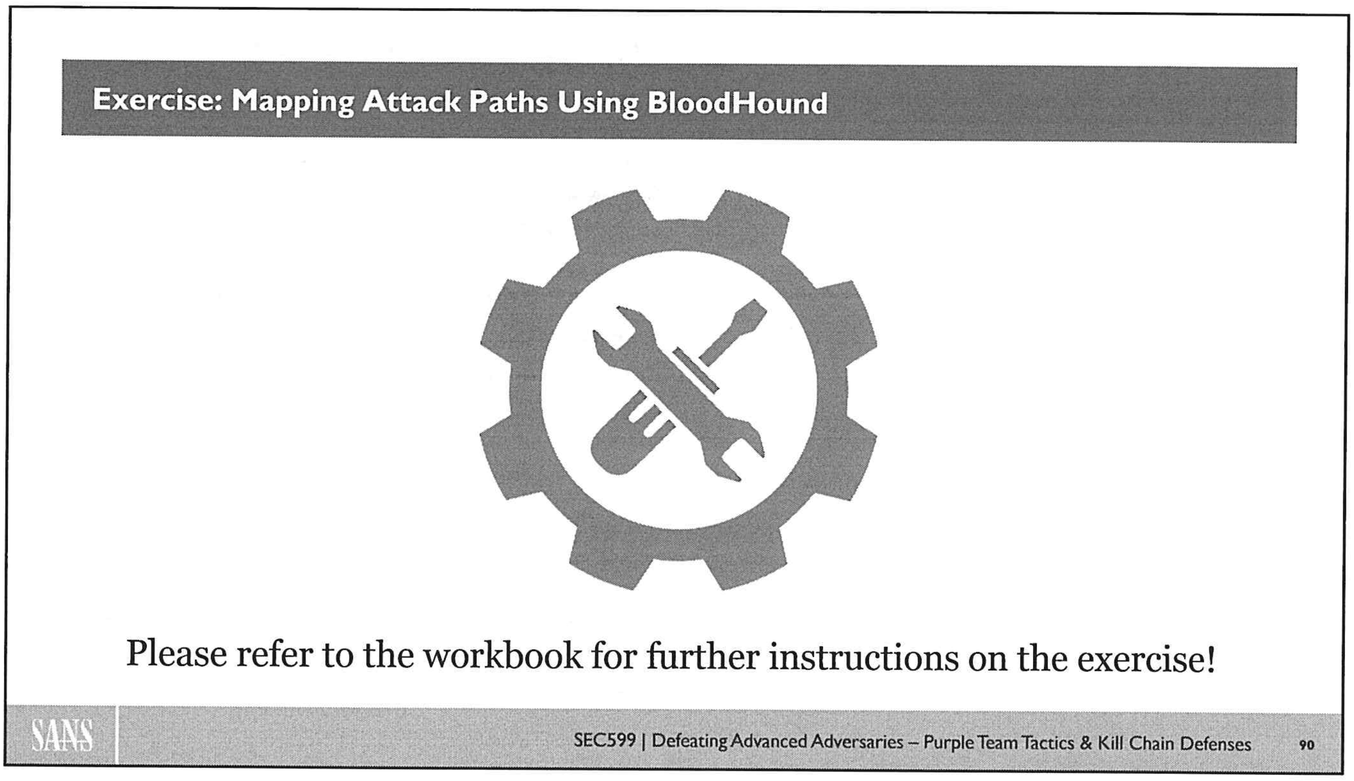
Course Roadmap

- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- **Day 4: Lateral Movement**
- Day 5: Action on Objectives, Threat Hunting & Incident Response
- Day 6: APT Defender Capstone

SEC599.4

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 89

This page intentionally left blank.



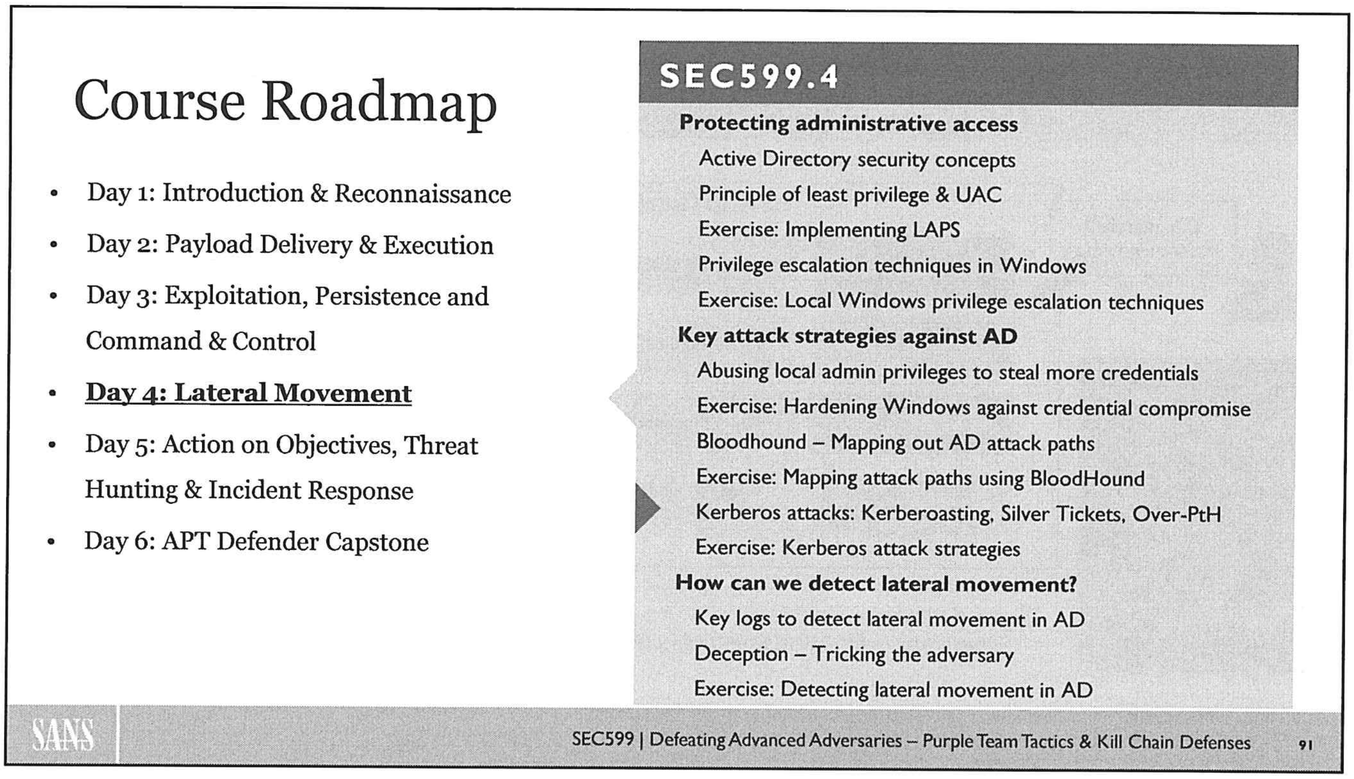
Exercise: Mapping Attack Paths Using BloodHound

Please refer to the workbook for further instructions on the exercise!

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 90

This page intentionally left blank.

© 2019 Erik Van Buggenhout & Stephen Sims

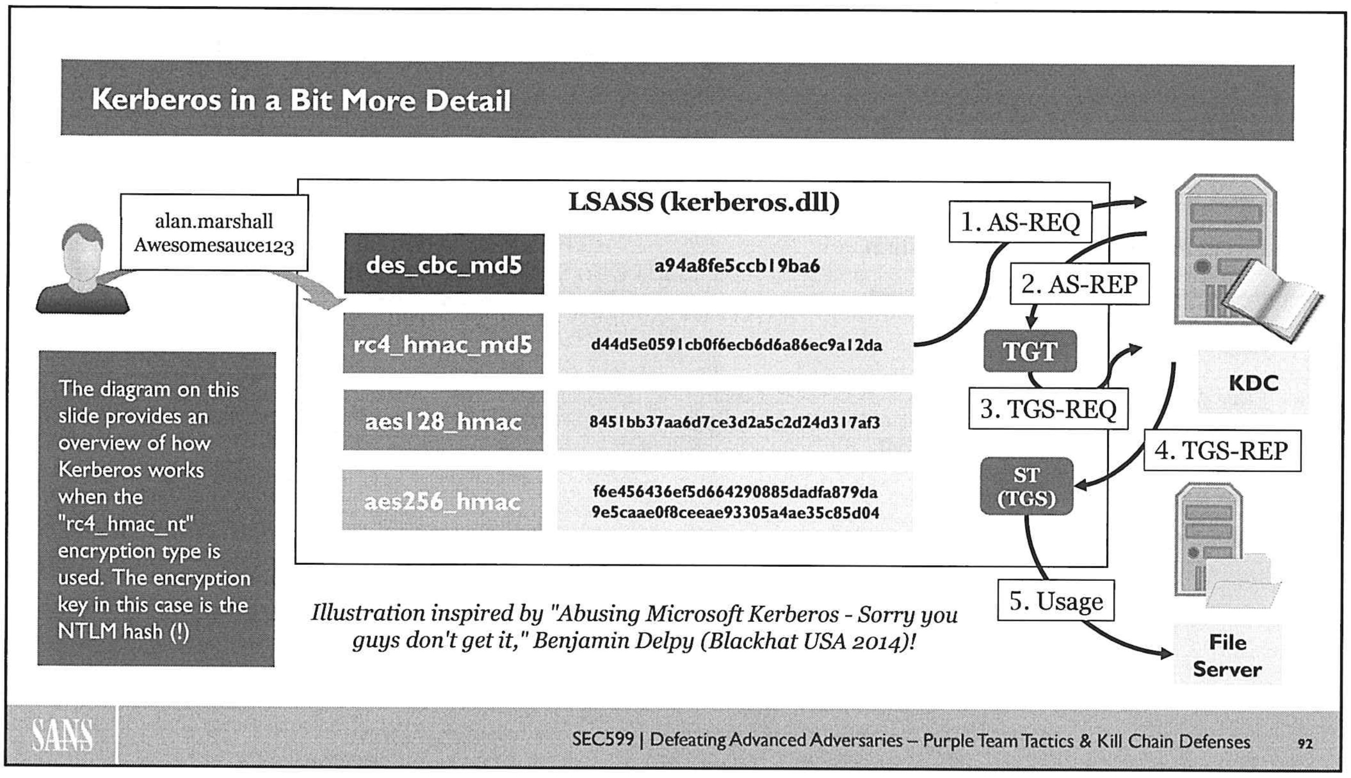


Course Roadmap

- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- **Day 4: Lateral Movement**
- Day 5: Action on Objectives, Threat Hunting & Incident Response
- Day 6: APT Defender Capstone

This page intentionally left blank.

© 2019 Erik Van Buggenhout & Stephen Sims



Kerberos in a Bit More Detail

The diagram on this slide provides an overview of how Kerberos works when the "rc4_hmac_nt" encryption type is used. The encryption key in this case is the NTLM hash (!)

Illustration inspired by "Abusing Microsoft Kerberos - Sorry you guys don't get it," Benjamin Delpy (Blackhat USA 2014)!

Kerberos in a Bit More Detail

We already briefly discussed how Kerberos works, but let's look at it a bit more… The diagram on the slide above illustrates how Kerberos works using the rc4_hmac_md5 encryption type. At first, the user of course provides his / her password, after which a number of keys are stored in LSASS. Tickets are requested in the following way:

1. AS-REQ – As an initial step, the client performs pre-authentication by sending an AS-REQ to the KDC (in this case, the AS component of the KDC). This AS-REQ includes an encrypted version of the timestamp (encrypted using the password hash of the client account), which is validated by the KDC.

2. AS-REP – If authentication succeeds (timestamp can be decrypted), the KDC sends two items to the client:
   - The Ticket Granting Ticket (TGT), which includes the Client / TGS session key and is encrypted using the KDC long-term key (in case of rc4_hmac_md5, the krbtgt NT password hash). This KDC long-term key is also used to sign the PAC (which is also part of the TGT and includes information on who the user is and what groups he is a member of).
   - The Client / TGS session key, encrypted using the client long-term key (in case of rc4_hmac_md5, the user password hash).

3. TGS-REQ – Thirdly, the user wants to authenticate to a certain service and thus sends the following to the KDC (in this case, the TGS component of the KDC):
   - An authenticator message (encrypted using the Client / TGS session key).
   - The encrypted TGT and a ticket request (referencing a certain Service Principle Name, SPN).

4. TGS-REP – The KDC will validate whether the service exists and create a Service Ticket (ST) that is sent back. It's important to note that the KDC does not do any validation of privileges (it leaves that to the target service). The Service Ticket has two parts:
   - A client portion, which is encrypted using the Client / TGS session key (so this can be decrypted by the client).
   - A server portion, which is encrypted using the target long-term key (in case of rc4_hmac_md5, the password hash of the target service). This server portion also includes the PAC of the user. It is this Service Ticket that the user will subsequently submit to the service he or she is trying to access.

© 2019 Erik Van Buggenhout & Stephen Sims

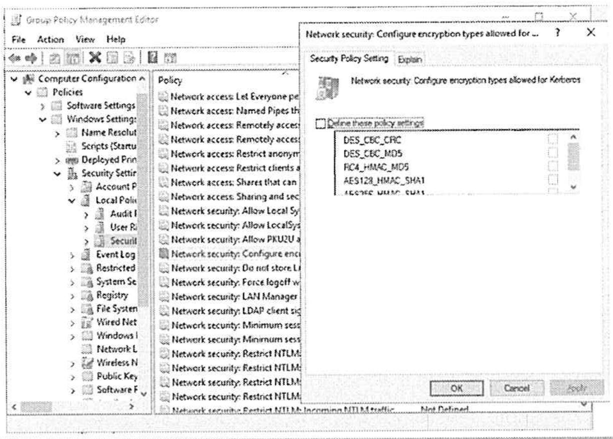
5. Finally, the client can now use the server portion of the ticket to try accessing the desired service (e.g. a file server). The target server will try to decrypt the server portion of the Service Ticket using the target long-term key (in case of rc4_hmac_md5, the password hash of the target service). It will use the PAC information included in the Service Ticket to validate the privileges of the user.

Please note that the illustration used in this slide was based on an amazing presentation by Benjamin Delpy at Blackhat 2014: "Abusing Microsoft Kerberos – Sorry you guys don't get it".

Kerberos Encryption Types

TGT
ST

So what encryption type does Kerberos use? By default, the system will use the highest method of encryption that is supported by the client! For Microsoft-based systems, as of Windows Vista, all Microsoft machines support AES encryption types!



Computer Configuration
Policies
Software Settings
Windows Settings
Name Resolut
Scripts (Startu
Deployed Prin
Security Settir
Account P
Local Polic
Audit I
User Ri
Securit
Event Log
Restricted
System Se
Registry
File Systen
Wired Net
Windows I
Network L
Wireless N
Public Key
Software F

Policy
Network access: Let Everyone pe
Network access: Named Pipes th
Network access: Remotely acces
Network access: Remotely acces
Network access: Restrict anonym
Network access: Restrict clients a
Network access: Shares that can
Network access: Sharing and sec
Network security: Allow Local Sy
Network security: Allow LocalSys
Network security: Allow PKU2U a
Network security: Configure enc
Network security: Do not store L
Network security: Force logoff w
Network security: LAN Manager
Network security: LDAP client sig
Network security: Minimum sess
Network security: Minimum sess
Network security: Restrict NTLM
Network security: Restrict NTLM
Network security: Restrict NTLM
Network security: Restrict NTLM
Network security: Restrict NTLM: Incoming NTLM traffic Not Defined

Network security: Configure encryption types allowed for ...

Network security: Configure encryption types allowed for Kerberos

☐ Define these policy settings

DES_CBC_CRC ☐
DES_CBC_MD5 ☐
RC4_HMAC_MD5 ☐
AES128_HMAC_SHA1 ☐

Policy settings in Windows AD

It's rather straightforward to configure Windows systems to only support specific encryption types for Kerberos. By default, all encryption types are enabled and supported!

It might be worth noting that as of Windows Vista, all Windows systems support the rather secure AES encryption method (which uses 4096 iterations of PBKDF2).

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 94

Kerberos Encryption Types

So what encryption type does Kerberos use? By default, the system will use the highest method of encryption that is supported by the client! For Microsoft-based systems, as of Windows Vista, all Microsoft machines support AES encryption types! It's rather straightforward to configure Windows systems to only support specific encryption types for Kerberos. By default, a Windows Server 2016 will support the following encryption types:

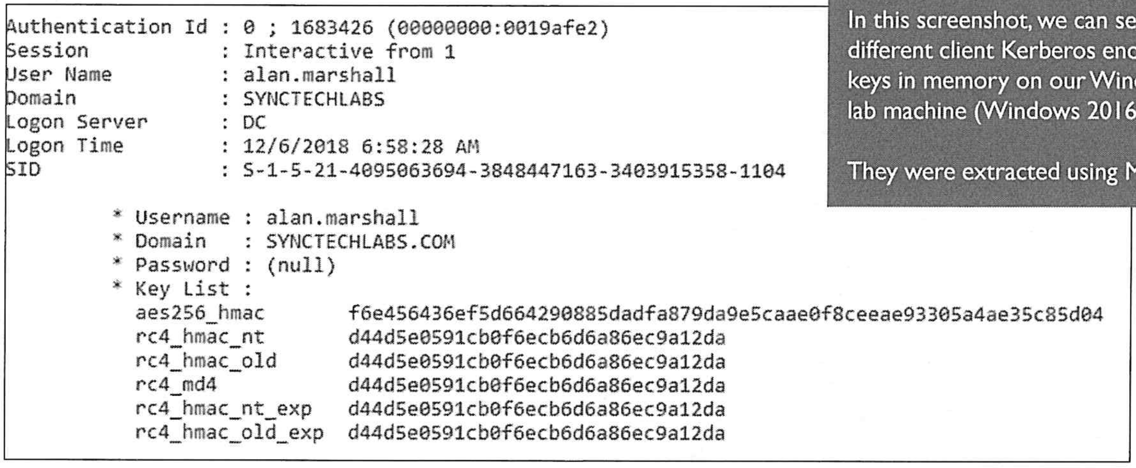
- DES_CBC_CRC
- DES_CBC_MD5
- RC4_HMAC_MD5
- AES128_HMAC_SHA1
- AES256_HMAC_SHA1

As indicated, as of Windows Vista, all Window systems support the rather secure AES encryption method (which uses 4096 iterations of PBKDF2). Supported Kerberos encryption types can be configured under the following setting:

Computer Configuration -> Policies -> Windows Settings -> Security Settings -> Local Policies -> Security Options -> Network Security: Configure encryption types allowed for Kerberos

© 2019 Erik Van Buggenhout & Stephen Sims

Kerberos Encryption Types – Our Environment



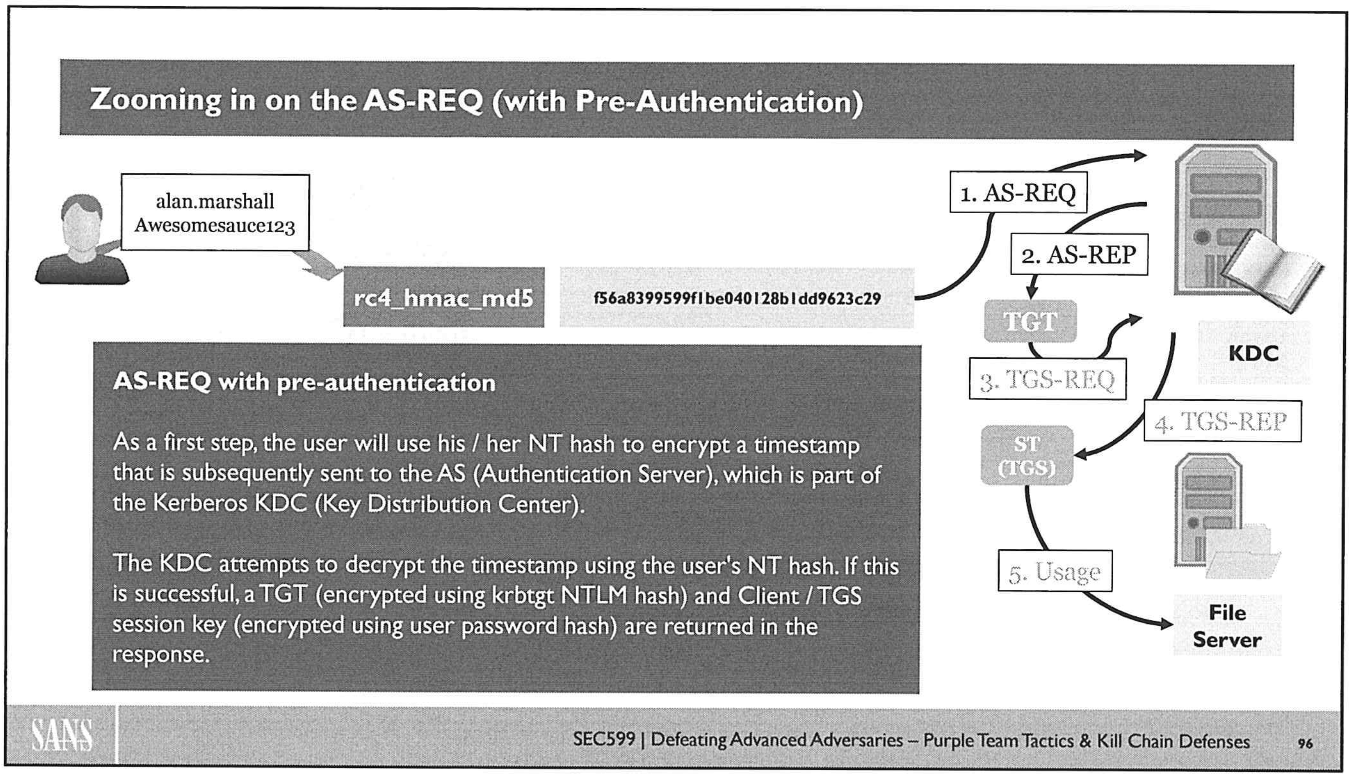
```
Authentication Id : 0 ; 1683426 (00000000:0019afe2)
Session           : Interactive from 1
User Name         : alan.marshall
Domain            : SYNCTECHLABS
Logon Server      : DC
Logon Time        : 12/6/2018 6:58:28 AM
SID               : S-1-5-21-4095063694-3848447163-3403915358-1104

         * Username : alan.marshall
         * Domain   : SYNCTECHLABS.COM
         * Password : (null)
         * Key List :
           aes256_hmac       f6e456436ef5d664290885dadfa879da9e5caae0f8ceeae93305a4ae35c85d04
           rc4_hmac_nt       d44d5e0591cb0f6ecb6d6a86ec9a12da
           rc4_hmac_old      d44d5e0591cb0f6ecb6d6a86ec9a12da
           rc4_md4           d44d5e0591cb0f6ecb6d6a86ec9a12da
           rc4_hmac_nt_exp   d44d5e0591cb0f6ecb6d6a86ec9a12da
           rc4_hmac_old_exp  d44d5e0591cb0f6ecb6d6a86ec9a12da
```

In this screenshot, we can see the different client Kerberos encryption keys in memory on our Windows 10 lab machine (Windows 2016 DC).

They were extracted using Mimikatz!

Kerberos Encryption Types – Our Environment
In this screenshot, we can see the different client Kerberos encryption keys in memory on our Windows 10 lab machine (Windows 2016 DC). We are using Mimikatz to extract credentials and you can see that both an AES key and an RC4 key are present! As it's Windows 10 with a Windows 2016, however, the system will rely on AES by default when using Kerberos.



Zooming in on the AS-REQ (with Pre-Authentication)

AS-REQ with pre-authentication

As a first step, the user will use his / her NT hash to encrypt a timestamp that is subsequently sent to the AS (Authentication Server), which is part of the Kerberos KDC (Key Distribution Center).

The KDC attempts to decrypt the timestamp using the user's NT hash. If this is successful, a TGT (encrypted using krbtgt NTLM hash) and Client / TGS session key (encrypted using user password hash) are returned in the response.

SANS

SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 96

Zooming in on the AS-REQ (with Pre-Authentication)
Let's have a look at the initial step here AS-REQ!

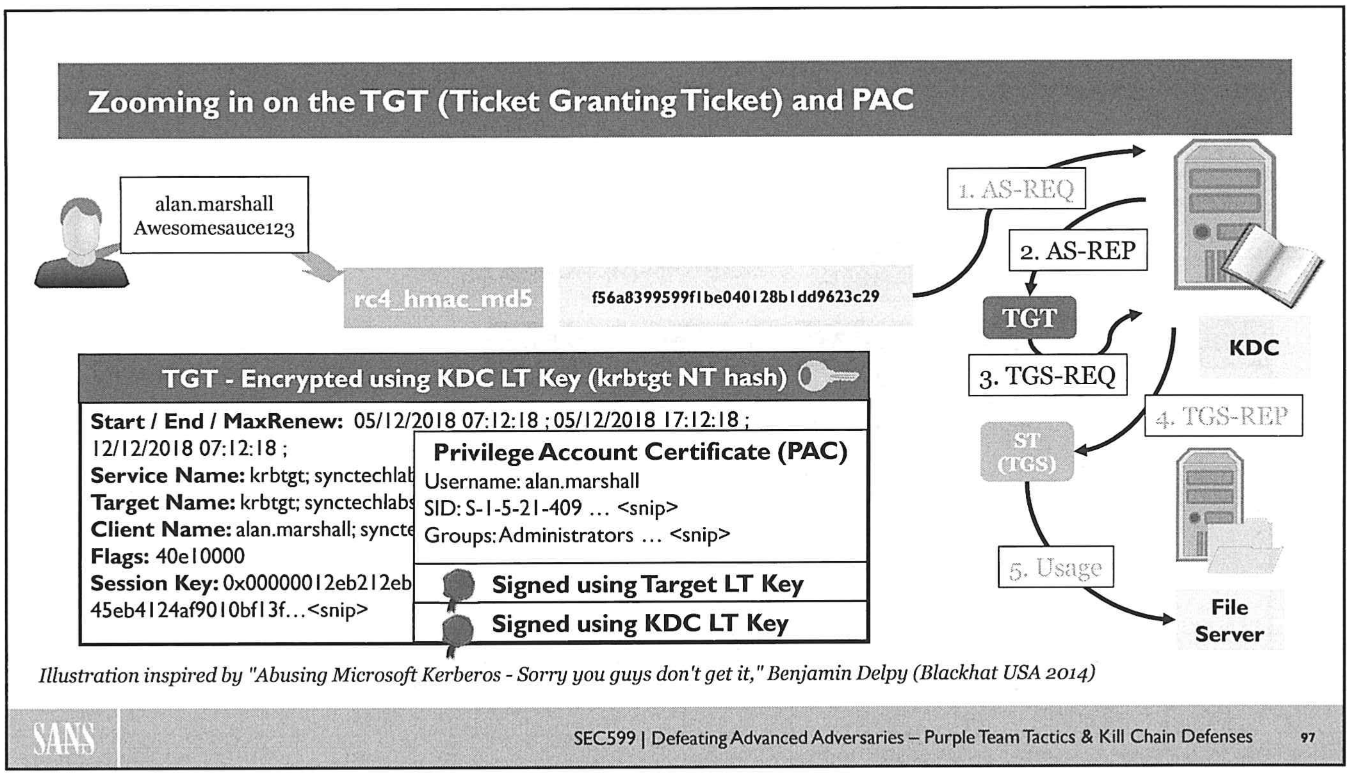
1. AS-REQ – As an initial step, the client performs pre-authentication by sending an AS-REQ to the KDC (in this case, the AS component of the KDC). This AS-REQ includes an encrypted version of the timestamp (encrypted using the password hash of the client account), which is validated by the KDC. This "encrypted timestamp" is referred to as "pre-authentication" and is enabled (required) by default in Microsoft Kerberos environments.

2. AS-REP – If pre-authentication succeeds (timestamp can be decrypted), the KDC sends two items to the client:

   - The Ticket Granting Ticket (TGT), which includes the Client / TGS session key and is encrypted using the KDC long-term key (in case of rc4_hmac_md5, the krbtgt NT password hash). This KDC long-term key is also used to sign the PAC (which is also part of the TGT and includes information on who the user is and what groups he is a member of).
   - The Client / TGS session key, encrypted using the client long-term key (in case of rc4_hmac_md5, the user password hash).

   It's interesting to note that when pre-authentication would not be enabled, the KDC will provide an AS-REP response without validating the encrypted timestamp. While this is not a "critical" issue that would compromise security of the Kerberos protocol (both response messages would still be encrypted with a key unavailable to the adversary), it would allow adversaries to launch brute-force attacks against the part of the response that is encrypted using the client long-term key (in case of rc4_hmac_md5, the user password hash).

© 2019 Erik Van Buggenhout & Stephen Sims



Zooming in on the TGT (Ticket Granting Ticket) and PAC

TGT - Encrypted using KDC LT Key (krbtgt NT hash)

Start / End / MaxRenew: 05/12/2018 07:12:18 ; 05/12/2018 17:12:18 ; 12/12/2018 07:12:18 ;
Service Name: krbtgt; synctechlat
Target Name: krbtgt; synctechlabs
Client Name: alan.marshall; syncte
Flags: 40e10000
Session Key: 0x0000012eb212eb 45eb4124af9010bf13f…<snip>

Privilege Account Certificate (PAC)
Username: alan.marshall
SID: S-1-5-21-409 … <snip>
Groups: Administrators … <snip>

Signed using Target LT Key

Signed using KDC LT Key

Illustration inspired by "Abusing Microsoft Kerberos - Sorry you guys don't get it," Benjamin Delpy (Blackhat USA 2014)

Zooming in on the TGT (Ticket Granting Ticket) and PAC
The TGT is the first ticket received by the client. What does it look like? The TGT includes the following information:

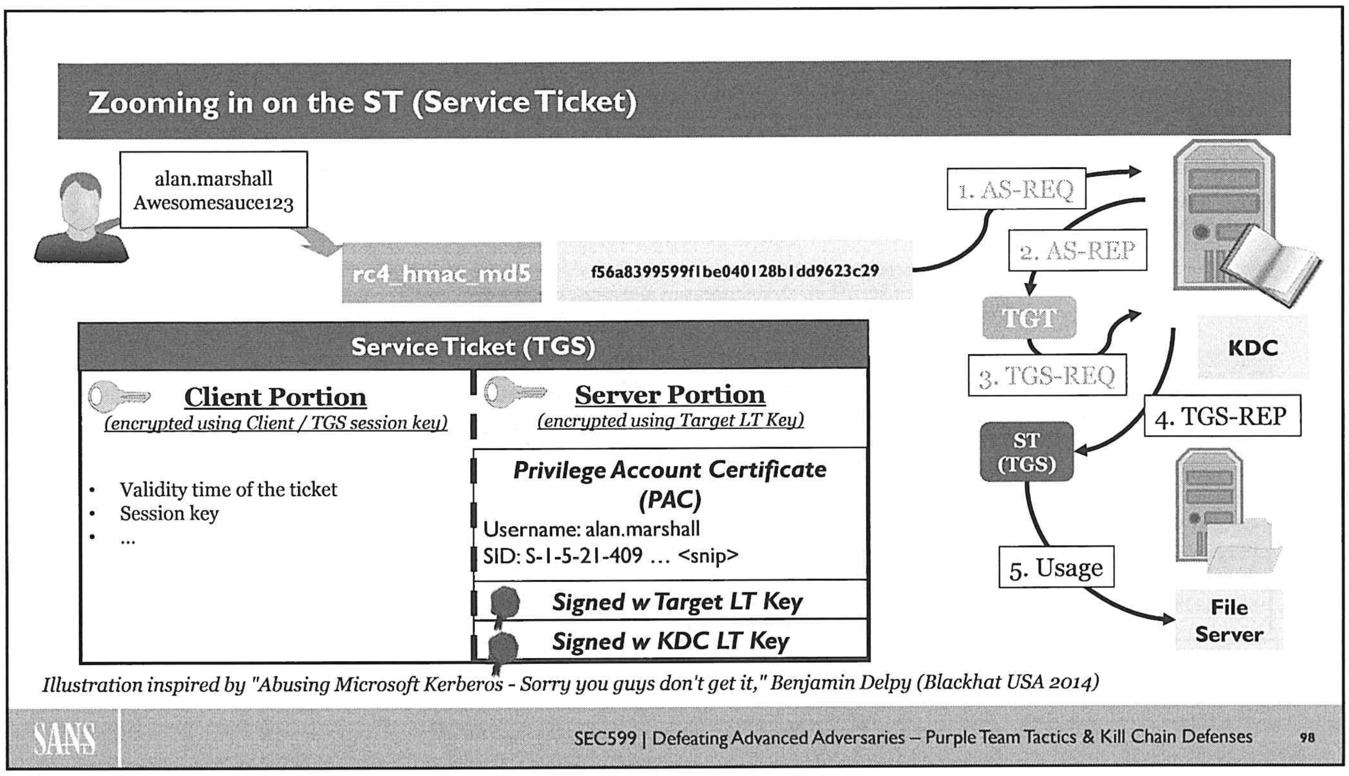
- Name: The user's name the ticket is associated with.
- Start time and End time: marks the validity period of the ticket. By default, in Windows AD environments, this would be 10 hours.
- Authorization-data: Authorization data details the user's privileges and access rights. In Windows, the authorization data takes the form of a Privilege Attribute Certificate (PAC) object.
- The Client / TGS session key that can be used for future communications between the client and the TGS.

It should be clear that the PAC is extremely sensitive and should under no circumstance be tampered with. In order to protect its integrity, it is signed with two keys:

- It is signed using the Target Long Term Key (LT Key). Because this is a TGT, the target is the krbtgt account (so in rc4_hmac_md5, this would be the krbtgt NT hash).
- It is also signed using the KDC LT Key. As indicated many times before, in rc4_hmac_md5, this would be the krbtgt NT hash.

Finally, to prevent the entire TGT from being tampered with, it is encrypted using the KDC long-term key (which is the krbtgt NT hash when rc4_hmac_md5 is used). The TGT cannot be read by the client, but that's fine, as it only needs to send it back to the KDC for future validation (e.g. when a Service Ticket is requested). In the TGS-REQ, the user wants to authenticate to a certain service and thus sends the following to the KDC (in this case, the TGS component of the KDC):

- An authenticator message (encrypted using the Client / TGS session key).
- The encrypted TGT and a ticket request (referencing a certain Service Principle Name, SPN).



Zooming in on the ST (Service Ticket)

- Validity time of the ticket
- Session key
- ...

Illustration inspired by "Abusing Microsoft Kerberos - Sorry you guys don't get it," Benjamin Delpy (Blackhat USA 2014)

SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 98

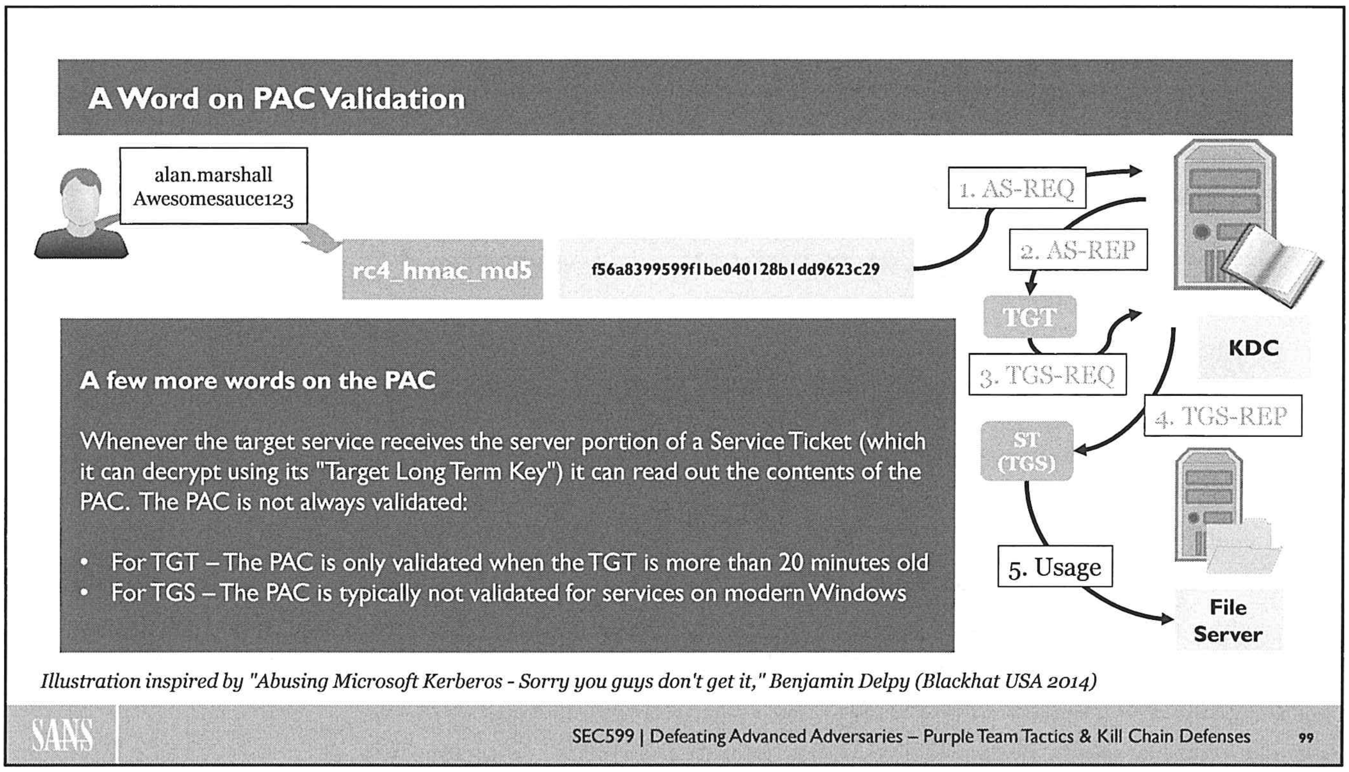
Zooming in on the ST (Service Ticket)
If the KDC receives a TGS-REQ with a valid TGT, it will send back a response, TGS-REP. The KDC will validate whether the service to which the client requests access exists and subsequently creates a Service Ticket (ST) that is sent back. It's important to note that the KDC does not do any validation of privileges (the target service can do that by itself, by reviewing the PAC that is included in the Service Ticket, see below).

The Service Ticket has two parts:

- A client portion, which is encrypted using the Client / TGS session key (so this can be decrypted by the client).
- A server portion, which is encrypted using the target long-term key (in case of rc4_hmac_md5, the password hash of the target service). This server portion also includes the PAC of the user. It is this Service Ticket that the user will subsequently submit to the service he or she is trying to access.

Illustration inspired by "Abusing Microsoft Kerberos - Sorry you guys don't get it," Benjamin Delpy (Blackhat USA 2014).

© 2019 Erik Van Buggenhout & Stephen Sims



A Word on PAC Validation

A few more words on the PAC

Whenever the target service receives the server portion of a Service Ticket (which it can decrypt using its "Target Long Term Key") it can read out the contents of the PAC. The PAC is not always validated:

- For TGT – The PAC is only validated when the TGT is more than 20 minutes old
- For TGS – The PAC is typically not validated for services on modern Windows

Illustration inspired by "Abusing Microsoft Kerberos - Sorry you guys don't get it," Benjamin Delpy (Blackhat USA 2014)

A Word on PAC Validation

Whenever the target service receives the server portion of a Service Ticket (which it can decrypt using its "Target Long Term Key"), it can read out the contents of the PAC. The target service will validate the signature that was created using the Target Long Term Key but will not always validate the signature that was created using the KDC Long Term Key!

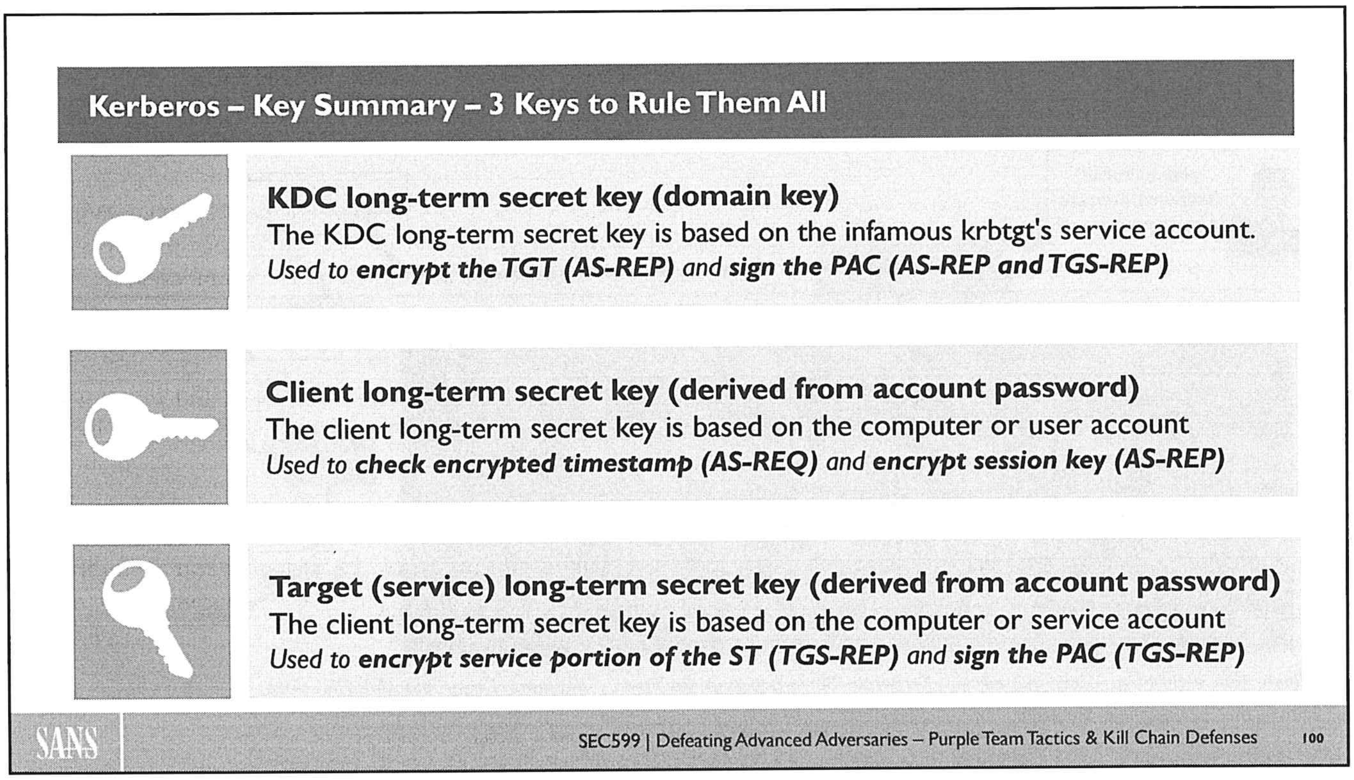
This is from Microsoft's MS-KILE specification:
"Kerberos V5 does not provide account revocation checking for TGS requests, which allows TGT renewals and Service Tickets to be issued as long as the TGT is valid even if the account has been revoked. KILE provides a check account policy (section 3.3.5.7.1) that limits the exposure to a shorter time. KILE KDCs in the account domain are required to check accounts when the TGT is older than 20 minutes. This limits the period that a client can get a ticket with a revoked account while limiting the performance cost for AD queries."

What does this practically mean for us?

- For a TGT: If the TGT is more than 20 minutes old, the PAC contents are validated. This thus opens a 20-minute window where the contents are not validated. This means that, if you have the KDC long-term hash (NTLM hash of the krbtgt account when using rc4_hmac_md5), you can even create tickets with bogus account information (e.g. user "Invisible" as a member of the "Domain Admins" group), which will be valid for 20 minutes. Within these 20 minutes, you could request multiple Service Tickets that are valid for 10 hours.

- For a Service Ticket: Modern Windows systems (as of Windows Vista) do not validate the PAC by default for services. Windows still validates the PAC for processes that are not running as services. PAC validation can be enabled when the application server is not running in the context of local system, network service, or local service.

We will discuss some possible attack vectors soon!



Kerberos – Key Summary – 3 Keys to Rule Them All

KDC long-term secret key (domain key)
The KDC long-term secret key is based on the infamous krbtgt's service account.
Used to ***encrypt the TGT (AS-REP)*** *and* ***sign the PAC (AS-REP and TGS-REP)***

Client long-term secret key (derived from account password)
The client long-term secret key is based on the computer or user account
Used to ***check encrypted timestamp (AS-REQ)*** *and* ***encrypt session key (AS-REP)***

Target (service) long-term secret key (derived from account password)
The client long-term secret key is based on the computer or service account
Used to ***encrypt service portion of the ST (TGS-REP)*** *and* ***sign the PAC (TGS-REP)***

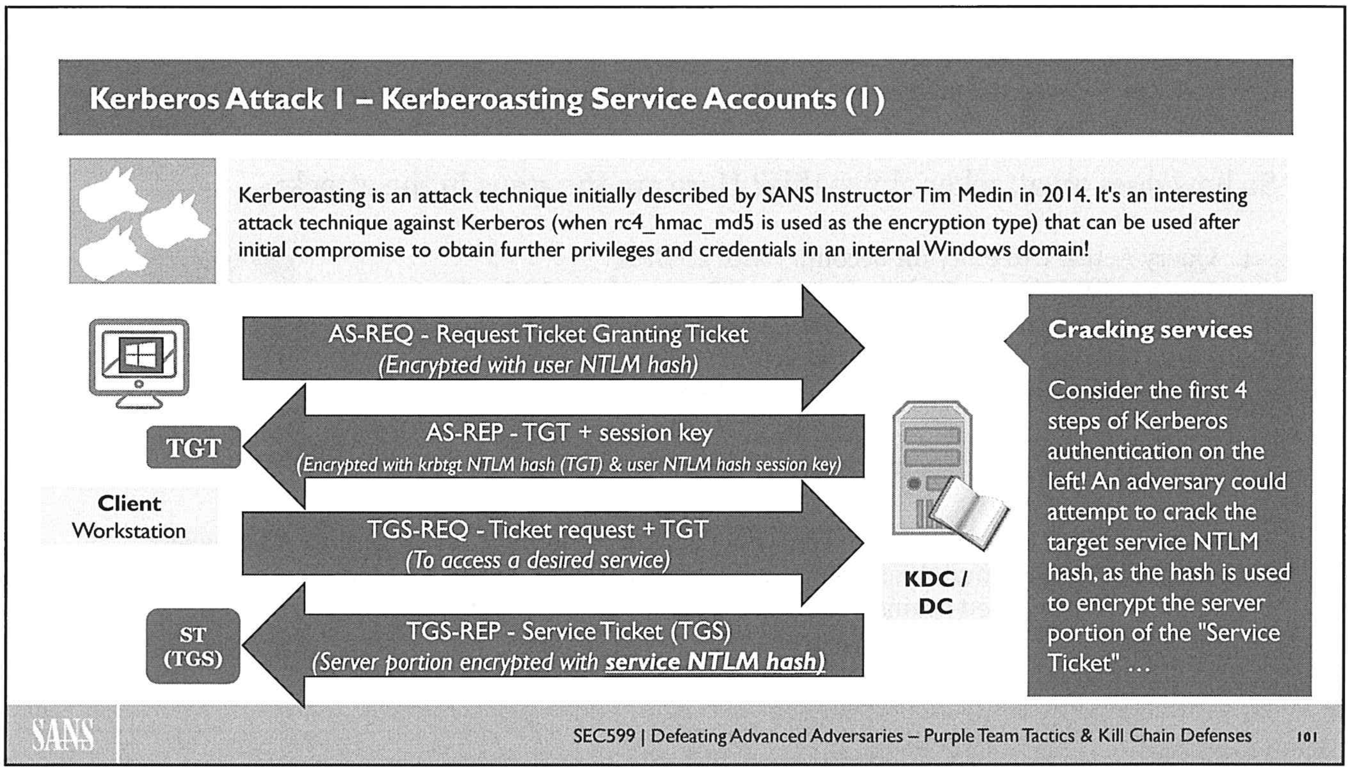
SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 100

Kerberos – Key Summary – 3 Keys to Rule Them All
Although Kerberos can look complex, it's actually perfectly understandable! It's important to note that three main security keys are used throughout the protocol:

- The KDC long-term secret key, often called the "domain key", which is derived from the krbtgt service account. It is used in Kerberos to encrypt the TGT and sign the PAC.
- The client long-term secret key, which is based on the password of the client account. It is used in Kerberos to check the encrypted timestamp (AS-REQ) and encrypt the session key.
- The target long-term secret key, which is based on the password of the target service. It is used in Kerberos to encrypt the TGS and sign the PAC.

It should be clear that compromise of any key represents a security risk. There is, however, one key that is more important than the others: If the KDC long-term secret key is compromised, adversaries can recreate TGTs and sign PACs, which would allow them to obtain all privileges within the domain.

© 2019 Erik Van Buggenhout & Stephen Sims



Kerberos Attack 1 – Kerberoasting Service Accounts (1)

Kerberoasting is an attack technique initially described by SANS Instructor Tim Medin in 2014. It's an interesting attack technique against Kerberos (when rc4_hmac_md5 is used as the encryption type) that can be used after initial compromise to obtain further privileges and credentials in an internal Windows domain!

AS-REQ - Request Ticket Granting Ticket
(Encrypted with user NTLM hash)

AS-REP - TGT + session key
(Encrypted with krbtgt NTLM hash (TGT) & user NTLM hash session key)

TGS-REQ - Ticket request + TGT
(To access a desired service)

TGS-REP - Service Ticket (TGS)
*(Server portion encrypted with **service NTLM hash**)*

Cracking services

Consider the first 4 steps of Kerberos authentication on the left! An adversary could attempt to crack the target service NTLM hash, as the hash is used to encrypt the server portion of the "Service Ticket" ...

Kerberos Attack 1 – Kerberoasting Service Accounts (1)
Kerberoasting is an attack technique initially described by SANS Instructor Tim Medin in 2014. It's an interesting attack technique against Kerberos (when rc4_hmac_md5 is used as the encryption type) that can be used after initial compromise to obtain further privileges and credentials in an internal Windows domain! The slides of his original talk have been published here:

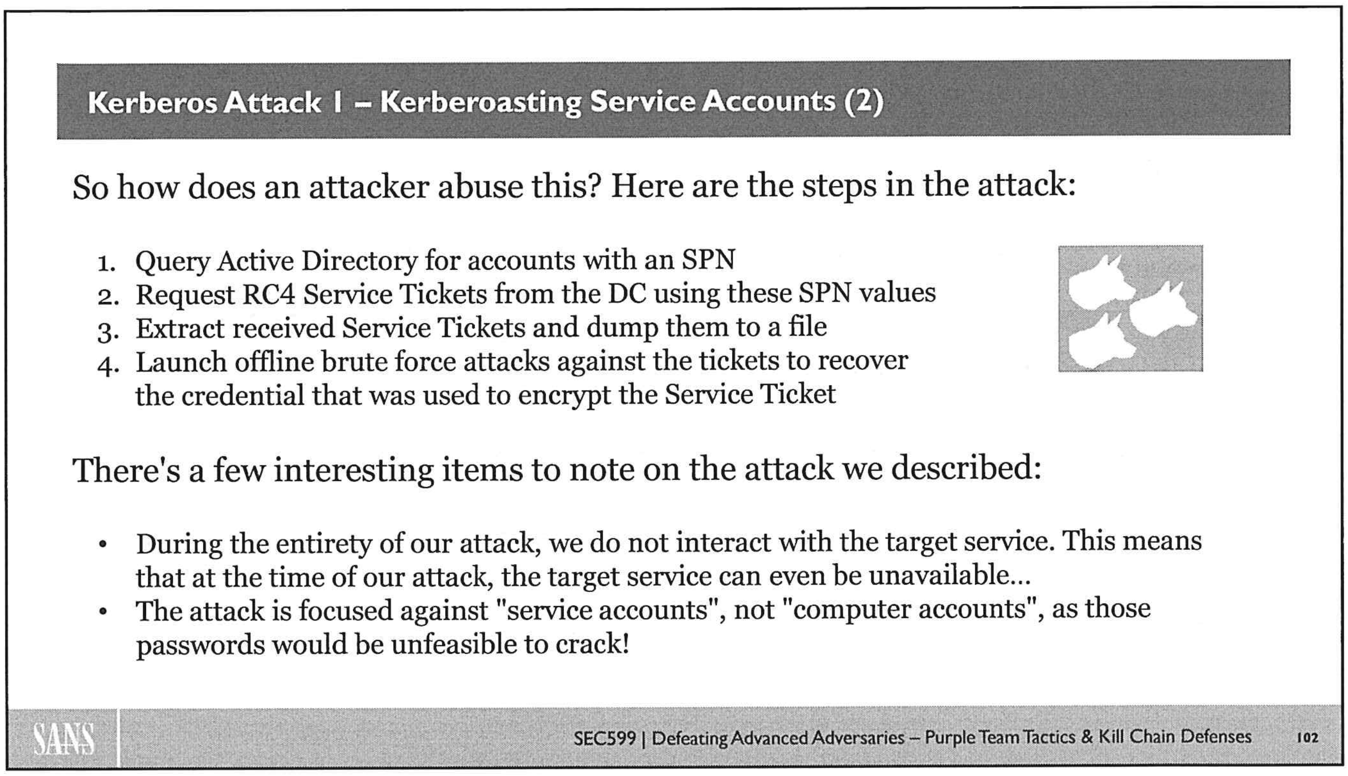
https://www.sans.org/cyber-security-summit/archives/file/summit-archive-1493862736.pdf

How does it work?

Let's review the 4 first steps of Kerberos authentication:

1. As an initial step, the user requests a Ticket Granting Ticket (TGT) from the domain controller (who acts as a Key Distribution Center). When rc4_hmac_md5 is used, the request is encrypted using the user's NTLM hash.
2. As a second step, the user receives a Ticket Granting Ticket and a session key that are encrypted using the krbtgt's NTLM hash.
3. Thirdly, the user wants to authenticate to a certain service and thus sends a ticket request to the KDC (DC).
4. The KDC does not do any validation of privileges (it leaves that to the target service) and sends back a Service Ticket, of which the server portion is encrypted using the NTLM hash of the target service. It is this Service Ticket that the user will subsequently submit to the service he or she is trying to access. The service will be able to read the contents of the Service Ticket by decrypting it with its hash.

Given the above, an adversary could attempt to crack the target service NTLM hash, as the hash is used to encrypt the server portion of the "Service Ticket"...



Kerberos Attack 1 – Kerberoasting Service Accounts (2)

So how does an attacker abuse this? Here are the steps in the attack:

1. Query Active Directory for accounts with an SPN
2. Request RC4 Service Tickets from the DC using these SPN values
3. Extract received Service Tickets and dump them to a file
4. Launch offline brute force attacks against the tickets to recover the credential that was used to encrypt the Service Ticket

There's a few interesting items to note on the attack we described:

- During the entirety of our attack, we do not interact with the target service. This means that at the time of our attack, the target service can even be unavailable...
- The attack is focused against "service accounts", not "computer accounts", as those passwords would be unfeasible to crack!

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 102

Kerberos Attack 1 – Kerberoasting Service Accounts (2)
So how could an attacker launch such an attack? Several tools are available that implement this attack strategy. Most of them operate using the following steps:

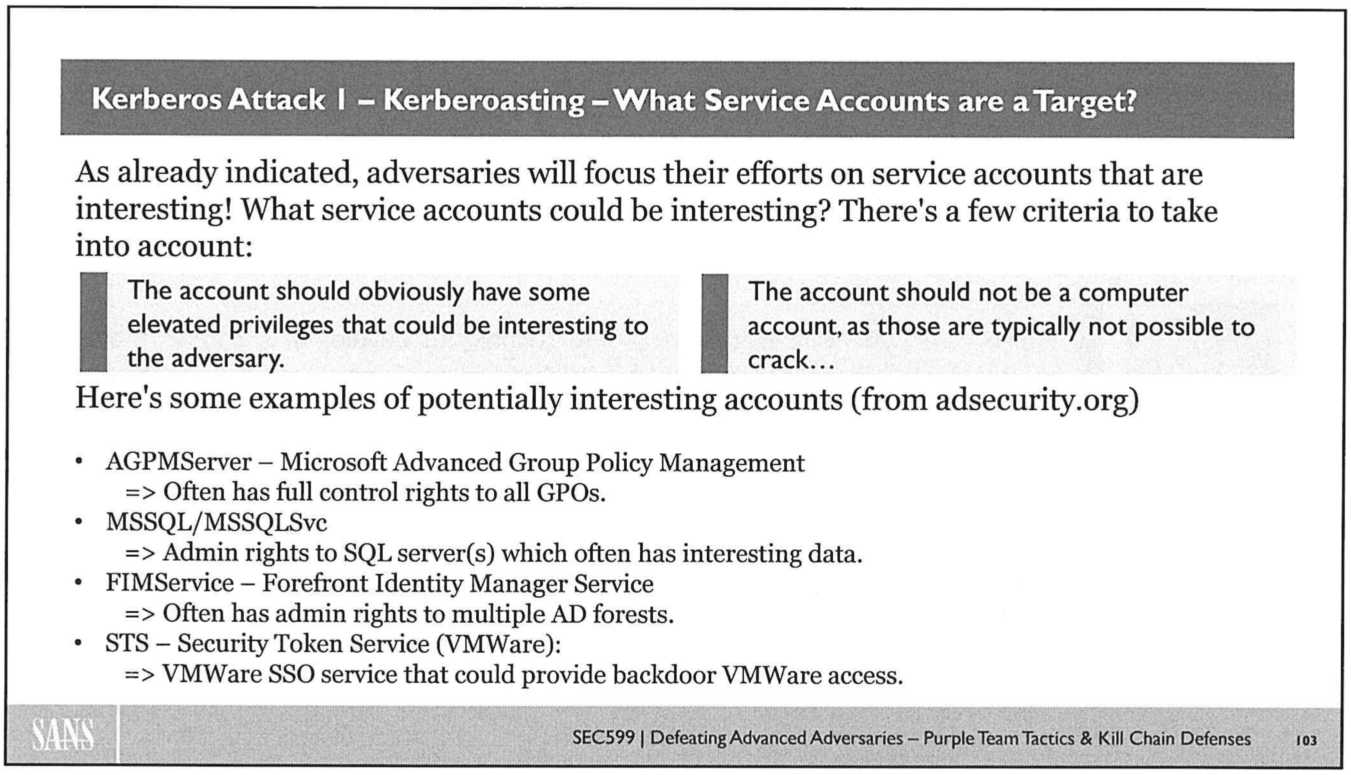
1. Query Active Directory for accounts with a Service Principal Name (SPN).
2. Request RC4 Service Tickets from the DC using these SPN values.
3. Extract received Service Tickets and dump them to a file.
4. Launch offline brute force attacks against the tickets to recover the credential that was used to encrypt the Service Ticket.

There are a few interesting items to note on the attack we described:

- During the entirety of our attack, we do not interact with the target service. This means that at the time of our attack, the target service can even be unavailable… As a defender, this means that detection capabilities are limited (we will get back to that shortly!).
- The attack is focused against "service accounts", not "computer accounts", as those passwords would be unfeasible to crack!

So, let's have a look at some of the tools used for this…

© 2019 Erik Van Buggenhout & Stephen Sims



Kerberos Attack 1 – Kerberoasting – What Service Accounts are a Target?

As already indicated, adversaries will focus their efforts on service accounts that are interesting! What service accounts could be interesting? There's a few criteria to take into account:

The account should obviously have some elevated privileges that could be interesting to the adversary.

The account should not be a computer account, as those are typically not possible to crack...

Here's some examples of potentially interesting accounts (from adsecurity.org)

- AGPMServer – Microsoft Advanced Group Policy Management
  => Often has full control rights to all GPOs.
- MSSQL/MSSQLSvc
  => Admin rights to SQL server(s) which often has interesting data.
- FIMService – Forefront Identity Manager Service
  => Often has admin rights to multiple AD forests.
- STS – Security Token Service (VMWare):
  => VMWare SSO service that could provide backdoor VMWare access.

SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 103

Kerberos Attack 1 – Kerberoasting – What Service Accounts are a Target?
As already indicated, adversaries will focus their efforts on service accounts that are interesting!
What service accounts could be interesting? There can be quite a few of them! Here are a few criteria to take into account:

- The account should obviously have some elevated privileges that could be interesting to the adversary.
- The account should not be a computer account, as those are typically not possible to crack.

Here are some examples of potentially interesting accounts (from adsecurity.org)

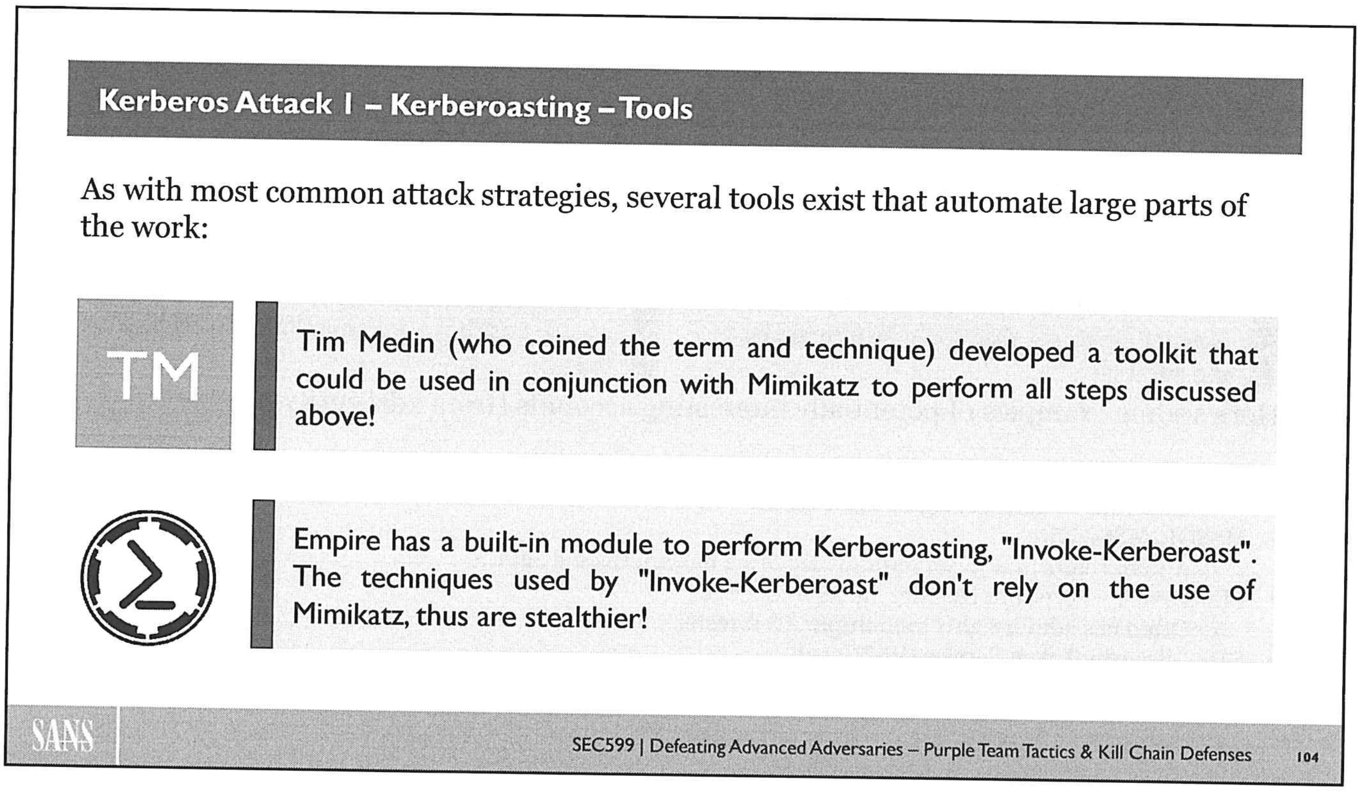
- AGPMServer – Microsoft Advanced Group Policy Management
  => Often has full control rights to all GPOs.

- MSSQL/MSSQLSvc
  => Admin rights to SQL server(s) which often has interesting data.

- FIMService – Forefront Identity Manager Service
  => Often has admin rights to multiple AD forests.

- STS – Security Token Service (VMWare):
  => VMWare SSO service which could provide backdoor VMWare access.

Depending on your exact configuration, you may, of course, have other interesting service accounts with elevated privileges!



Kerberos Attack 1 – Kerberoasting – Tools

As with most common attack strategies, several tools exist that automate large parts of the work:

TM — Tim Medin (who coined the term and technique) developed a toolkit that could be used in conjunction with Mimikatz to perform all steps discussed above!

Empire has a built-in module to perform Kerberoasting, "Invoke-Kerberoast". The techniques used by "Invoke-Kerberoast" don't rely on the use of Mimikatz, thus are stealthier!

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 104

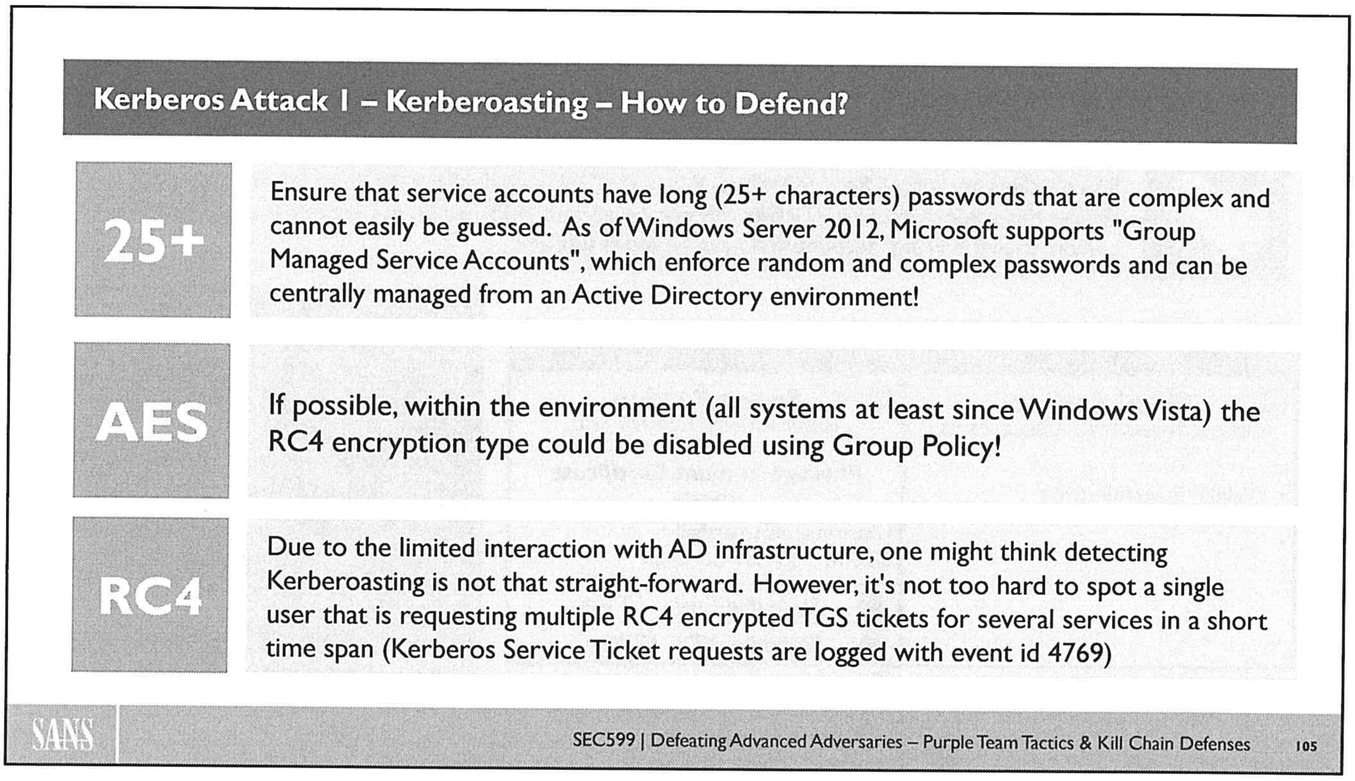
Kerberos Attack 1 – Kerberoasting – Tools
Based on the previous slides, it appears there's quite a bit of work that is to be done to mount a successful Kerberoasting attack… As with most common attack strategies, several tools, however, exist that automate large parts of the work:

- When he initially presented the Kerberoasting attack technique, Tim Medin developed a toolkit that could be used in conjunction with Mimikatz to perform all steps discussed above! Tim presented his slides at Pentest Hackfest and Derbycon in 2014!

- The well-known post-exploitation toolkit Empire has a built-in module to perform Kerberoasting, "Invoke-Kerberoast". The techniques used by "Invoke-Kerberoast" don't rely on the use of Mimikatz, thus are stealthier!

We will leverage Empire's tool during the upcoming lab, where we will test the technique and implement defenses!

© 2019 Erik Van Buggenhout & Stephen Sims



Kerberos Attack 1 – Kerberoasting – How to Defend?

25+ Ensure that service accounts have long (25+ characters) passwords that are complex and cannot easily be guessed. As of Windows Server 2012, Microsoft supports "Group Managed Service Accounts", which enforce random and complex passwords and can be centrally managed from an Active Directory environment!

AES If possible, within the environment (all systems at least since Windows Vista) the RC4 encryption type could be disabled using Group Policy!

RC4 Due to the limited interaction with AD infrastructure, one might think detecting Kerberoasting is not that straight-forward. However, it's not too hard to spot a single user that is requesting multiple RC4 encrypted TGS tickets for several services in a short time span (Kerberos Service Ticket requests are logged with event id 4769)

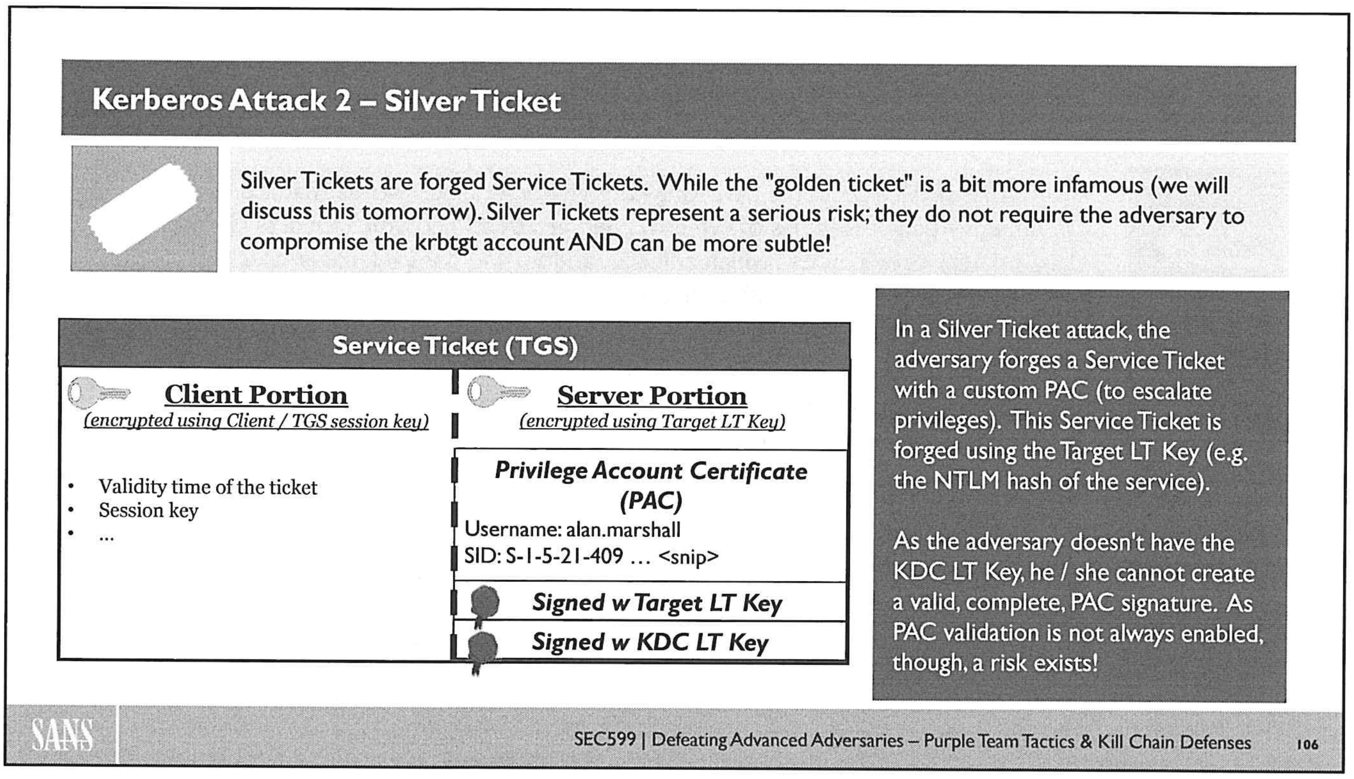
SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 105

Kerberos Attack 1 – Kerberoasting – How to Defend?
So how can we defend against Kerberoasting? There are a few things we could consider doing:

- First and foremost, we should ensure that service accounts have long (25+ characters) passwords that are complex and cannot easily be guessed. Remember: An adversary still has to crack the password, so we can leverage password complexity here! There's also an interesting feature in Windows as of Windows Server 2012. As of Windows Server 2012, Microsoft supports "Group Managed Service Accounts", which enforce random and complex passwords and can be centrally managed from the Active Directory environment!

- If possible, within the environment (all systems at least Windows Vista), the RC4 encryption type could be disabled at group policy level! Take those legacy systems into account, though!

- Finally, we'd like to provide some pointers that can help detect Kerberoasting in the environment. Due to the limited interaction with AD infrastructure, one might think detecting Kerberoasting is not that straight-forward. We could, however, try to detect a single user that is requesting several RC4 encrypted TGS tickets for several services (Kerberos Service Ticket requests are logged with event id 4769). Furthermore, the use of RC4 should be highly limited in today's environments, as the default cipher used by Windows systems as of Windows Vista is AES.



Kerberos Attack 2 – Silver Ticket

Silver Tickets are forged Service Tickets. While the "golden ticket" is a bit more infamous (we will discuss this tomorrow). Silver Tickets represent a serious risk; they do not require the adversary to compromise the krbtgt account AND can be more subtle!

Service Ticket (TGS)
Client Portion
(encrypted using Client / TGS session key)
Validity time of the ticket
Session key
...
Server Portion
(encrypted using Target LT Key)
Privilege Account Certificate (PAC)
Username: alan.marshall
SID: S-1-5-21-409 ... <snip>
Signed w Target LT Key
Signed w KDC LT Key

In a Silver Ticket attack, the adversary forges a Service Ticket with a custom PAC (to escalate privileges). This Service Ticket is forged using the Target LT Key (e.g. the NTLM hash of the service).

As the adversary doesn't have the KDC LT Key, he / she cannot create a valid, complete, PAC signature. As PAC validation is not always enabled, though, a risk exists!

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 106

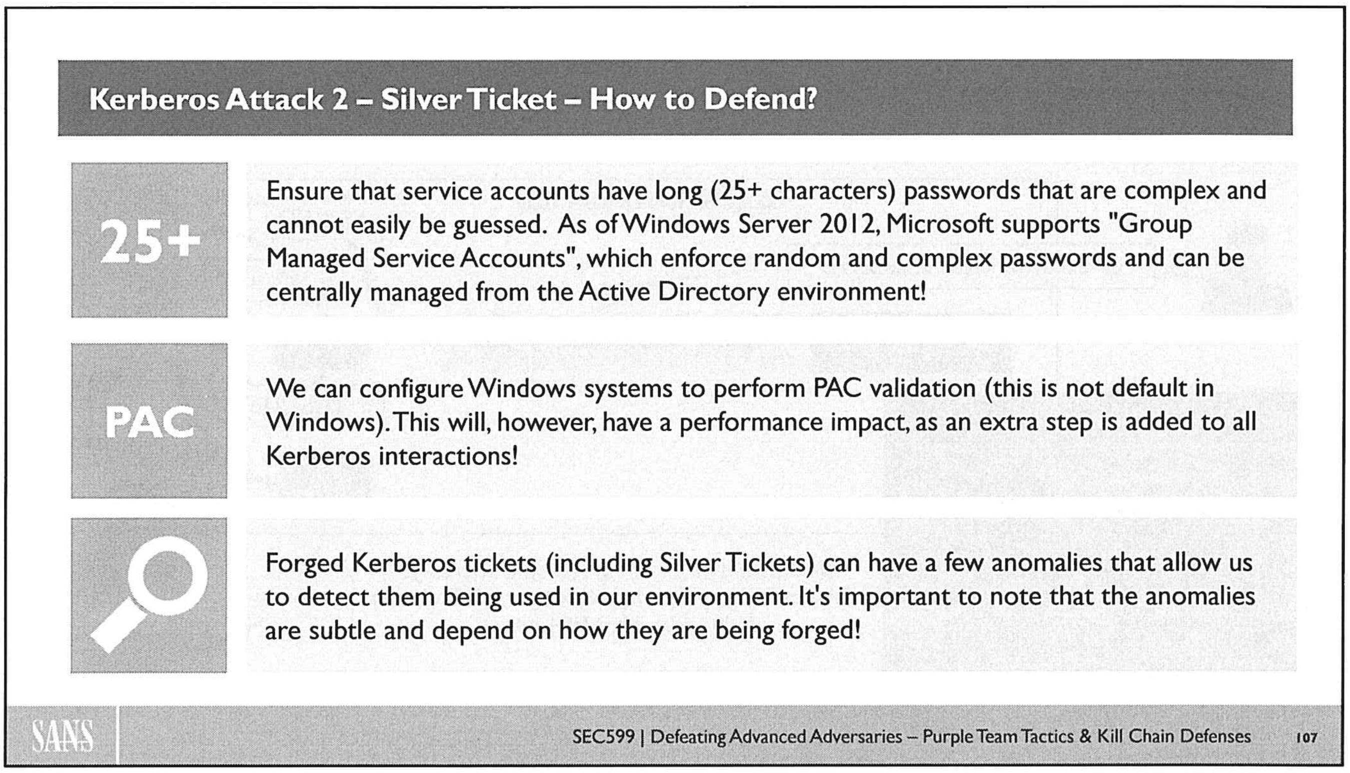
Kerberos Attack 2 – Silver Ticket
Silver Tickets are forged Service Tickets. While the "golden ticket" is a bit more infamous (we will discuss this tomorrow), Silver Tickets represent a serious risk: They do not require the adversary to compromise the krbtgt account AND can be more subtle!

In a Silver Ticket attack, the adversary forges a Service Ticket with a custom PAC (e.g. to escalate privileges). Mimikatz allows adversaries to forge silver tickets, using the following default group membership:

- Domain Users
- Domain Admins
- Schema Admins
- Enterprise Admins
- Group Policy Creator Owners

This Service Ticket is forged using the Target LT Key (e.g. the NTLM hash of the service). As the adversary doesn't have the KDC LT Key, he / she cannot create a valid, complete, PAC signature. As PAC validation is not always enabled, though, a risk exists!

© 2019 Erik Van Buggenhout & Stephen Sims



Kerberos Attack 2 – Silver Ticket – How to Defend?

Ensure that service accounts have long (25+ characters) passwords that are complex and cannot easily be guessed. As of Windows Server 2012, Microsoft supports "Group Managed Service Accounts", which enforce random and complex passwords and can be centrally managed from the Active Directory environment!

We can configure Windows systems to perform PAC validation (this is not default in Windows). This will, however, have a performance impact, as an extra step is added to all Kerberos interactions!

Forged Kerberos tickets (including Silver Tickets) can have a few anomalies that allow us to detect them being used in our environment. It's important to note that the anomalies are subtle and depend on how they are being forged!

Kerberos Attack 2 – Silver Ticket – How to Defend?
So how can we defend against these Silver Tickets? It's not that straightforward, unfortunately. There are a few things we could consider doing:

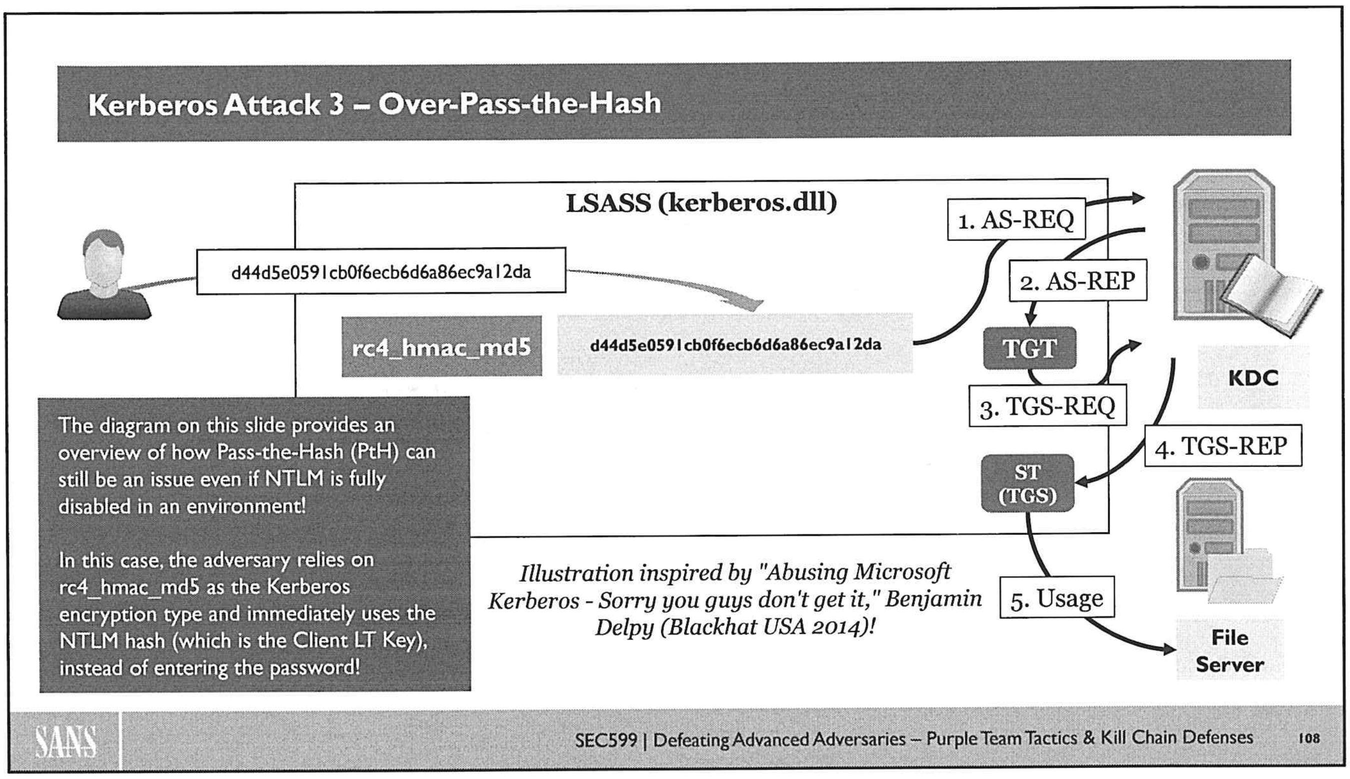
- First and foremost, as adversaries need the Target Long Term Keys (e.g. NTLM hash of service account), we should ensure that service accounts have long (25+ characters) passwords that are complex and cannot easily be guessed. Remember: An adversary still has to crack the password, so we can leverage password complexity here! There's also an interesting feature in Windows as of Windows Server 2012. As of Windows Server 2012, Microsoft supports "Group Managed Service Accounts", which enforce random and complex passwords and can be centrally managed from the Active Directory environment!

- We can configure Windows systems to perform PAC validation (this is not default in Windows). This will, however, have a performance impact, as an extra step is added in all of the authentications that occur. The registry key to create is the following:

  HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Parameters\ValidateKdcPacSignature (set value to 1)

- Finally, we would like to point out that forged Kerberos tickets (including Silver Tickets) can have a few anomalies that allow us to detect them being used in our environment. It's important to note that the anomalies are subtle and depend on how they are being forged! Some of the known anomalies include (from adsecurity.org - https://adsecurity.org/?p=1515):

  - The Account Domain field is blank when it should be DOMAIN.
  - The Account Domain field is DOMAIN FQDN when it should be DOMAIN.
  - In event 4624 (Logon): Account Domain is FQDN and should be short domain name.
  - In event 4634 (Logoff): Account Domain is blank and should be short domain name.
  - In event 4672 (Admin Logon): Account Domain is blank and should be short domain name.



Kerberos Attack 3 – Over-Pass-the-Hash

The diagram on this slide provides an overview of how Pass-the-Hash (PtH) can still be an issue even if NTLM is fully disabled in an environment!

In this case, the adversary relies on rc4_hmac_md5 as the Kerberos encryption type and immediately uses the NTLM hash (which is the Client LT Key), instead of entering the password!

Illustration inspired by "Abusing Microsoft Kerberos - Sorry you guys don't get it," Benjamin Delpy (Blackhat USA 2014)!

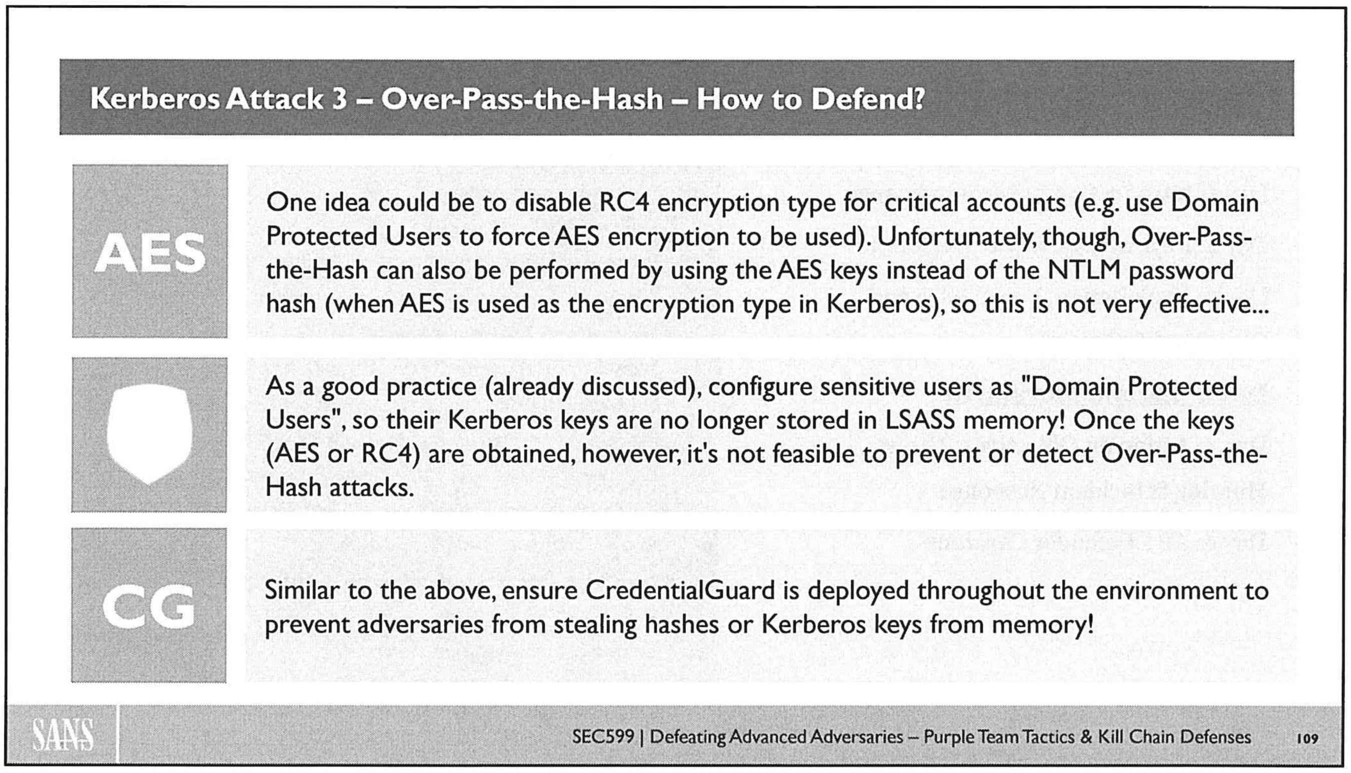
SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 108

Kerberos Attack 3 – Over-Pass-the-Hash
When you are reading this, it's good to remember the initial Pass-the-Hash attack into account. Remember that Pass-the-Hash relies on NTLM authentication, so a possible defense might be to just disable NTLM? While this is not a straightforward control to implement, it also doesn't fully protect us against Pass-the-Hash:

An adversary could use the stolen NTLM hash in order to "kick off" the Kerberos process and request tickets for a certain service: The diagram on this slide provides an overview of how Pass-the-Hash (PtH) can still be an issue even if NTLM is fully disabled in an environment! In this case, the adversary relies on rc4_hmac_md5 as the Kerberos encryption type and immediately uses the NT hash (which is the Client LT Key), instead of entering the password!

This technique is commonly referred to as "Over-Pass-the-Hash".

© 2019 Erik Van Buggenhout & Stephen Sims



Kerberos Attack 3 – Over-Pass-the-Hash – How to Defend?

AES — One idea could be to disable RC4 encryption type for critical accounts (e.g. use Domain Protected Users to force AES encryption to be used). Unfortunately, though, Over-Pass-the-Hash can also be performed by using the AES keys instead of the NTLM password hash (when AES is used as the encryption type in Kerberos), so this is not very effective...

As a good practice (already discussed), configure sensitive users as "Domain Protected Users", so their Kerberos keys are no longer stored in LSASS memory! Once the keys (AES or RC4) are obtained, however, it's not feasible to prevent or detect Over-Pass-the-Hash attacks.

CG — Similar to the above, ensure CredentialGuard is deployed throughout the environment to prevent adversaries from stealing hashes or Kerberos keys from memory!

SANS — SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 109

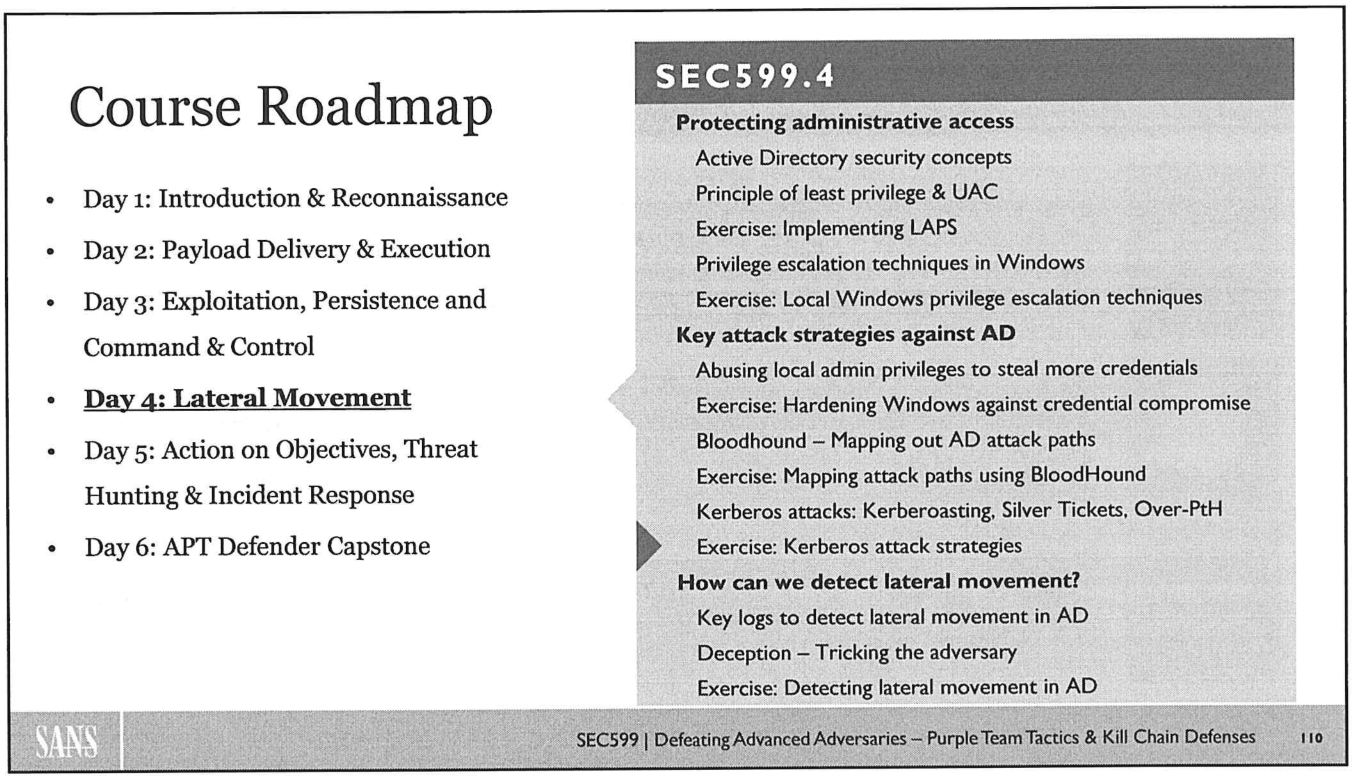
Kerberos Attack 3 – Over-Pass-the-Hash – How to Defend?
How can we defend ourselves against Over-Pass-the-hash? This is not straightforward, here are some ideas:

- One idea could be to disable RC4 encryption type for critical accounts (e.g. use Domain Protected Users to force AES encryption to be used). Unfortunately though, Over-Pass-the-Hash can also be performed by using the AES keys instead of the NTLM password hash (when AES is used as the encryption type in Kerberos), so this is not very effective...

- As a good practice (already discussed), configure sensitive users as "Domain Protected Users", so their Kerberos keys are no longer stored in LSASS memory! Once the keys (AES or RC4) are obtained, however, it's not feasible to prevent or detect Over-Pass-the-Hash attacks.

- Similar to the above, ensure CredentialGuard is deployed throughout the environment to prevent adversaries from stealing hashes or Kerberos keys from memory!

SANS Instructor Chad Tilbury created an interesting presentation that discusses some of these controls (and others) here: https://www.sans.org/cyber-security-summit/archives/file/summit-archive-1498398319.pdf



Course Roadmap

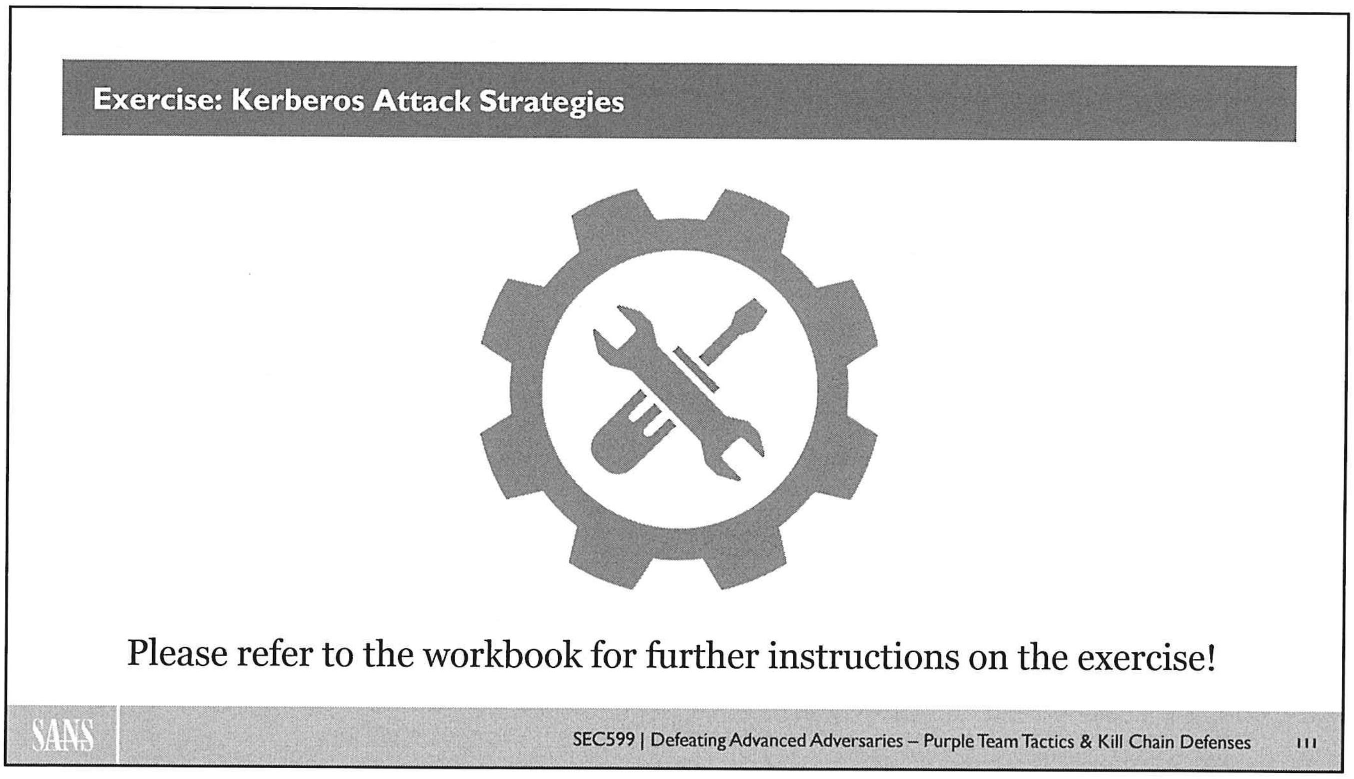
- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- **Day 4: Lateral Movement**
- Day 5: Action on Objectives, Threat Hunting & Incident Response
- Day 6: APT Defender Capstone

SEC599.4

SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 110

This page intentionally left blank.

© 2019 Erik Van Buggenhout & Stephen Sims

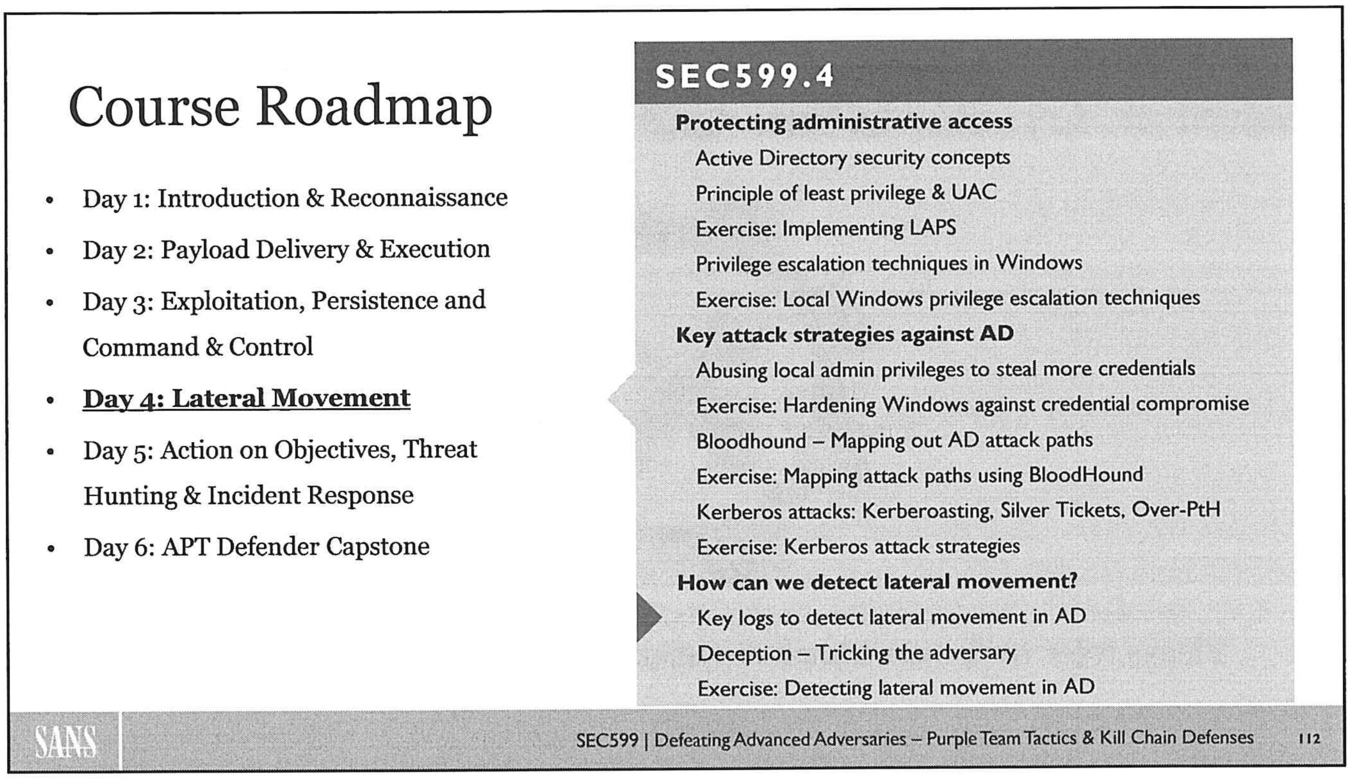


Exercise: Kerberos Attack Strategies

Please refer to the workbook for further instructions on the exercise!

SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 111

This page intentionally left blank.



Course Roadmap

- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- **Day 4: Lateral Movement**
- Day 5: Action on Objectives, Threat Hunting & Incident Response
- Day 6: APT Defender Capstone

SEC599.4

Protecting administrative access
- Active Directory security concepts
- Principle of least privilege & UAC
- Exercise: Implementing LAPS
- Privilege escalation techniques in Windows
- Exercise: Local Windows privilege escalation techniques

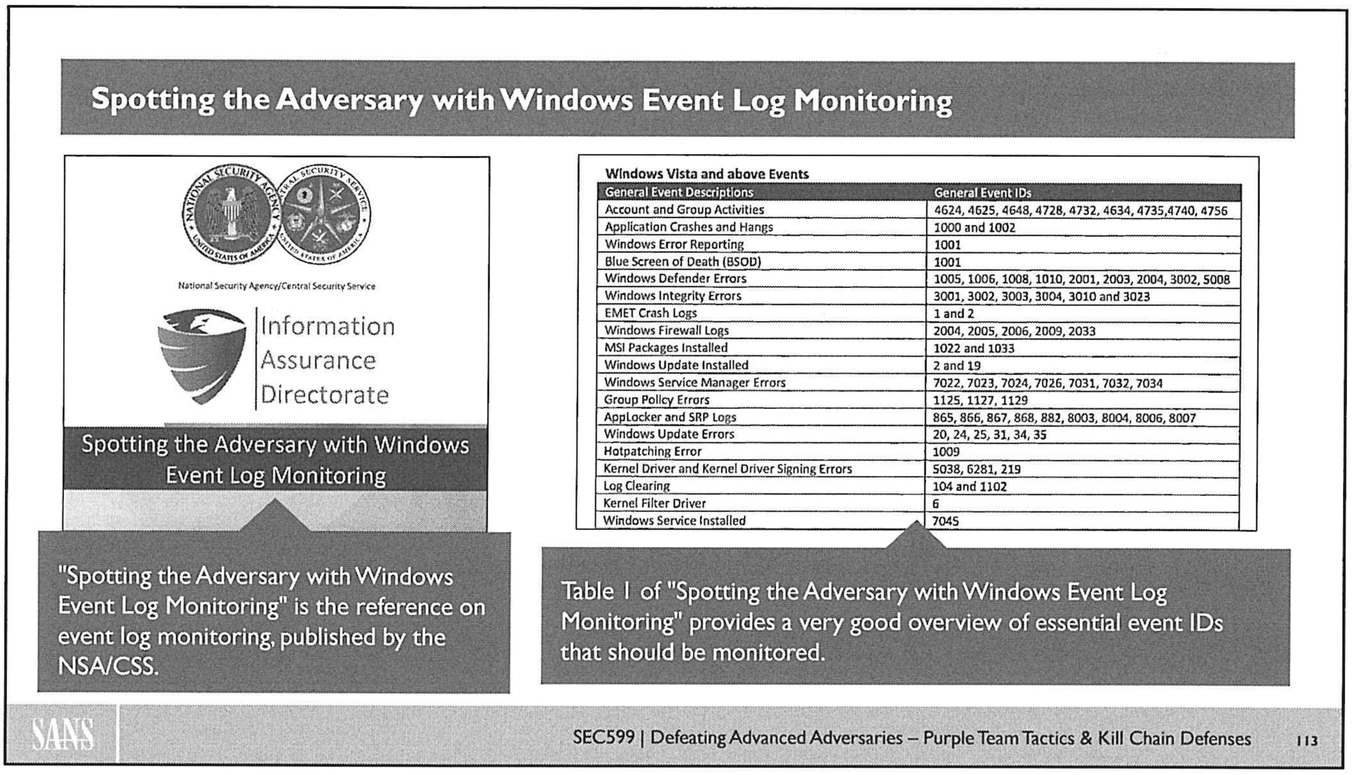
Key attack strategies against AD
- Abusing local admin privileges to steal more credentials
- Exercise: Hardening Windows against credential compromise
- Bloodhound – Mapping out AD attack paths
- Exercise: Mapping attack paths using BloodHound
- Kerberos attacks: Kerberoasting, Silver Tickets, Over-PtH
- Exercise: Kerberos attack strategies

How can we detect lateral movement?
- Key logs to detect lateral movement in AD
- Deception – Tricking the adversary
- Exercise: Detecting lateral movement in AD

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 112

This page intentionally left blank.

© 2019 Erik Van Buggenhout & Stephen Sims



Spotting the Adversary with Windows Event Log Monitoring

National Security Agency/Central Security Service

Information Assurance Directorate

Spotting the Adversary with Windows Event Log Monitoring

"Spotting the Adversary with Windows Event Log Monitoring" is the reference on event log monitoring, published by the NSA/CSS.

Windows Vista and above Events

General Event Descriptions	General Event IDs
Account and Group Activities	4624, 4625, 4648, 4728, 4732, 4634, 4735,4740, 4756
Application Crashes and Hangs	1000 and 1002
Windows Error Reporting	1001
Blue Screen of Death (BSOD)	1001
Windows Defender Errors	1005, 1006, 1008, 1010, 2001, 2003, 2004, 3002, 5008
Windows Integrity Errors	3001, 3002, 3003, 3004, 3010 and 3023
EMET Crash Logs	1 and 2
Windows Firewall Logs	2004, 2005, 2006, 2009, 2033
MSI Packages Installed	1022 and 1033
Windows Update Installed	2 and 19
Windows Service Manager Errors	7022, 7023, 7024, 7026, 7031, 7032, 7034
Group Policy Errors	1125, 1127, 1129
AppLocker and SRP Logs	865, 866, 867, 868, 882, 8003, 8004, 8006, 8007
Windows Update Errors	20, 24, 25, 31, 34, 35
Hotpatching Error	1009
Kernel Driver and Kernel Driver Signing Errors	5038, 6281, 219
Log Clearing	104 and 1102
Kernel Filter Driver	6
Windows Service Installed	7045

Table 1 of "Spotting the Adversary with Windows Event Log Monitoring" provides a very good overview of essential event IDs that should be monitored.

SANS

SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 113

Spotting the Adversary with Windows Event Log Monitoring

"Spotting the Adversary with Windows Event Log Monitoring" is the reference on event log monitoring, published by the NSA/CSS. This lengthy document (around 50 pages) is a must read for blue teams. It was last reviewed on July 16, 2015.

The following topics are covered in detail:

- Deployment
- Hardening Event Collectio
- Recommended Events to Collect.
- Event Log Retention
- Final Recommendations

Deployment will not only cover the configuration of Windows event logs but also centralization of these logs using Microsoft's publisher/subscriber model. It must be said that this is not the only way to centralize event logs. There are many other systems, for example, Splunk comes to mind.

https://cryptome.org/2014/01/nsa-windows-event.pdf

Table 1 of "Spotting the Adversary with Windows Event Log Monitoring" provides a very good overview of essential event IDs that should be monitored.

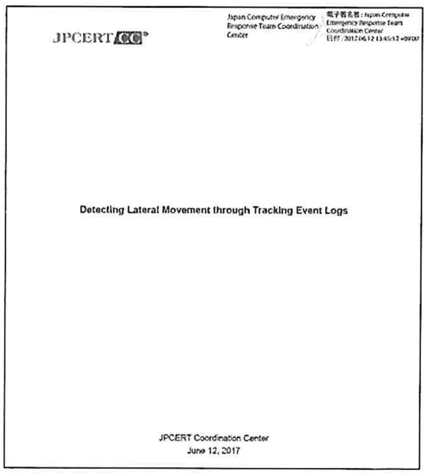
The event IDs themselves are not explained in much detail in this document, however. Let's illustrate this with event ID 4648. This ID is mentioned first in table 1 (under account and group activities), and second in table 9. From table 9, we know that this event is from the Security event log. The description of this event is "Account login with explicit credentials." However, the document does not provide more information about this event, and why it is important.

"This event is generated when a process attempts to log on an account by explicitly specifying that account's credentials. This most commonly occurs in batch-type configurations such as scheduled tasks, or when using the RUNAS command."

And there is also a user comment that explains in which circumstances the user saw this event.

From this information, we can indeed conclude that this is an important event ID to monitor: With single-sign-on, users on Windows don't have to provide explicit credentials to access a remote service. If this happens, it means that a user could not use single-sign-on. This can indicate an attack, where the adversary is impersonating another user.

Detecting Lateral Movement Through Tracking Event Logs (1)


JPCERT/CC

Japan Computer Emergency Response Team Coordination Center

Detecting Lateral Movement through Tracking Event Logs

JPCERT Coordination Center
June 12, 2017

Next to the NSA's interesting article on Windows event logs, there is a highly useful paper that was released by JPCert in June 2017

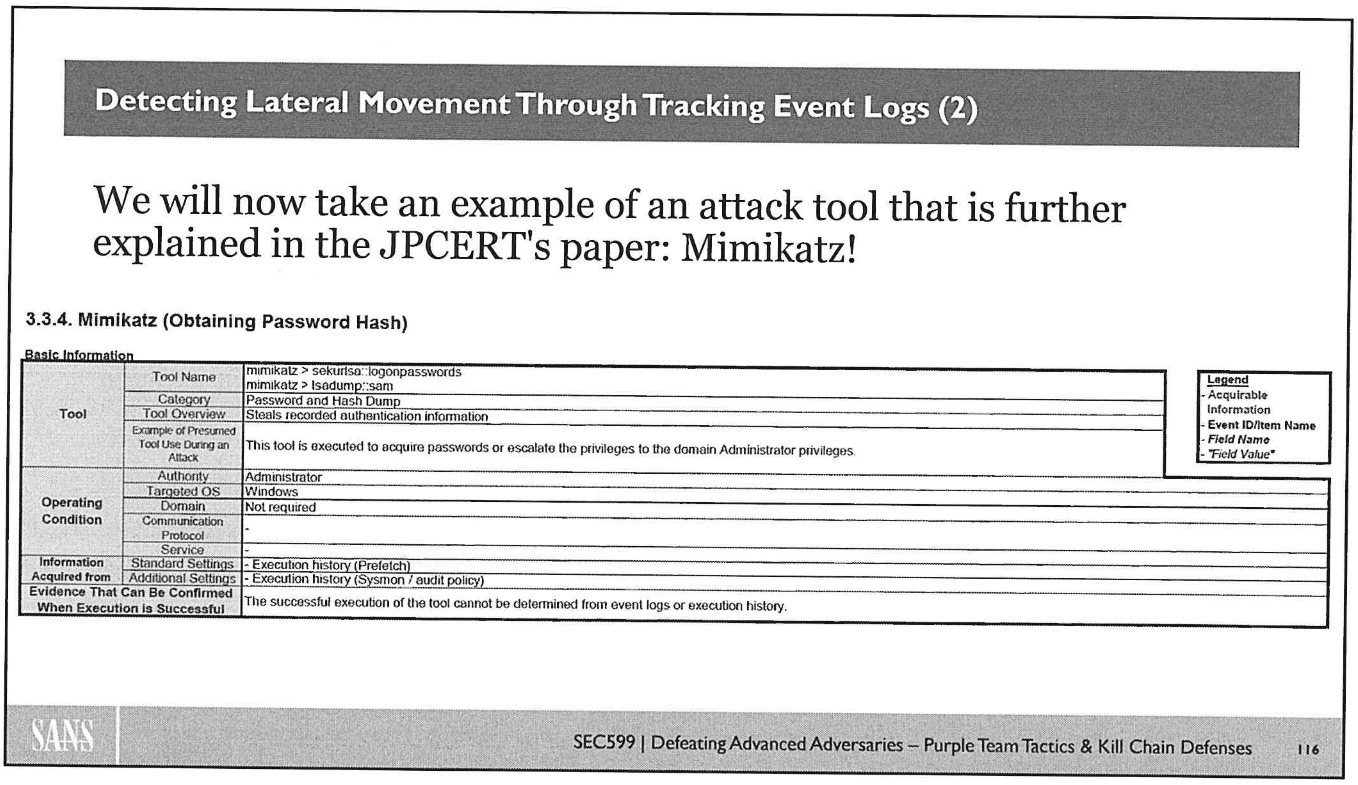
The document aims to provide a good overview of common attack techniques and tools and how they can be detected!

https://www.jpcert.or.jp/english/pub/sr/20170612ac-ir_research_en.pdf

Detecting Lateral Movement Through Tracking Event Logs (1)
Next to the NSA's interesting article on Windows event logs, there is a highly useful paper that was released by JPCert in June 2017. The document aims to provide a good overview of common attack techniques and tools and how they can be detected! We will zoom in on the document by walking through a specific example of such a tool: Mimikatz!

The full paper can be found at the following link:

https://www.jpcert.or.jp/english/pub/sr/20170612ac-ir_research_en.pdf



Detecting Lateral Movement Through Tracking Event Logs (2)

We will now take an example of an attack tool that is further explained in the JPCERT's paper: Mimikatz!

3.3.4. Mimikatz (Obtaining Password Hash)

Basic Information

Tool	Tool Name	mimikatz > sekurlsa::logonpasswords mimikatz > lsadump::sam
	Category	Password and Hash Dump
	Tool Overview	Steals recorded authentication information
	Example of Presumed Tool Use During an Attack	This tool is executed to acquire passwords or escalate the privileges to the domain Administrator privileges.
Operating Condition	Authority	Administrator
	Targeted OS	Windows
	Domain	Not required
	Communication Protocol	-
	Service	-
Information Acquired from	Standard Settings	- Execution history (Prefetch)
	Additional Settings	- Execution history (Sysmon / audit policy)
Evidence That Can Be Confirmed When Execution is Successful		The successful execution of the tool cannot be determined from event logs or execution history.

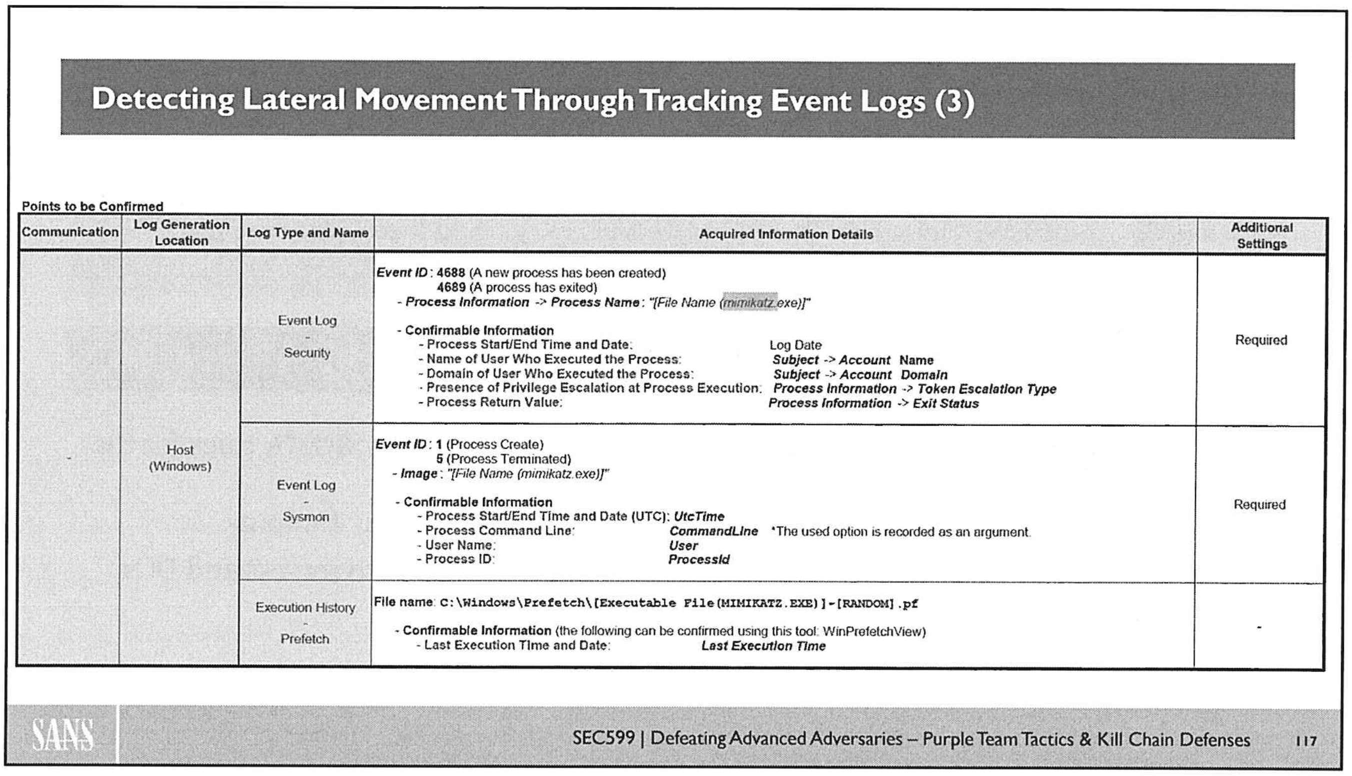
Legend
- Acquirable Information
- **Event ID/Item Name**
- ***Field Name***
- *"Field Value"*

SANS

SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 116

Detecting Lateral Movement Through Tracking Event Logs (2)
We will now take an example of an attack tool that is further explained in the JPCERT's paper: Mimikatz! In this first section, the paper will describe the typical purpose of the tool and the type of forensic artifacts that can be used to detect use of the tool.

© 2019 Erik Van Buggenhout & Stephen Sims



Detecting Lateral Movement Through Tracking Event Logs (3)

Points to be Confirmed

Communication	Log Generation Location	Log Type and Name	Acquired Information Details	Additional Settings
-	Host (Windows)	Event Log - Security	*Event ID*: **4688** (A new process has been created) **4689** (A process has exited) - ***Process Information*** -> ***Process Name***: *"[File Name (mimikatz.exe)]"* - **Confirmable Information** - Process Start/End Time and Date: Log Date - Name of User Who Executed the Process: ***Subject*** -> ***Account*** **Name** - Domain of User Who Executed the Process: ***Subject*** -> ***Account*** **Domain** - Presence of Privilege Escalation at Process Execution: ***Process Information*** -> ***Token Escalation Type*** - Process Return Value: ***Process Information*** -> ***Exit Status***	Required
		Event Log - Sysmon	*Event ID*: **1** (Process Create) **5** (Process Terminated) - ***Image***: *"[File Name (mimikatz.exe)]"* - **Confirmable Information** - Process Start/End Time and Date (UTC): ***UtcTime*** - Process Command Line: ***CommandLine*** *The used option is recorded as an argument. - User Name: ***User*** - Process ID: ***ProcessId***	Required
		Execution History - Prefetch	File name: `C:\Windows\Prefetch\[Executable File(MIMIKATZ.EXE)]-[RANDOM].pf` - **Confirmable Information** (the following can be confirmed using this tool: WinPrefetchView) - Last Execution Time and Date: ***Last Execution Time***	-

SANS

Detecting Lateral Movement Through Tracking Event Logs (3)
Below the initial table (as described in the previous slide), a more detailed section is included, where it is illustrated what event IDs are required to successfully detect use of the tools. In this example, you will notice the following event IDs are being used:

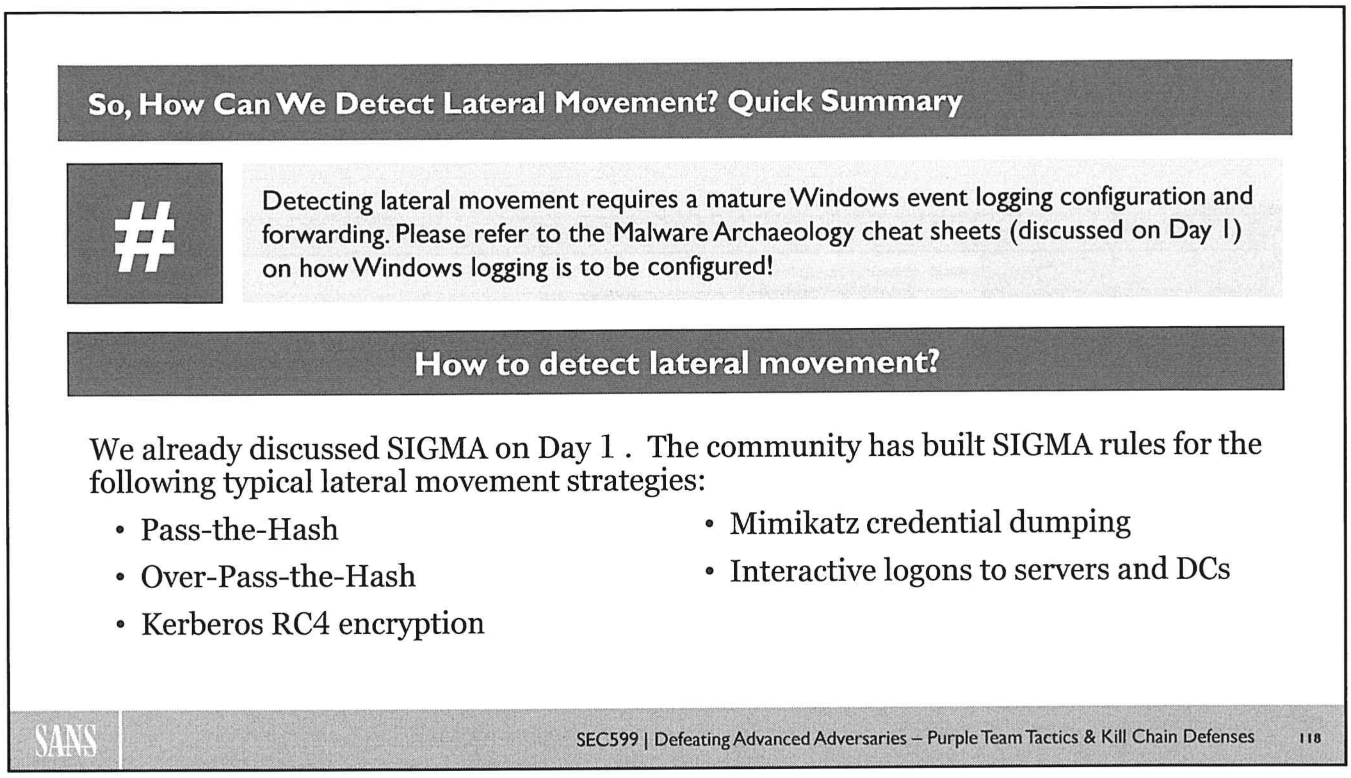
Windows event logs

- 4688: A new process has been created.
- 4689: A process has exited.

Sysmon

- 1: Process create
- 5: Process Terminated

Furthermore, it also uses the name of the tool (Mimikatz…). This is a rather basic control, as adversaries could perfectly rename the tool. We could, however, also monitor for use of the typical command line arguments passed to Mimikatz ("sekurlsa"…).



So, How Can We Detect Lateral Movement? Quick Summary

#

Detecting lateral movement requires a mature Windows event logging configuration and forwarding. Please refer to the Malware Archaeology cheat sheets (discussed on Day 1) on how Windows logging is to be configured!

How to detect lateral movement?

We already discussed SIGMA on Day 1 . The community has built SIGMA rules for the following typical lateral movement strategies:

- Pass-the-Hash
- Over-Pass-the-Hash
- Kerberos RC4 encryption
- Mimikatz credential dumping
- Interactive logons to servers and DCs

SANS

SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 118

So, How Can We Detect Lateral Movement? Quick Summary
Detecting lateral movement requires a mature Windows event logging configuration and forwarding. Please refer to the Malware Archaeology cheat sheets (discussed on Day 1) on how Windows logging is to be configured! Many organizations do not perform proper windows event log forwarding and, for example, only enable log domain controller logs. While this is a start, proper detection of lateral movement will require analysis of workstations and server logs as well.

We discussed SIGMA on Day 1 already, the community has built SIGMA rules for the following typical lateral movement strategies:

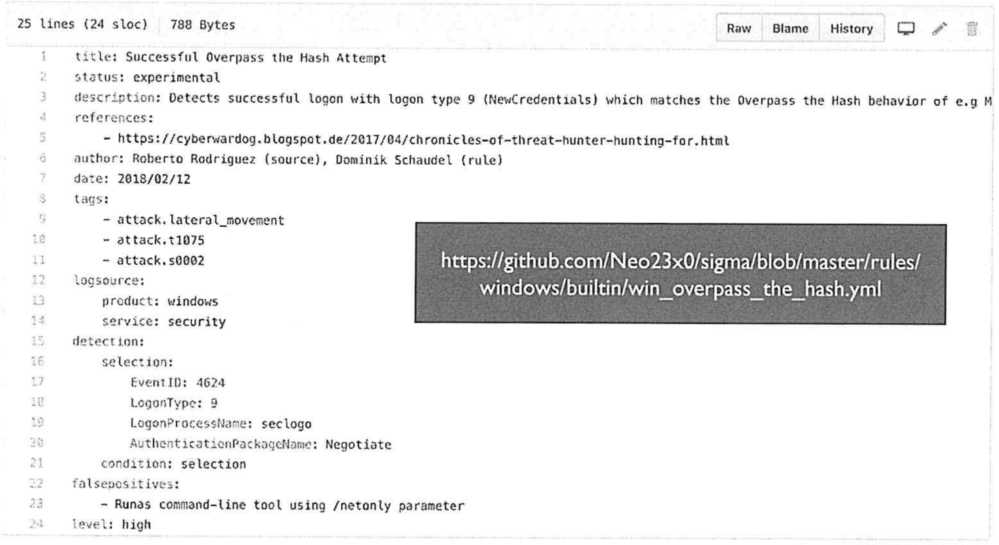
- Pass-the-Hash
- Over-Pass-the-Hash
- Kerberos RC4 encryption
- Mimikatz credential dumping
- Interactive logons to servers and DCs

Furthermore, CERT-EU (the European CERT) wrote an excellent whitepaper that further explains what specific events can be leveraged to detect lateral movement:

https://cert.europa.eu/static/WhitePapers/CERT-EU_SWP_17-002_Lateral_Movements.pdf

© 2019 Erik Van Buggenhout & Stephen Sims

SIGMA Example – Over-Pass-the-Hash



```
title: Successful Overpass the Hash Attempt
status: experimental
description: Detects successful logon with logon type 9 (NewCredentials) which matches the Overpass the Hash behavior of e.g M
references:
    - https://cyberwardog.blogspot.de/2017/04/chronicles-of-threat-hunter-hunting-for.html
author: Roberto Rodriguez (source), Dominik Schaudel (rule)
date: 2018/02/12
tags:
    - attack.lateral_movement
    - attack.t1075
    - attack.s0002
logsource:
    product: windows
    service: security
detection:
    selection:
        EventID: 4624
        LogonType: 9
        LogonProcessName: seclogo
        AuthenticationPackageName: Negotiate
    condition: selection
falsepositives:
    - Runas command-line tool using /netonly parameter
level: high
```

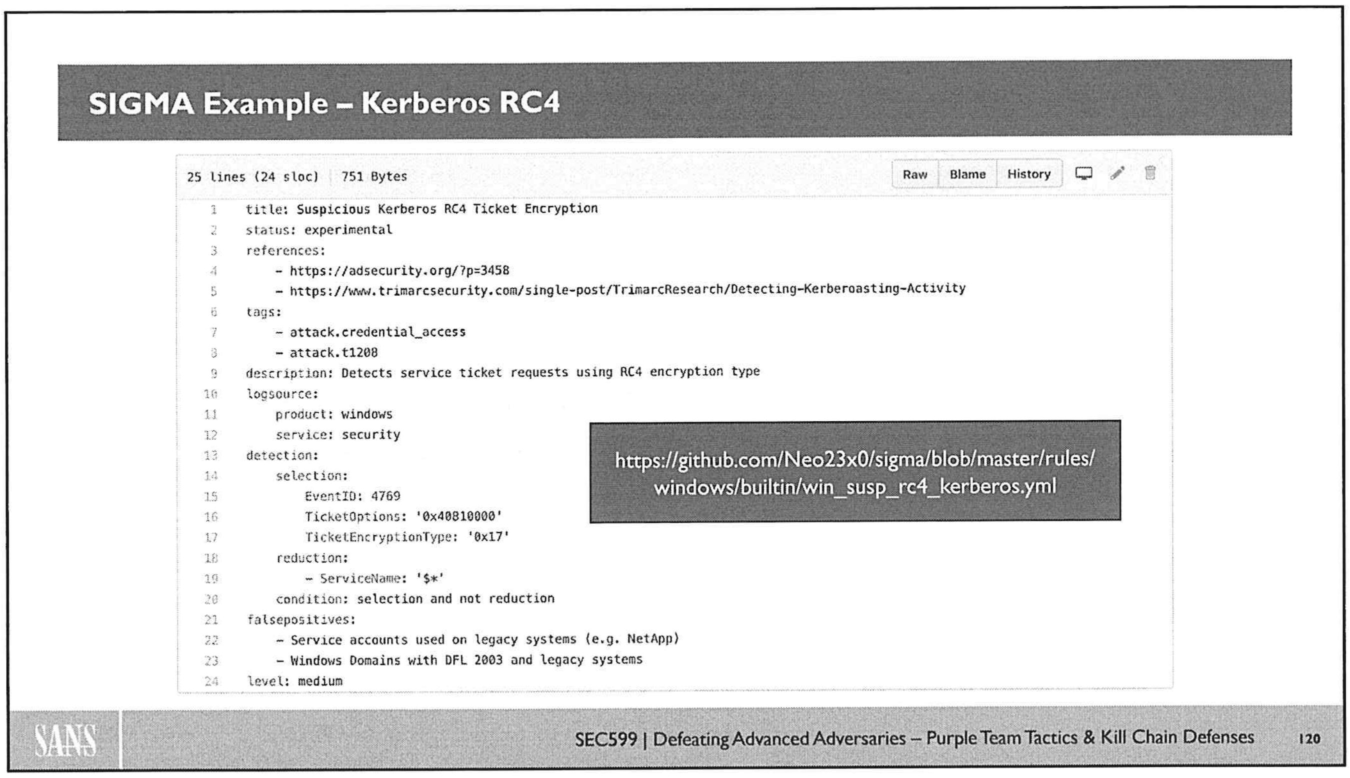
https://github.com/Neo23x0/sigma/blob/master/rules/windows/builtin/win_overpass_the_hash.yml

SIGMA Example – Over-Pass-the-Hash

The rule on the slide above was written by Dominik Schaudel, inspired by Roberto Rodriguez. It tries to detect adversaries using the previously described "Over-Pass-the-Hash" technique. It is purely based on the Windows logon event (event ID 4624) and looks for the following properties:

- LogonType should be 9 (NewCredentials).
- LogonProcessName should be "seclogo" (Secondary Login).
- AuthenticationPackageName should be "Negotiate".

The SIGMA rule can be found here:
https://github.com/Neo23x0/sigma/blob/master/rules/windows/builtin/win_overpass_the_hash.yml



SIGMA Example – Kerberos RC4

```
title: Suspicious Kerberos RC4 Ticket Encryption
status: experimental
references:
    - https://adsecurity.org/?p=3458
    - https://www.trimarcsecurity.com/single-post/TrimarcResearch/Detecting-Kerberoasting-Activity
tags:
    - attack.credential_access
    - attack.t1208
description: Detects service ticket requests using RC4 encryption type
logsource:
    product: windows
    service: security
detection:
    selection:
        EventID: 4769
        TicketOptions: '0x40810000'
        TicketEncryptionType: '0x17'
    reduction:
        - ServiceName: '$*'
    condition: selection and not reduction
falsepositives:
    - Service accounts used on legacy systems (e.g. NetApp)
    - Windows Domains with DFL 2003 and legacy systems
level: medium
```

https://github.com/Neo23x0/sigma/blob/master/rules/windows/builtin/win_susp_rc4_kerberos.yml

SIGMA Example – Kerberos RC4

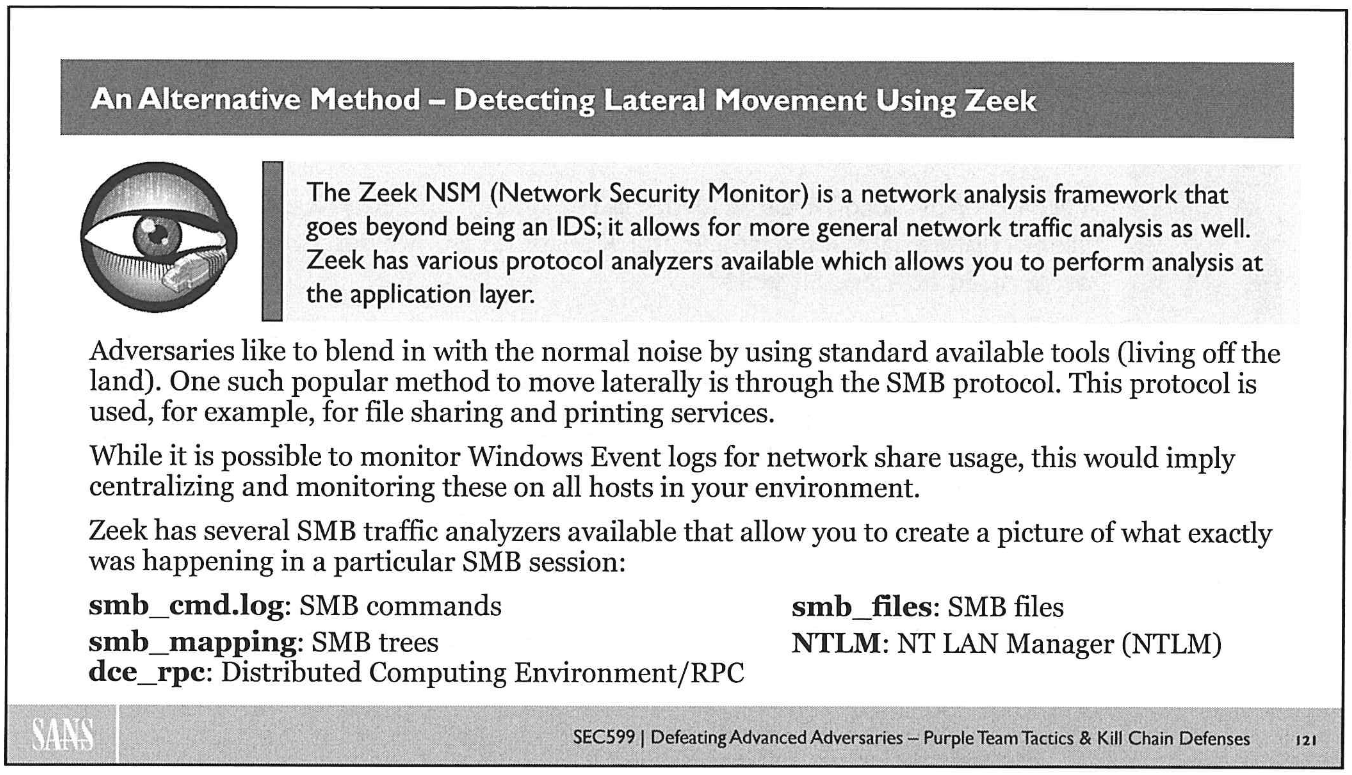
The rule on the slide above was written by Florian Roth. It tries to detect adversaries requesting Service Tickets using the RC4 encryption type. It is purely based on the Service Ticket request event (event ID 4769) and looks for the following properties:

- TicketOptions: 0x40810000
- TicketEncryptionType: 0x17 (RC4)
- The rule also ignores computer accounts (that start with $).

There's also a comment on known false positives, which include service accounts on legacy systems and of course older Windows systems.

The SIGMA rule can be found here:
https://github.com/Neo23x0/sigma/blob/master/rules/windows/builtin/win_susp_rc4_kerberos.yml

© 2019 Erik Van Buggenhout & Stephen Sims



An Alternative Method – Detecting Lateral Movement Using Zeek

The Zeek NSM (Network Security Monitor) is a network analysis framework that goes beyond being an IDS; it allows for more general network traffic analysis as well. Zeek has various protocol analyzers available which allows you to perform analysis at the application layer.

Adversaries like to blend in with the normal noise by using standard available tools (living off the land). One such popular method to move laterally is through the SMB protocol. This protocol is used, for example, for file sharing and printing services.

While it is possible to monitor Windows Event logs for network share usage, this would imply centralizing and monitoring these on all hosts in your environment.

Zeek has several SMB traffic analyzers available that allow you to create a picture of what exactly was happening in a particular SMB session:

smb_cmd.log: SMB commands
smb_mapping: SMB trees
dce_rpc: Distributed Computing Environment/RPC
smb_files: SMB files
NTLM: NT LAN Manager (NTLM)

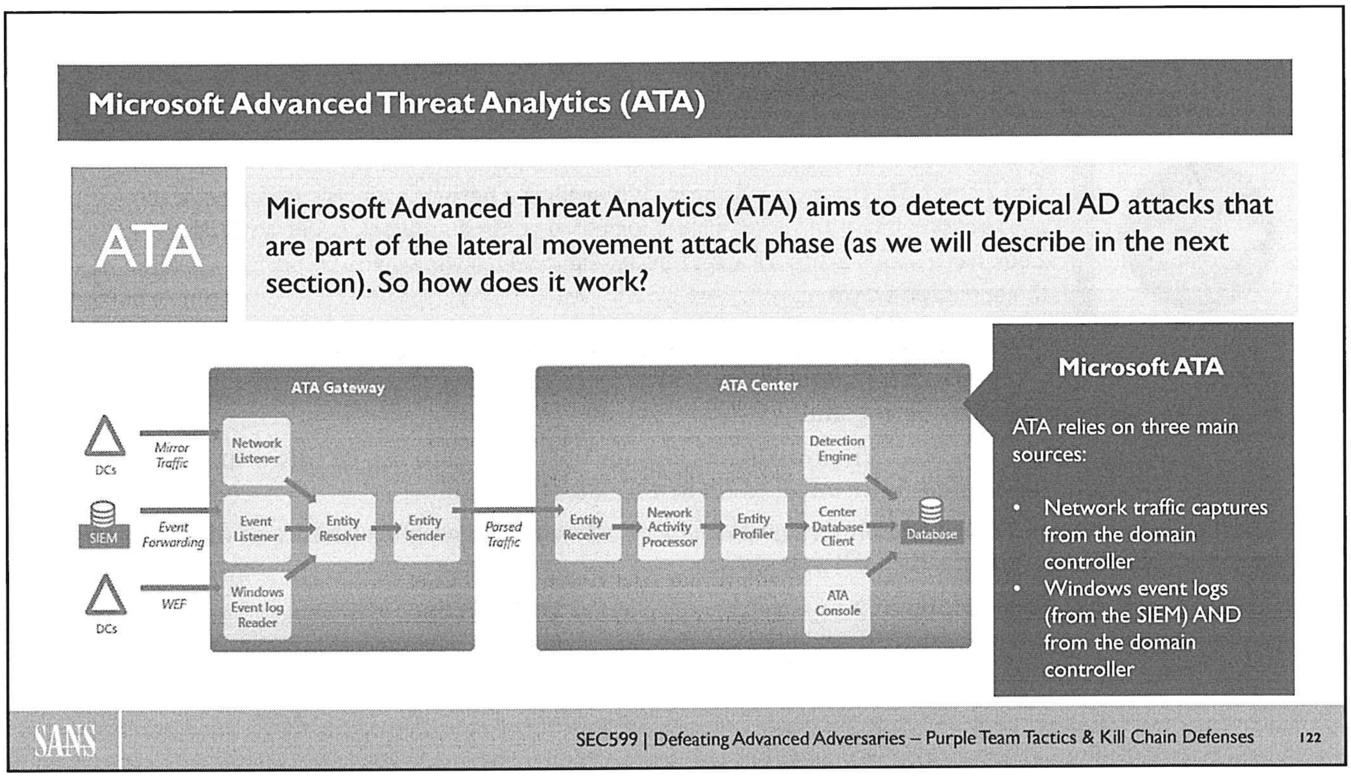
SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 121

An Alternative Method – Detecting Lateral Movement Using Zeek
In environments with proper segmentation and tapping points between network segments, NSM tools such as Zeek could be used in addition to analysis of the Windows event logs!

In order to move laterally, most attackers will use readily available tools rather than using self-created tools for further exploitation (living off the land). Using standard Windows utilities will help the attacker to blend in with the noise generated by normal operations. The SMB protocol on Windows is, therefore, a tool greatly appreciated by adversaries. Zeek has several SMB traffic analyzers available that allow you to create a picture of what exactly was happening in a particular SMB session:

- smb_cmd.log: SMB commands
- smb_files: SMB files
- smb_mapping: SMB trees
- NTLM: NT LAN Manager (NTLM)
- dce_rpc: Distributed Computing Environment/RPC

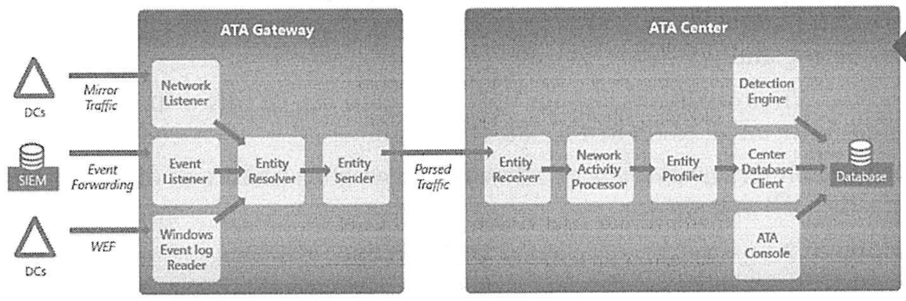
In order to activate these analyzers, you'll need to ensure these are enabled in the configuration file.



Microsoft Advanced Threat Analytics (ATA)

ATA

Microsoft Advanced Threat Analytics (ATA) aims to detect typical AD attacks that are part of the lateral movement attack phase (as we will describe in the next section). So how does it work?



Microsoft ATA

ATA relies on three main sources:

- Network traffic captures from the domain controller
- Windows event logs (from the SIEM) AND from the domain controller

Microsoft Advanced Threat Analytics (ATA)
Microsoft ATA aims to detect typical Active Directory attacks that take place in your environment. It is mostly focused on the lateral movement phase of the Kill Chain, where adversaries attempt to escalate privileges and obtain administrative access over the domain.

So how does ATA work?

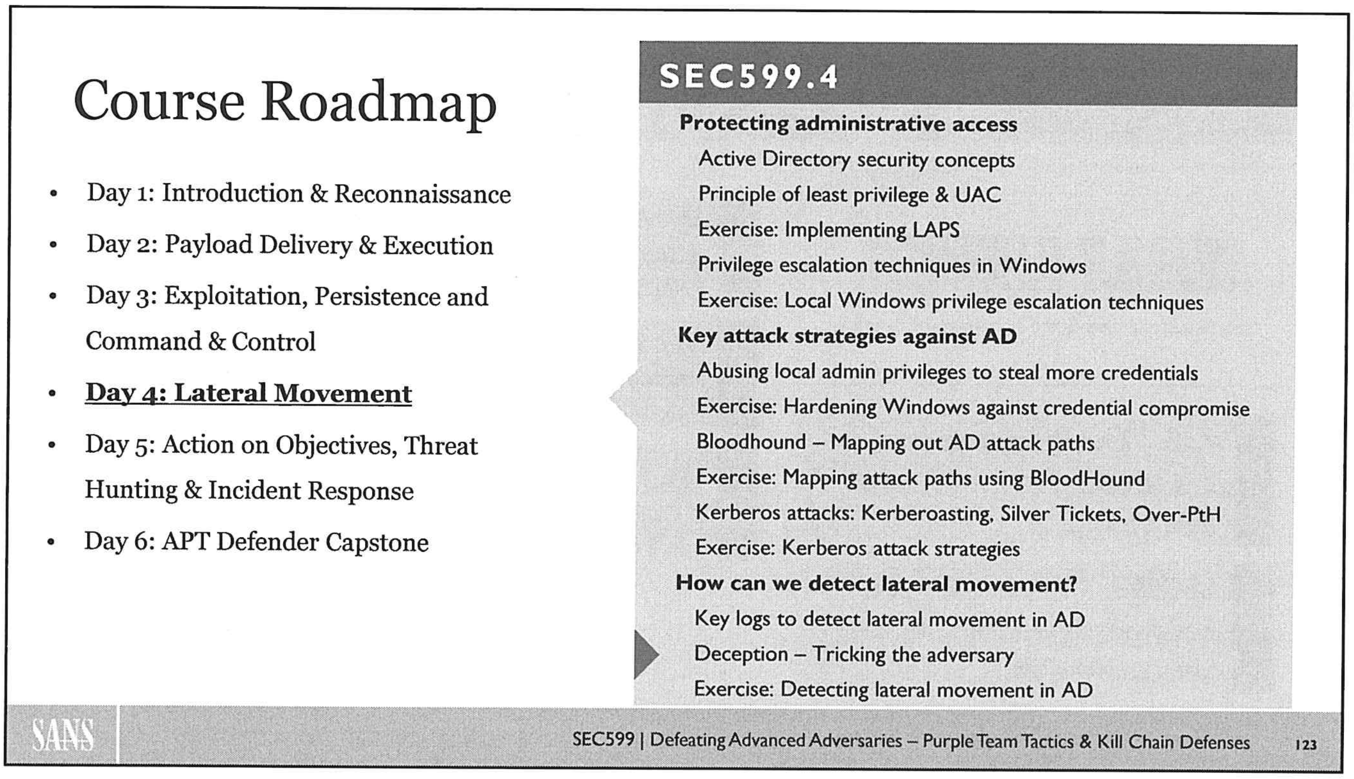
ATA relies on three main sources of information for its analytics:

- It captures network traffic on the domain controller(s)
- It processes Windows event logs from the SIEM and the local domain controller event log.

ATA offers a number of interesting detection options, although depending on the type of logs that it consumes, it cannot successfully detect all attacks. If the system, for example, does not receive event logs from local Windows workstations and servers, it will be hard to spot lateral movement by local account reuse.

As a final note, we would like to indicate that ATA is a commercial add-on that is not available "by default" on Windows.

© 2019 Erik Van Buggenhout & Stephen Sims



Course Roadmap

- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- **Day 4: Lateral Movement**
- Day 5: Action on Objectives, Threat Hunting & Incident Response
- Day 6: APT Defender Capstone

SEC599.4

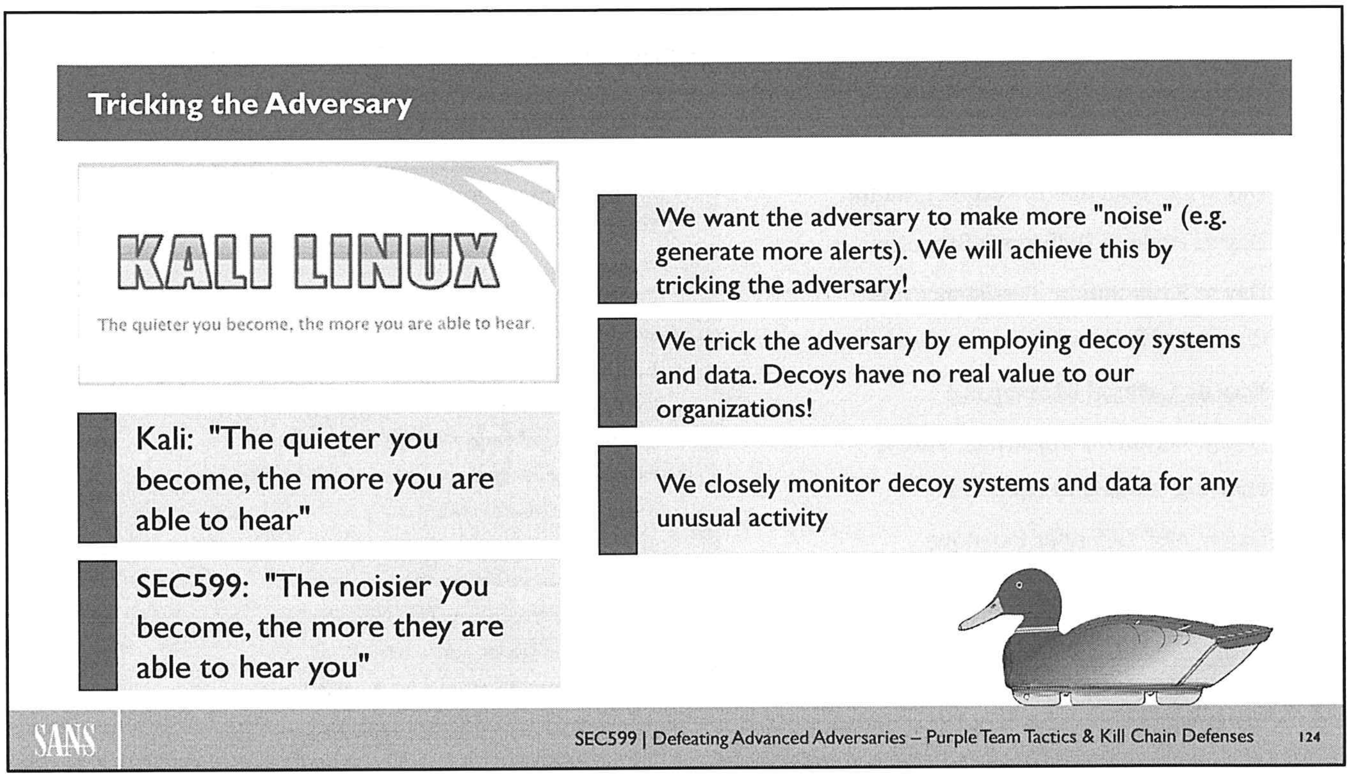
Protecting administrative access
- Active Directory security concepts
- Principle of least privilege & UAC
- Exercise: Implementing LAPS
- Privilege escalation techniques in Windows
- Exercise: Local Windows privilege escalation techniques

Key attack strategies against AD
- Abusing local admin privileges to steal more credentials
- Exercise: Hardening Windows against credential compromise
- Bloodhound – Mapping out AD attack paths
- Exercise: Mapping attack paths using BloodHound
- Kerberos attacks: Kerberoasting, Silver Tickets, Over-PtH
- Exercise: Kerberos attack strategies

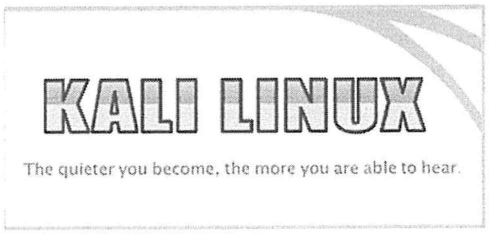
How can we detect lateral movement?
- Key logs to detect lateral movement in AD
- ▶ Deception – Tricking the adversary
- Exercise: Detecting lateral movement in AD

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 123

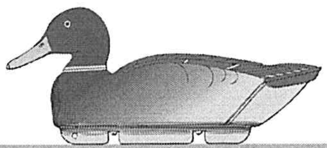
This page intentionally left blank.



Tricking the Adversary



- Kali: "The quieter you become, the more you are able to hear"
- SEC599: "The noisier you become, the more they are able to hear you"
- We want the adversary to make more "noise" (e.g. generate more alerts). We will achieve this by tricking the adversary!
- We trick the adversary by employing decoy systems and data. Decoys have no real value to our organizations!
- We closely monitor decoy systems and data for any unusual activity



SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 124

Tricking the Adversary
Kali Linux is a well-known Linux distribution for penetration testers.

Its motto is:

"The quieter you become, the more you are able to hear"

As a motto to "tricking the adversary", we will use:

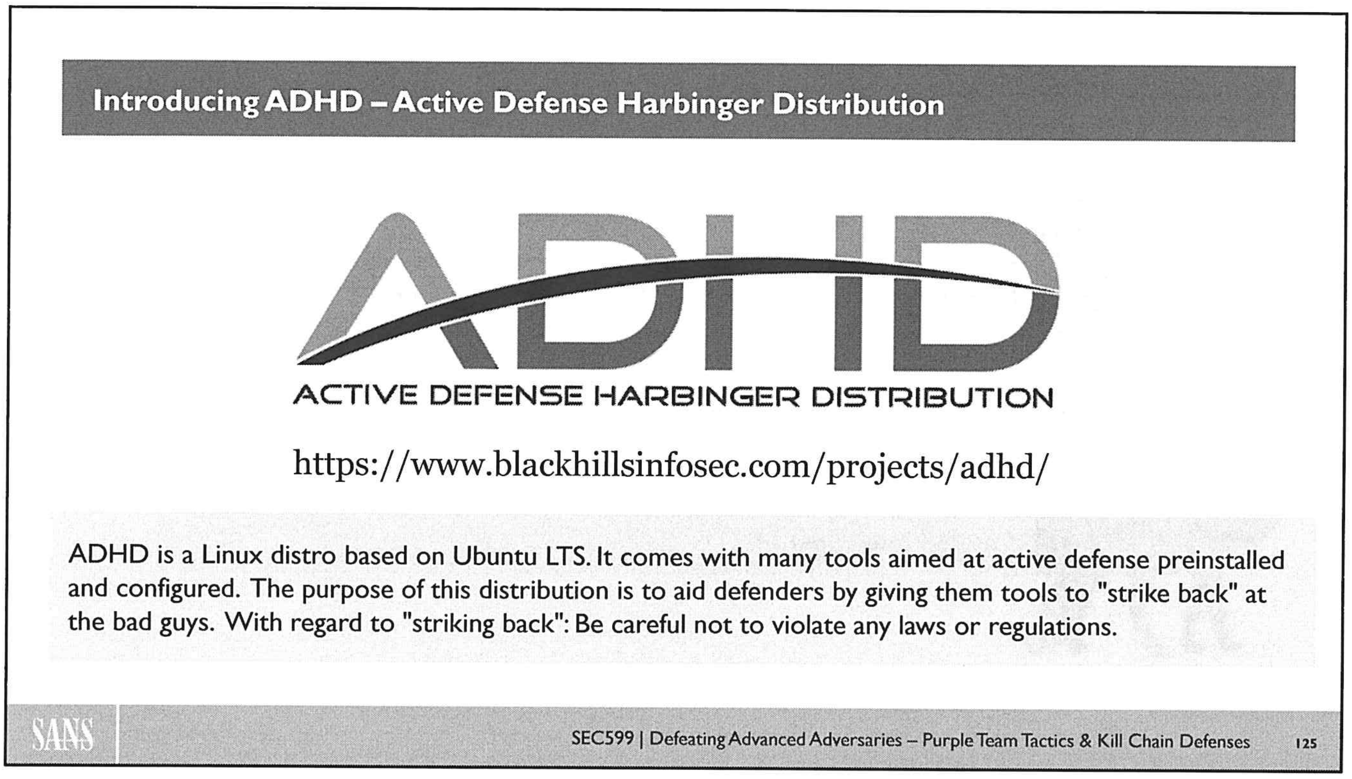
"The noisier you become, the more they are able to hear you"

By tricking the adversary, we want him to make mistakes that will produce noise (events) that we can detect. The idea behind "tricking the adversary" is that we want the adversary to make more "noise."

The adversary wants to be as quiet as possible in our networks and systems (e.g. not generate events and certainly not alerts), so that we will not detect them and that they have more time to perform their malicious deeds. We want the adversary to make more "noise", e.g. generate more alerts, and we will achieve this by tricking the adversary.

The way will achieve this "tricking of the adversary" is by deploying and using decoy systems and data. Decoys are fake systems and fake data. They have no real value to our corporation. We try to make these decoys enticing to the attackers so that they will find and try to compromise these systems and access this data. This will not only slow down the attackers by having them waste their time on systems and data that has no real value, but we will also closely monitor decoy systems and data for any unusual activity.

This enhanced monitoring activity for decoys is our trap: The snare with which we hope to catch attackers!



Introducing ADHD – Active Defense Harbinger Distribution

ACTIVE DEFENSE HARBINGER DISTRIBUTION

https://www.blackhillsinfosec.com/projects/adhd/

ADHD is a Linux distro based on Ubuntu LTS. It comes with many tools aimed at active defense preinstalled and configured. The purpose of this distribution is to aid defenders by giving them tools to "strike back" at the bad guys. With regard to "striking back": Be careful not to violate any laws or regulations.

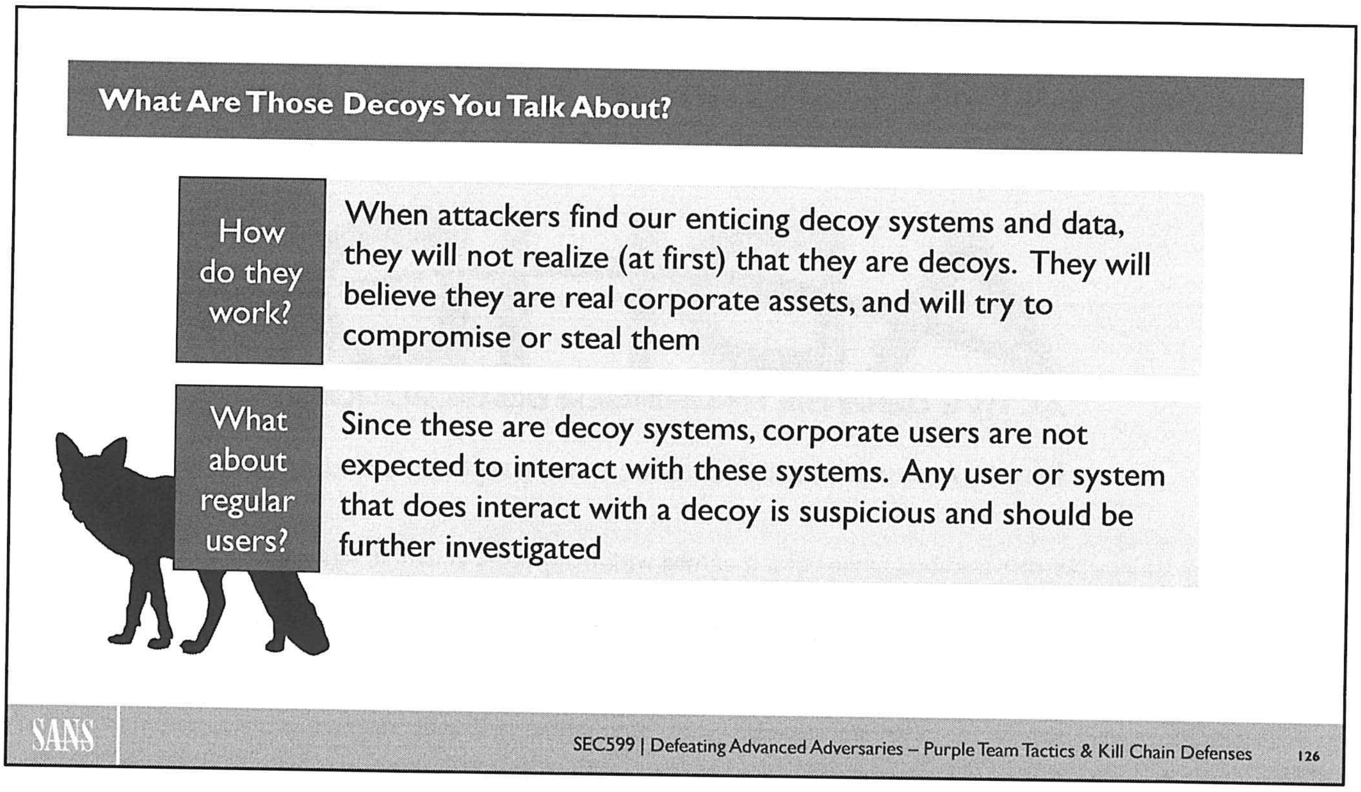
SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 125

Introducing ADHD – Active Defense Harbinger Distribution
Cyber deception strategies are often used as part of an overall "Active Defense" strategy, where defenders are trying to defeat adversaries by actively engaging them using tricks and dedicated systems setup.

An interesting toolkit for active defense is ADHD (Active Defense Harbinger Distribution). ADHD is a Linux distro based on Ubuntu LTS. It comes with many tools aimed at active defense preinstalled and configured. The purpose of this distribution is to aid defenders by giving them tools to "strike back" at the bad guys. With regard to "striking back": Be careful not to violate any laws or regulations.

We will not go through the entire ADHD tool suite during this course, but we will, for example, use HoneyBadger during an upcoming exercise.

You can find more information on ADHD here: https://www.blackhillsinfosec.com/projects/adhd/.



What Are Those Decoys You Talk About?

How do they work? When attackers find our enticing decoy systems and data, they will not realize (at first) that they are decoys. They will believe they are real corporate assets, and will try to compromise or steal them

What about regular users? Since these are decoy systems, corporate users are not expected to interact with these systems. Any user or system that does interact with a decoy is suspicious and should be further investigated

SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 126

What Are Those Decoys You Talk About?
It should not be obvious that a decoy system is a decoy, and not a real system, although not too much time and effort should be made to make a decoy system look like a real system.

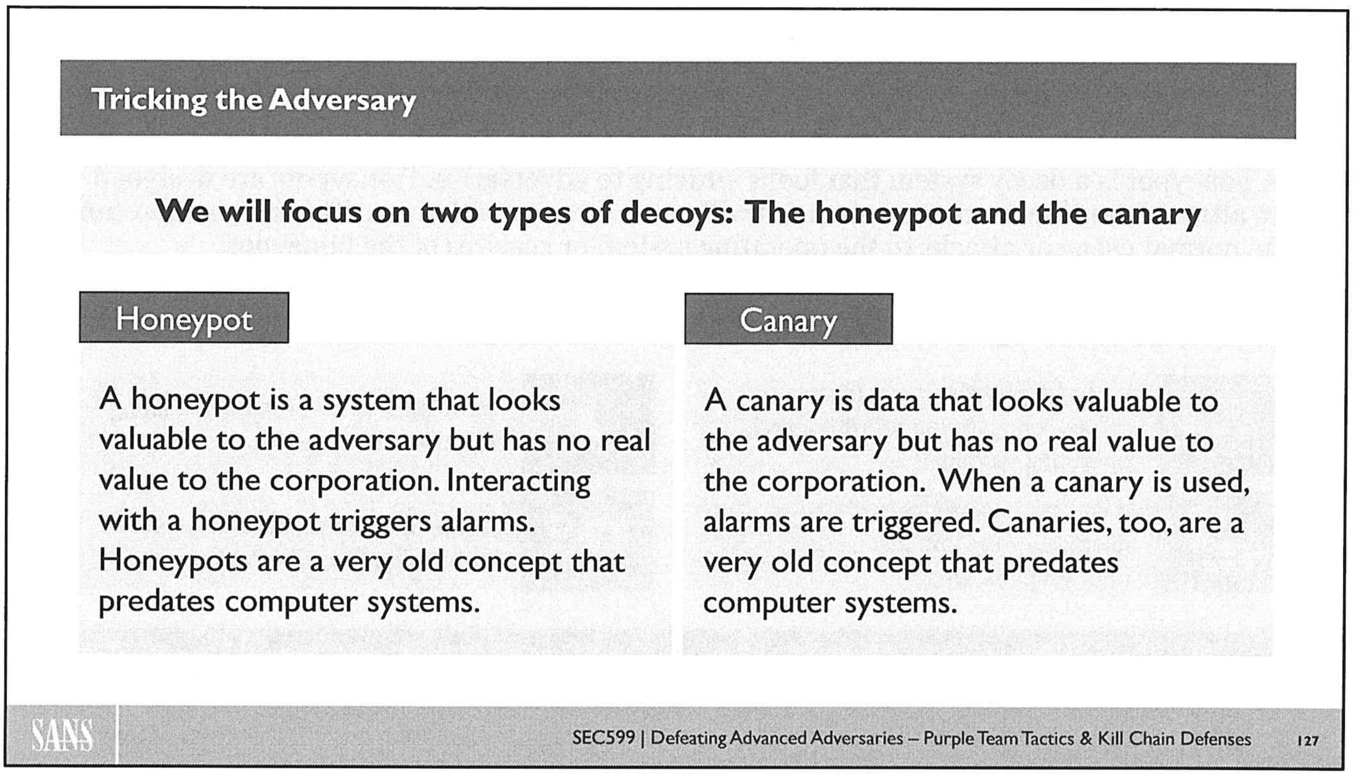
When attackers find our enticing decoy systems and data, they will not realize (at first) that they are decoys. That's why it is important that they are not obviously a decoy system, otherwise, the attackers will not be slowed down or tricked into generating alerts.

They will believe that the decoy systems and data are real corporate assets, and if we make them look like valuable corporate assets, attackers will try to compromise or steal them. Disguising a decoy as a valuable corporate asset requires subtlety. We don't want to make it too obvious that it is very valuable, otherwise the attackers might become suspicious. The examples we give here in this book are often too obvious, but that is to make the point clear.

Since these are decoy systems and data, corporate users are not expected to interact with these systems or access this data.

Any user or system that does interact with a decoy is suspicious and should be further investigated. It will, of course, happen that users will discover these assets and try to access them. We will consider these alerts as false positives, but nevertheless, it should be pointed out that these users should be talked to, just to figure out why they wanted to access these systems or data.

© 2019 Erik Van Buggenhout & Stephen Sims



Tricking the Adversary

We will focus on two types of decoys: The honeypot and the canary

Honeypot

A honeypot is a system that looks valuable to the adversary but has no real value to the corporation. Interacting with a honeypot triggers alarms. Honeypots are a very old concept that predates computer systems.

Canary

A canary is data that looks valuable to the adversary but has no real value to the corporation. When a canary is used, alarms are triggered. Canaries, too, are a very old concept that predates computer systems.

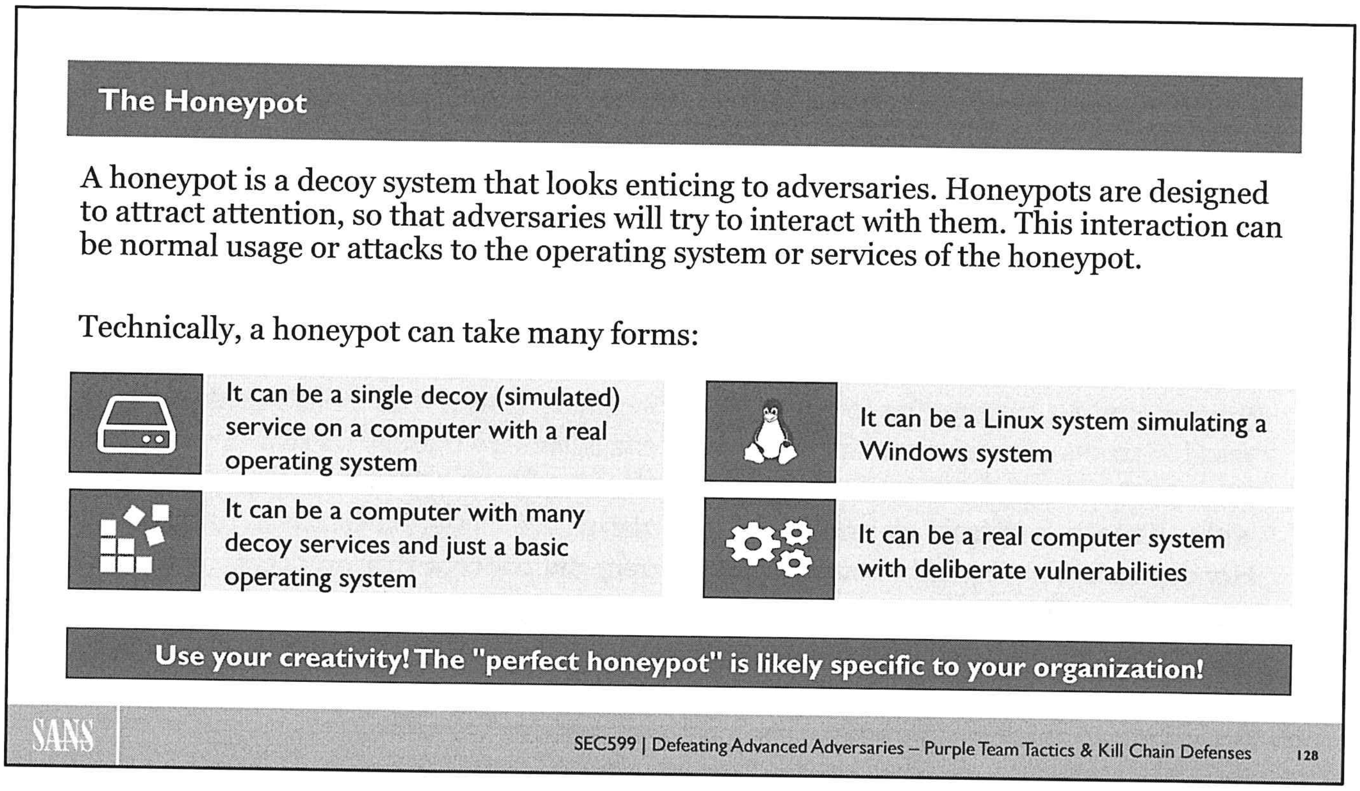
SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 127

Tricking the Adversary
The two types of decoy we will focus on in this chapter are:

- The honeypot
- The canary

A honeypot is a system that looks valuable to the adversary but has no real value to the corporation. Interacting with a honeypot triggers alarms. Honeypots are a very old concept that predates computer systems. They have been used by hunters and soldiers to ambush the adversary.

A canary is data that looks valuable to the adversary but has no real value to the corporation. When a canary is used, alarms are triggered. Canaries, too, are a very old concept that predates computer systems. They have been used by hunters and soldiers to find the adversary.



The Honeypot

A honeypot is a decoy system that looks enticing to adversaries. Honeypots are designed to attract attention, so that adversaries will try to interact with them. This interaction can be normal usage or attacks to the operating system or services of the honeypot.

Technically, a honeypot can take many forms:

- It can be a single decoy (simulated) service on a computer with a real operating system
- It can be a computer with many decoy services and just a basic operating system
- It can be a Linux system simulating a Windows system
- It can be a real computer system with deliberate vulnerabilities

Use your creativity! The "perfect honeypot" is likely specific to your organization!

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 128

The Honeypot
A honeypot is a decoy system that looks enticing to adversaries. Honeypots are designed to attract attention so that adversaries will try to interact with them. This interaction can be normal usage or attacks to the operating system or services of the honeypot.

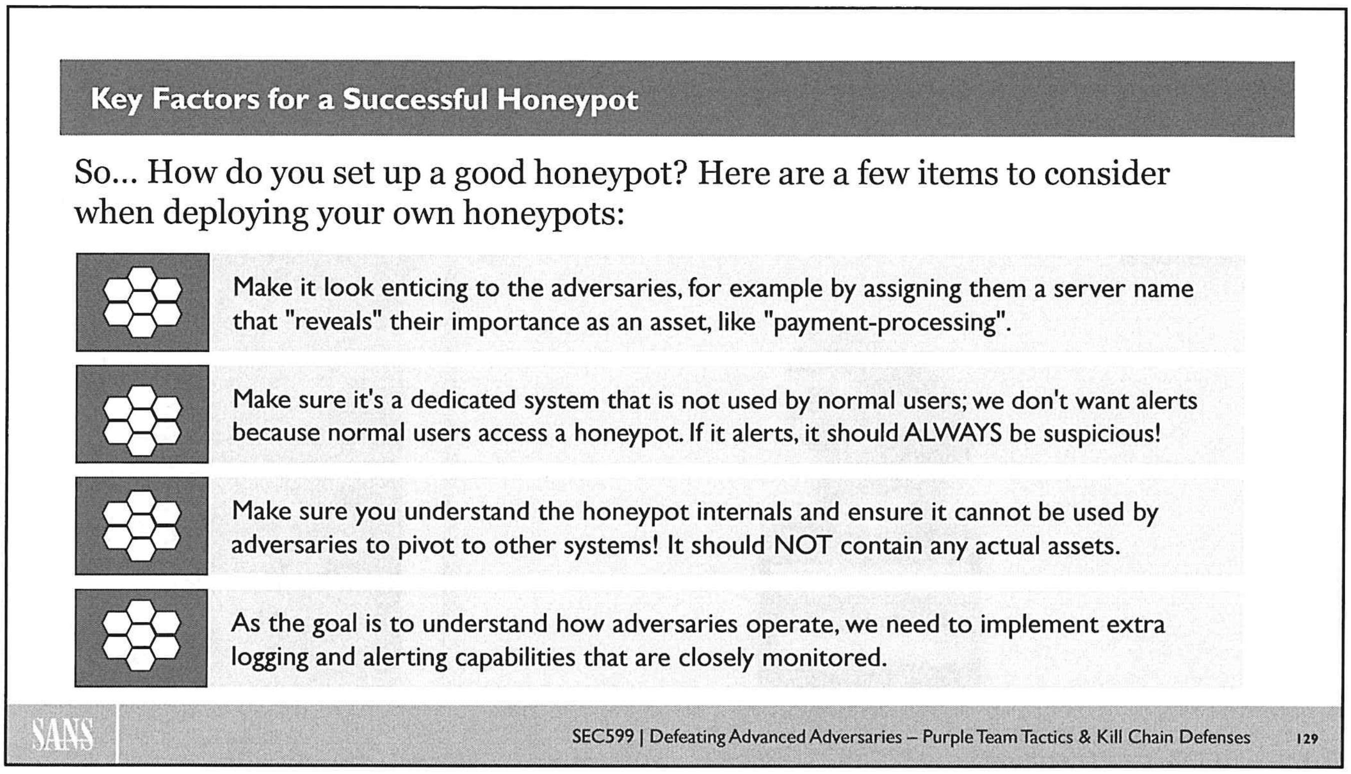
The name honeypot refers to children tales of bears liking honey: Bears like honey, and they will go the extraordinary length to obtain honey from beehives. A honeypot is the ultimate attraction for a bear: It's a container filled with pure honey, ready to be consumed.

Technically, a honeypot can take many forms:

- It can be a single decoy (simulated) service on a computer with a real operating system: We just install a computer with a normal operating system and install a service on it to attract adversaries. This can be a real service like a webserver or a program that simulates the basic features of a webserver.
- It can be a computer with many decoy services and just a basic operating system: We can install an operating system with the minimum amount of services, and then replace those services with decoy services.
- It can be a Linux system simulating a Windows system: For example, we install SAMBA and configure it to resemble a Windows file server as closely as possible.
- It can be a real computer system with deliberate vulnerabilities.
- ...

Don't forget to use your creativity! The "perfect honeypot" will most likely be different from organization to organization. If you want to catch targeted attacks, you should do more than just setting up a "generic" honeypot! Think about your crown jewels and make something that actually looks like that!

© 2019 Erik Van Buggenhout & Stephen Sims



Key Factors for a Successful Honeypot

So... How do you set up a good honeypot? Here are a few items to consider when deploying your own honeypots:

- Make it look enticing to the adversaries, for example by assigning them a server name that "reveals" their importance as an asset, like "payment-processing".
- Make sure it's a dedicated system that is not used by normal users; we don't want alerts because normal users access a honeypot. If it alerts, it should ALWAYS be suspicious!
- Make sure you understand the honeypot internals and ensure it cannot be used by adversaries to pivot to other systems! It should NOT contain any actual assets.
- As the goal is to understand how adversaries operate, we need to implement extra logging and alerting capabilities that are closely monitored.

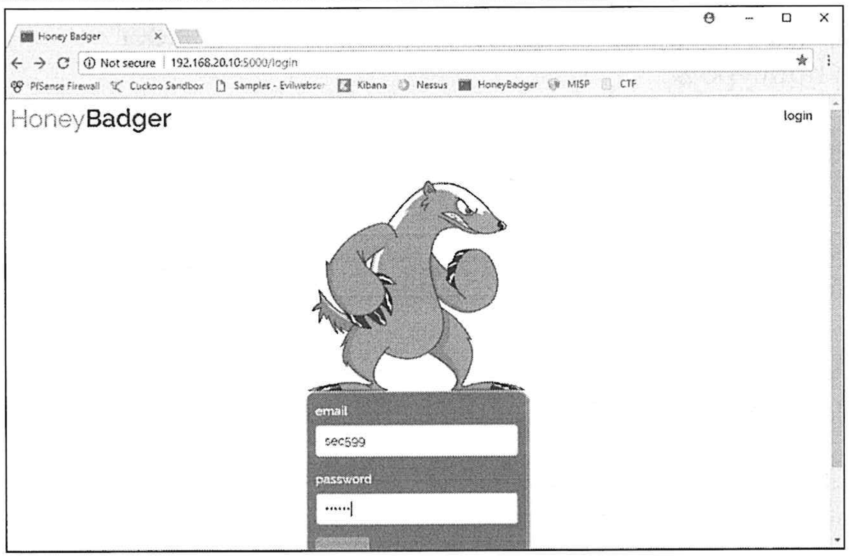
SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 129

Key Factors for a Successful Honeypot
We use honeypots to trick the adversary: We want the honeypot to be discovered by an adversary and look enticing so that the adversary will try to interact with it. By interacting with it, we can detect the presence of adversaries in our corporate infrastructure.

To achieve these goals, it is important that honeypots:

- Look enticing to the adversaries, for example, by assigning them a server name that "reveals" their importance as an asset, like "payment-processing". Honeypots have to be discovered by the adversaries when they search through our networks, and by assigning them enticing names, we try to increase the chances that adversaries will try to access them. But this must not be overdone, otherwise, the attackers will know it is a honeypot. For example, if you call your honeypot server "super-secret-file-server", that's probably overdoing it.
- Are not used (interacted with) by normal users: We don't want alerts because normal users try to access a honeypot. Honeypots should not have anything on them that normal users would need to use. We want to avoid false positive alerts as much as possible. That said, if a normal user does access a honeypot, it should be investigated.
- Have extra monitoring and alerting capabilities that are closely monitored. Honeypots should only generate alerts when adversaries access them, therefore, all alerts should be investigated.
- Should not contain corporate assets: We do not want to attract adversaries with real corporate assets like data and confidential documents. If you want to make a honeypot more enticing with documents, use fake confidential documents.
- Should not be able to be used by adversaries to pivot to other systems: Particular care should be taken that a honeypot does not contain tools and services or have access to other systems so that it can be used as a pivot.

Some Example Honeypots – HoneyBadger (ADHD)



HoneyBadger is part of the ADHD framework. It consists of a part of source code that can be added to an existing website, which will track visitor IP addresses and automatically perform GeoIP localization. It comes with a central admin interface for easy management and follow-up!

We will use it in an upcoming lab!

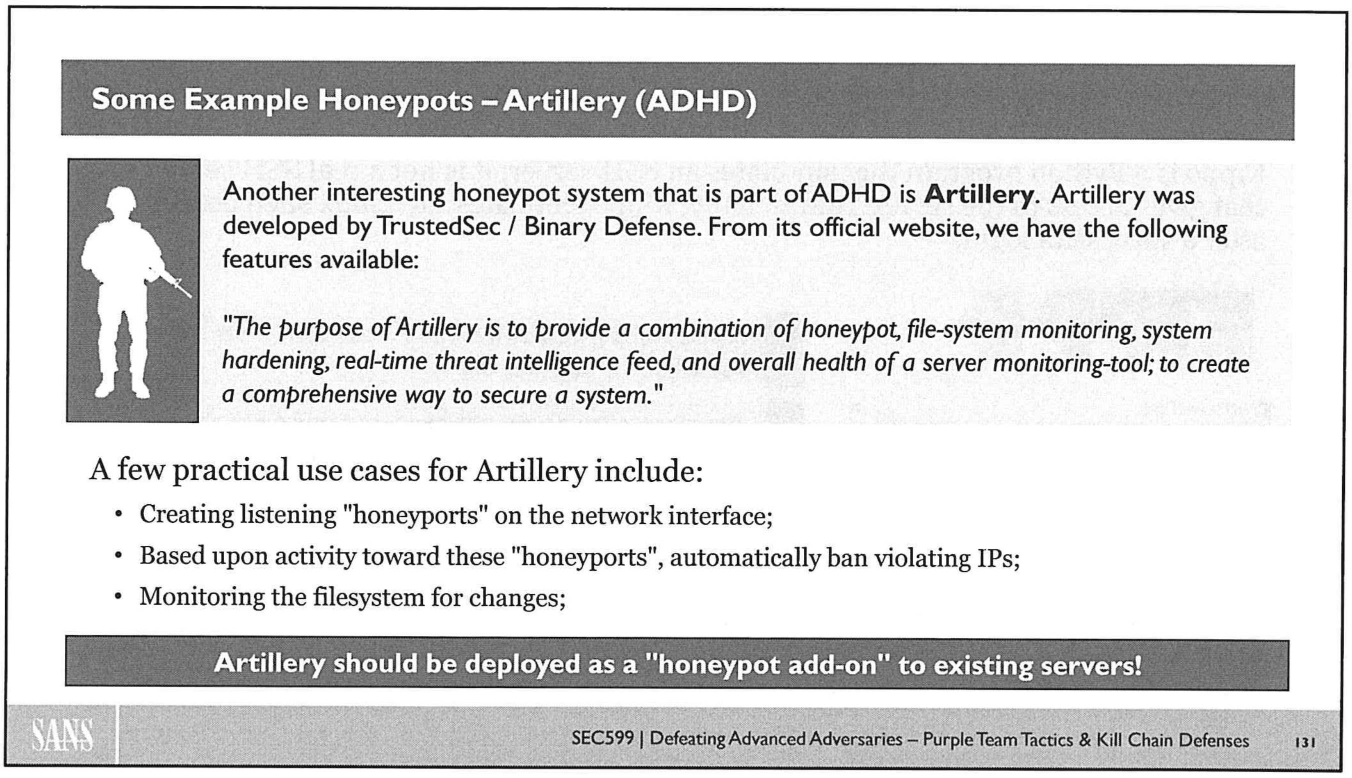
Some Example Honeypots – HoneyBadger (ADHD)
In the previous slide, we discussed setting up a fake administrative web page to lure adversaries to attempt authentication. But how we can effectively follow up on visits? We could implement our own piece of tracking code, but there are many frameworks already available for this purpose!

An interesting application to do this is "HoneyBadger". HoneyBadger provides an easy-to-use solution; it is part of the Active Defense Harbinger Distribution (ADHD). It consists of two core components:

- A part of source code that can be added to an existing website, which will track visitor IP addresses and automatically perform GeoIP localization.
- A central management console where you can follow up on "beacons".

You can find more information on HoneyBadger here: http://adhdproject.github.io/#!Tools/HoneyBadger.md. We will use it in an upcoming lab!

© 2019 Erik Van Buggenhout & Stephen Sims



Some Example Honeypots – Artillery (ADHD)

Another interesting honeypot system that is part of ADHD is **Artillery**. Artillery was developed by TrustedSec / Binary Defense. From its official website, we have the following features available:

"The purpose of Artillery is to provide a combination of honeypot, file-system monitoring, system hardening, real-time threat intelligence feed, and overall health of a server monitoring-tool; to create a comprehensive way to secure a system."

A few practical use cases for Artillery include:

- Creating listening "honeyports" on the network interface;
- Based upon activity toward these "honeyports", automatically ban violating IPs;
- Monitoring the filesystem for changes;

Artillery should be deployed as a "honeypot add-on" to existing servers!

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 131

Some Example Honeypots – Artillery (ADHD)

Another interesting honeypot system that is part of ADHD is Artillery. Artillery was developed by TrustedSec / Binary Defense. From its official website, we have the following features available:

"The purpose of Artillery is to provide a combination of honeypot, file-system monitoring, system hardening, real-time threat intelligence feed, and overall health of a server monitoring-tool; to create a comprehensive way to secure a system."

You can find the official documentation here: http://adhdproject.github.io/#!Tools/Artillery.md

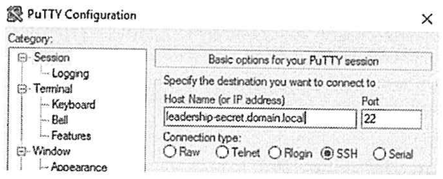
Artillery is not your typical honeypot system, as it wasn't designed to be a "stand-alone honeypot". Instead, it should be added to a high-value target server that it can protect. A few practical use cases for Artillery include:

- Creating fake listening ports on the target server (so-called "Honeyports").
- Upon activity toward these fake ports, it can automatically alert and ban violating IP addresses.
- Monitor the filesystem of the target server for changes to important files.

Some Example Honeypots – Kippo Fake SSH

Kippo is a Python program that simulates an SSH server; it is not a real SSH server that gives access to the server. Additionally, Kippo simulates the Linux shell obtained after a successful login.

KIPPO



Specify the destination you want to connect to
Host Name (or IP address): leadership-secret.domain.local
Port: 22
Connection type: Raw Telnet Rlogin [✓ SSH] Serial

- Kippo will log all activities, and these can be replayed to see what the attacker did
- Actual commands are not executed: For example, Kippo will download files for the attacker, but not execute them
- It will fake malfunctioning when executing a downloaded file

Kippo is one of the most well-known honeypots out there. A newer evolution (based upon Kippo) is "Cowrie", which is developed by Michel Oosterhof (https://github.com/micheloosterhof/cowrie). Cowrie is also part of ADHD!

Some Example Honeypots – Kippo Fake SSH
In this second honeypot example, we implement a fake SSH service. This can be done with Kippo, a honeypot service that simulates an SSH server. It is written in Python and is not a real SSH server that gives access to the server.

Kippo can be installed on a normal server, and we recommend using the normal SSH port (port 22) for Kippo. This means that to use Kippo on a Linux machine, the real SSH service should be put on another port or disabled. Kippo should also be able to run on a Windows machine, but this is only recommended if you have SSH servers running on your corporate machines. Otherwise, a single Windows machine with SSH server will stand out to the attackers, and they could realize that this is, in fact, a honeypot.

After a user logs on to Kippo via the normal SSH logon procedure, the user will be presented with a classic Linux command-line shell. This is not a real shell (bash for example), but it is simulated by Kippo.

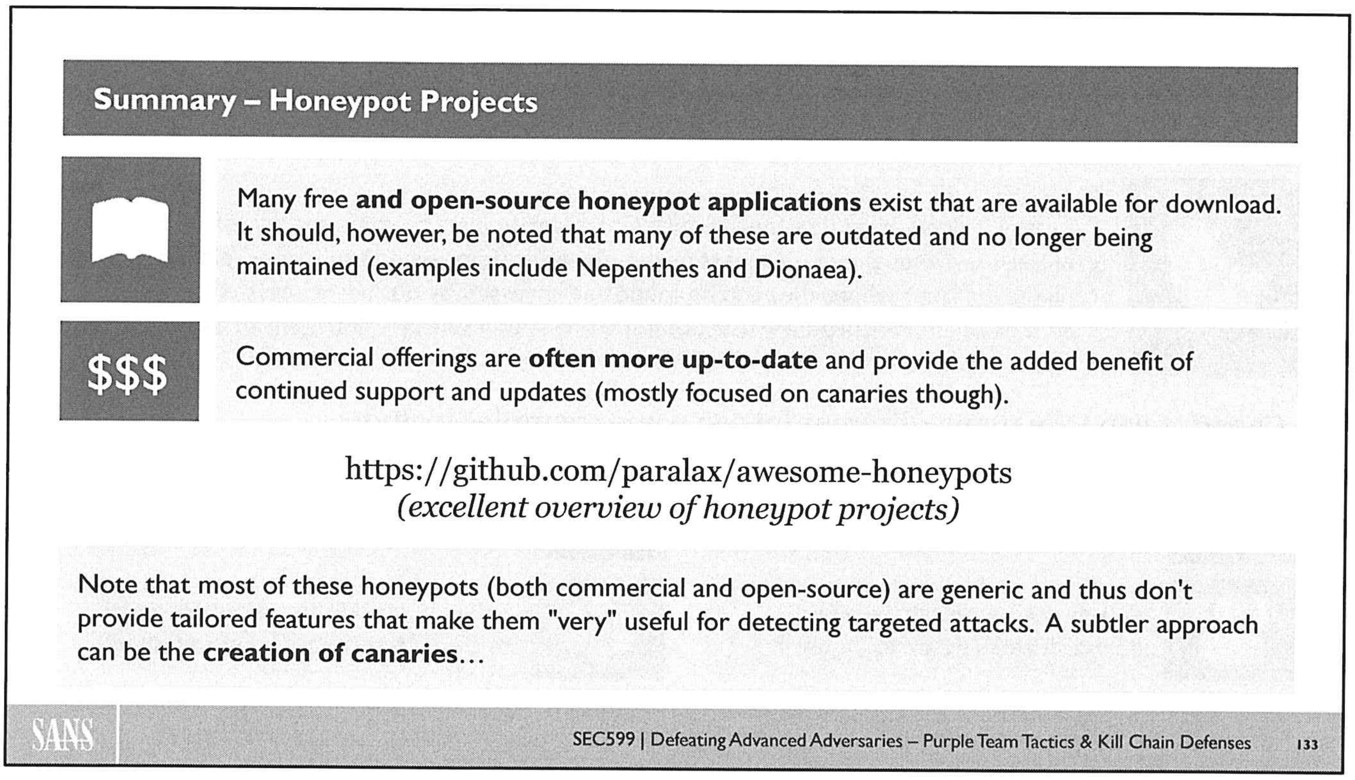
The commands that are typed by the user are also simulated by Kippo, to make it look like a real shell. For example, the attacker can issue a wget or curl command to download his attack tools on the server. Kippo will simulate this by effectively downloading the files and storing them for later review by the Kippo administrators.

Kippo will simulate the presence of the downloaded file in the current directory, but when the attacker tries to run the tool he/she downloaded, Kippo will not execute the tool but present an error message that tries to persuade the attacker that the file was corrupted during the download.

Kippo will log all activities, and these can be replayed to see what the attacker did. For use as a honeypot in our corporate network, Kippo must be configured to raise alerts when logons are performed.

Kippo is one of the most well-known honeypots out there. A newer evolution (based upon Kippo) is "Cowrie", which is developed by Michel Oosterhof (https://github.com/cowrie/cowrie).

© 2019 Erik Van Buggenhout & Stephen Sims


Summary – Honeypot Projects

Many free **and open-source honeypot applications** exist that are available for download. It should, however, be noted that many of these are outdated and no longer being maintained (examples include Nepenthes and Dionaea).

$$$

Commercial offerings are **often more up-to-date** and provide the added benefit of continued support and updates (mostly focused on canaries though).

https://github.com/paralax/awesome-honeypots
(excellent overview of honeypot projects)

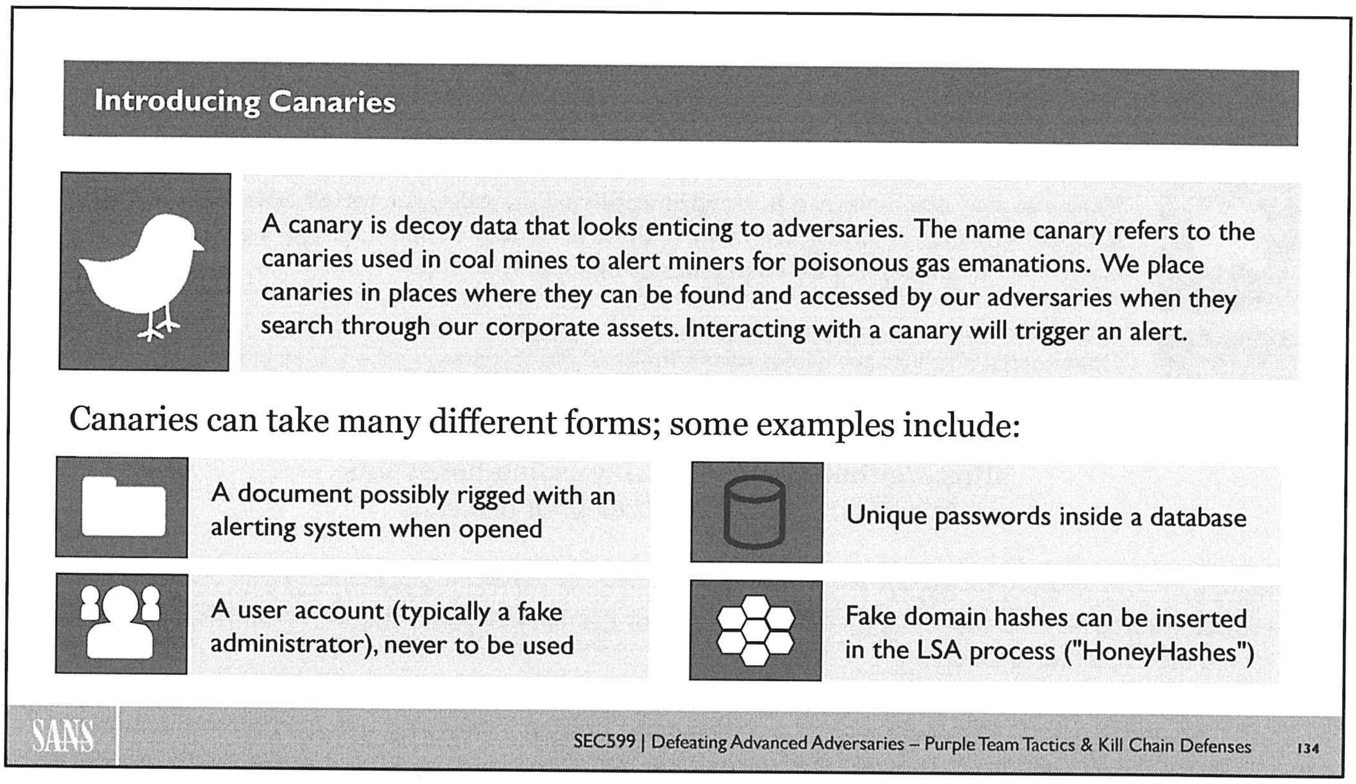
Note that most of these honeypots (both commercial and open-source) are generic and thus don't provide tailored features that make them "very" useful for detecting targeted attacks. A subtler approach can be the **creation of canaries**...

Available Honeypot Projects
There are a lot of free and open-source honeypot applications available for download. Not all of them are properly designed and safe to use in a corporate environment.

Many of these open-source honeypot applications are outdated or no longer maintained. Research into honeypots and development of honeypot applications was popular when Lance Spitzner published his book "Honeypots: Tracking Hackers" in 2003, but over the years, interest has faded a bit. This has resulted in many open-source honeypot projects being abandoned or no longer actively maintained. Examples of these are the well-known Nepenthes and Dionaea open-source honeypots. These days, commercial honeypot offerings are more up-to-date, we will see this when we discuss canaries.

An interesting overview of current honeypot projects can be found at https://github.com/paralax/awesome-honeypots.

Many of these honeypots are not suitable for our purposes: They are designed as "research" honeypots to research new malware and new zero-day exploits. They are not designed with corporate attackers in mind. Furthermore, most of these honeypots (both commercial and open-source) are generic and thus don't provide tailored features that make them "very" useful for detecting targeted attacks. A subtler approach can be the creation of canaries...



Introducing Canaries

A canary is decoy data that looks enticing to adversaries. The name canary refers to the canaries used in coal mines to alert miners for poisonous gas emanations. We place canaries in places where they can be found and accessed by our adversaries when they search through our corporate assets. Interacting with a canary will trigger an alert.

Canaries can take many different forms; some examples include:

- A document possibly rigged with an alerting system when opened
- A user account (typically a fake administrator), never to be used
- Unique passwords inside a database
- Fake domain hashes can be inserted in the LSA process ("HoneyHashes")

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 134

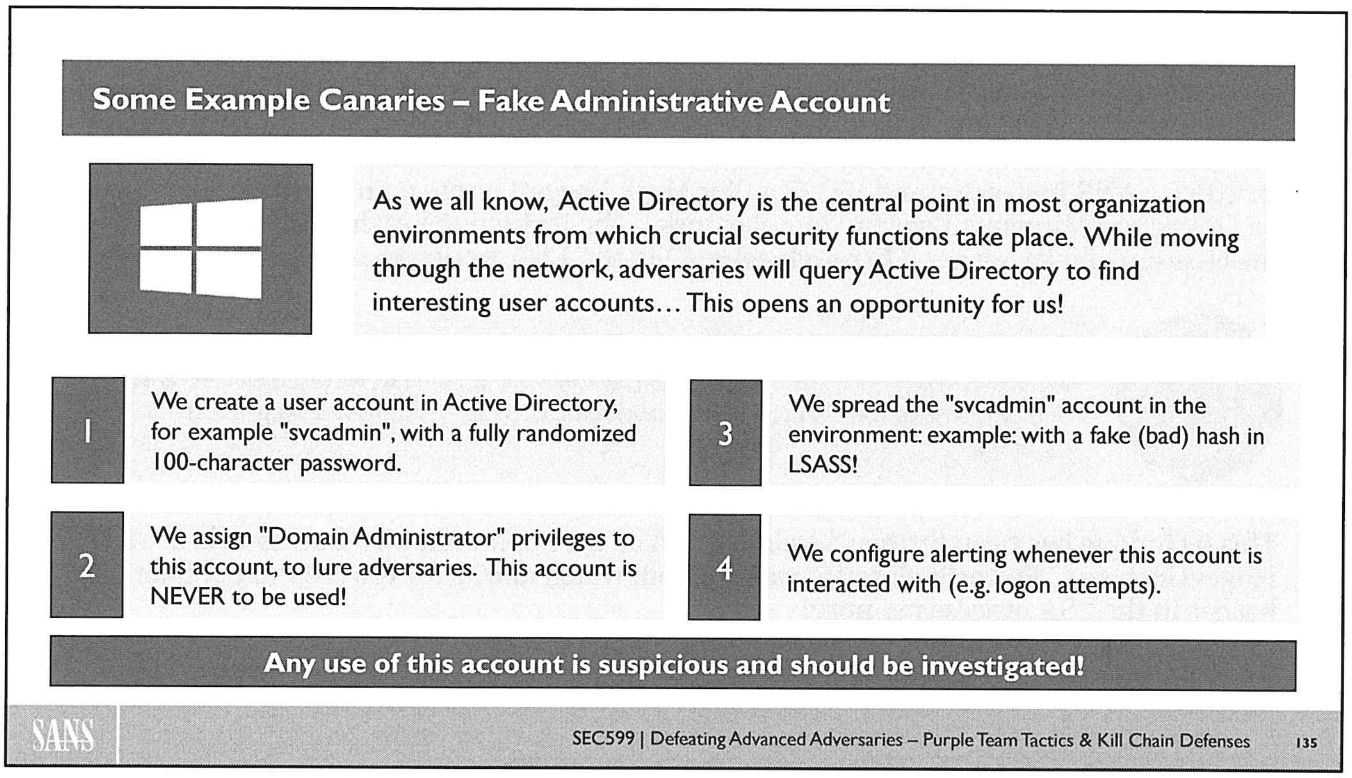
Introducing Canaries
Another popular method to trick adversaries is canaries. A canary is decoy data that looks enticing to adversaries. We place canaries in places where they can be found and accessed by our adversaries when they search through our corporate assets. Interacting with a canary will trigger an alert. The name canary refers to the canaries used in coal mines to alert miners for poisonous gas emanations.

Canaries can take many different forms; some examples include:

- A document possibly rigged with an alerting system that generates an alert when opened.
- A user account (typically a fake administrator), that is never to be used by corporate staff.
- Unique passwords inside a database that are never used as real passwords.
- Fake domain hashes can be inserted in the LSA process ("HoneyHashes").
- …

We will discuss a few of these techniques in the next few slides and use them in a few labs.

© 2019 Erik Van Buggenhout & Stephen Sims



Some Example Canaries – Fake Administrative Account

As we all know, Active Directory is the central point in most organization environments from which crucial security functions take place. While moving through the network, adversaries will query Active Directory to find interesting user accounts... This opens an opportunity for us!

1 We create a user account in Active Directory, for example "svcadmin", with a fully randomized 100-character password.

2 We assign "Domain Administrator" privileges to this account, to lure adversaries. This account is NEVER to be used!

3 We spread the "svcadmin" account in the environment: example: with a fake (bad) hash in LSASS!

4 We configure alerting whenever this account is interacted with (e.g. logon attempts).

Any use of this account is suspicious and should be investigated!

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 135

Some Example Canaries – Fake Administrative Account

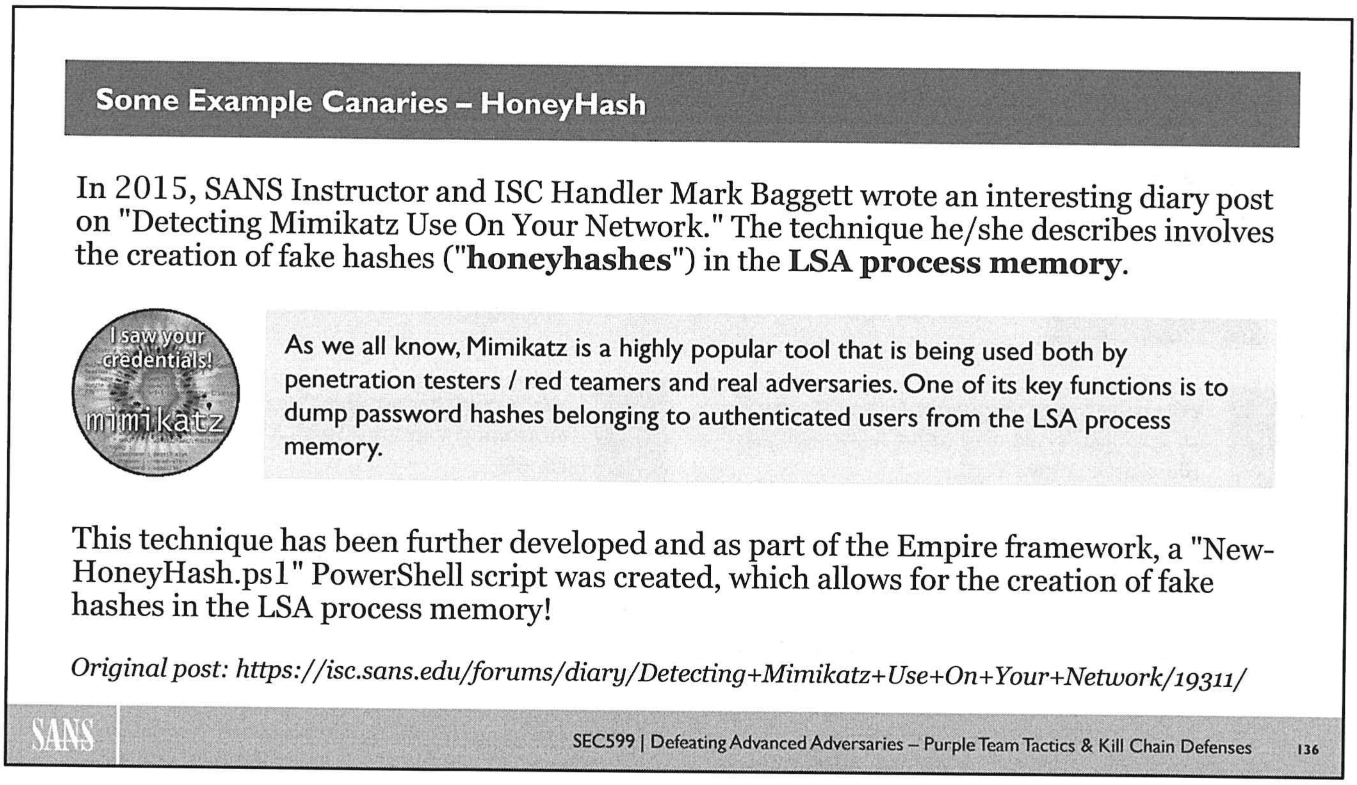
As we all know, Active Directory is the central point in most organization environments from which crucial security functions take place. While moving through the network, adversaries will query Active Directory to find interesting user accounts... This opens up an opportunity for us!

We can create a canary by adding a user account in Active Directory that looks enticing to our potential attackers, for example, "svcadmin" or any other account name that might look important in your corporate environment and would be attractive to attackers to compromise.

We assign "Domain Administrator" privileges to this account to lure adversaries. This account is NEVER to be used! We subsequently spread the "svcadmin" account in the environment:

- With a fake (bad) hash in LSASS on workstations.
- Document the account somewhere on the "Intranet" with a bad password.
- Write the account with a bad password in a fake script somewhere.
- ...

We configure alerting when events are created for this user account: Whenever this account is used (for example logon attempts) we want to be warned. We do this by configuring alerts in our monitoring systems each time events are created by Windows for this particular account.



Some Example Canaries – HoneyHash

In 2015, SANS Instructor and ISC Handler Mark Baggett wrote an interesting diary post on "Detecting Mimikatz Use On Your Network." The technique he/she describes involves the creation of fake hashes ("**honeyhashes**") in the **LSA process memory**.

As we all know, Mimikatz is a highly popular tool that is being used both by penetration testers / red teamers and real adversaries. One of its key functions is to dump password hashes belonging to authenticated users from the LSA process memory.

This technique has been further developed and as part of the Empire framework, a "New-HoneyHash.ps1" PowerShell script was created, which allows for the creation of fake hashes in the LSA process memory!

Original post: https://isc.sans.edu/forums/diary/Detecting+Mimikatz+Use+On+Your+Network/19311/

SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 136

Some Example Canaries – HoneyHash
In 2015, SANS Instructor & ISC Handler Mark Baggett wrote an interesting diary post on "Detecting Mimikatz Use On Your Network". The technique he/she describes involves the creation of fake hashes ("honeyhashes") in the LSA process memory. You can read Mark's original ISC diary post here:

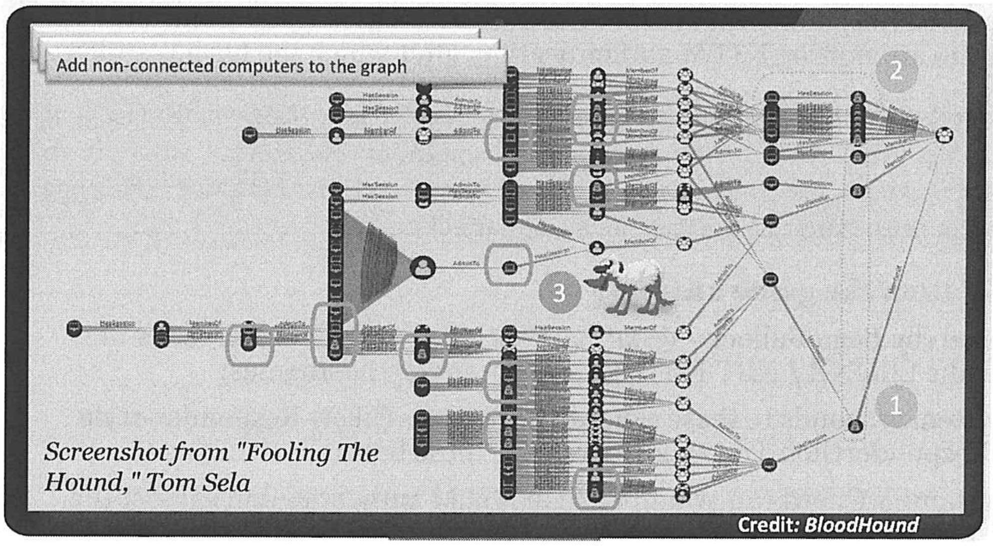
Original post: https://isc.sans.edu/forums/diary/Detecting+Mimikatz+Use+On+Your+Network/19311/

As we've seen during Day 4 of this course, Mimikatz is a highly popular tool that is being used both by penetration testers / red teamers and real adversaries. One of its key functions is to dump password hashes belonging to authenticated users from the LSA process memory. The idea here is, of course, to trick adversaries using Mimikatz by providing them with fake hashes for accounts that don't actually exist. We can then monitor attempted use of these credentials. Note that this attack technique will not only trick Mimikatz but also other tools that attempt to extract password hashes from memory.

This technique has been further developed and as part of the Empire framework, a "New-HoneyHash.ps1" PowerShell script was created, which allows for the creation of fake hashes in the LSA process memory!

© 2019 Erik Van Buggenhout & Stephen Sims

Some Example Canaries – Fooling The Hound


Add non-connected computers to the graph
2
3
1
Screenshot from "Fooling The Hound," Tom Sela
Credit: BloodHound

Fooling the hound

Tom Sela (Illusive Networks) presented an interesting talk called "Fooling The Hound."

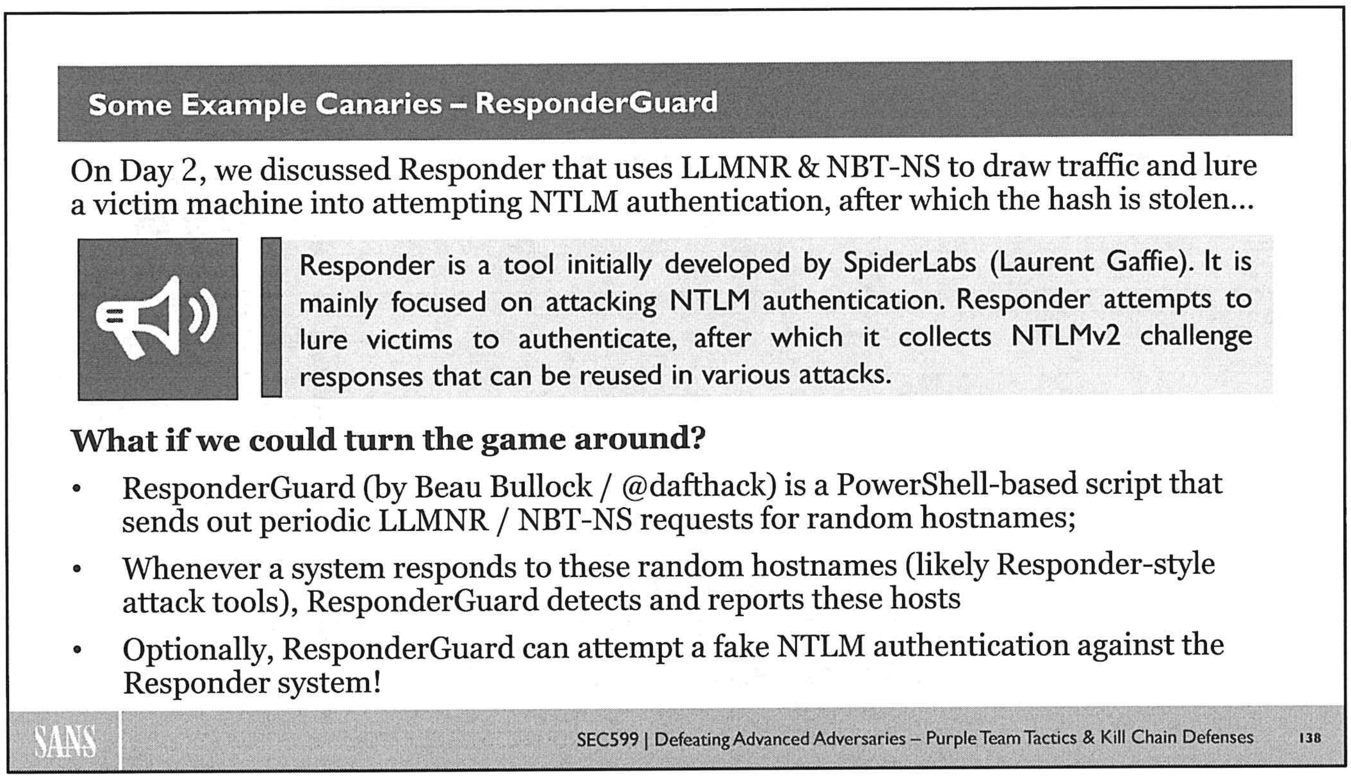
It is focused on generating additional attack paths to confuse / slow down adversaries in the environment.

Some Example Canaries – Fooling The Hound
Another good example of a creative deception strategy was presented by Tom Sela (Illusive Networks). He created a talk called "Fooling The Hound – Deceiving Domain Admin Hunters." The graph on the slide is an extract from the presentation. Fake paths are generated by providing fake responses to:

- RegEnumK (used to enumerate HKEY_USERS hive).
- NetWkstaUserEnum (similar to honeyhashes described previously).
- NetSessionEnum (used to enumerate user sessions).
- NetLocalGroupGetMembers (used to enumerate group memberships).

The full presentation can be found here: https://cdn.shopify.com/s/files/1/0177/9886/files/phv2017-tsela.pdf

© 2019 Erik Van Buggenhout & Stephen Sims



Some Example Canaries – ResponderGuard

On Day 2, we discussed Responder that uses LLMNR & NBT-NS to draw traffic and lure a victim machine into attempting NTLM authentication, after which the hash is stolen...

Responder is a tool initially developed by SpiderLabs (Laurent Gaffie). It is mainly focused on attacking NTLM authentication. Responder attempts to lure victims to authenticate, after which it collects NTLMv2 challenge responses that can be reused in various attacks.

What if we could turn the game around?

- ResponderGuard (by Beau Bullock / @dafthack) is a PowerShell-based script that sends out periodic LLMNR / NBT-NS requests for random hostnames;
- Whenever a system responds to these random hostnames (likely Responder-style attack tools), ResponderGuard detects and reports these hosts
- Optionally, ResponderGuard can attempt a fake NTLM authentication against the Responder system!

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 138

Some Example Canaries – ResponderGuard
On Day 2, we discussed Responder that uses LLMNR & NBT-NS to draw traffic and lure a victim machine into attempting NTLM authentication, after which the hash is stolen… As a quick refresher:

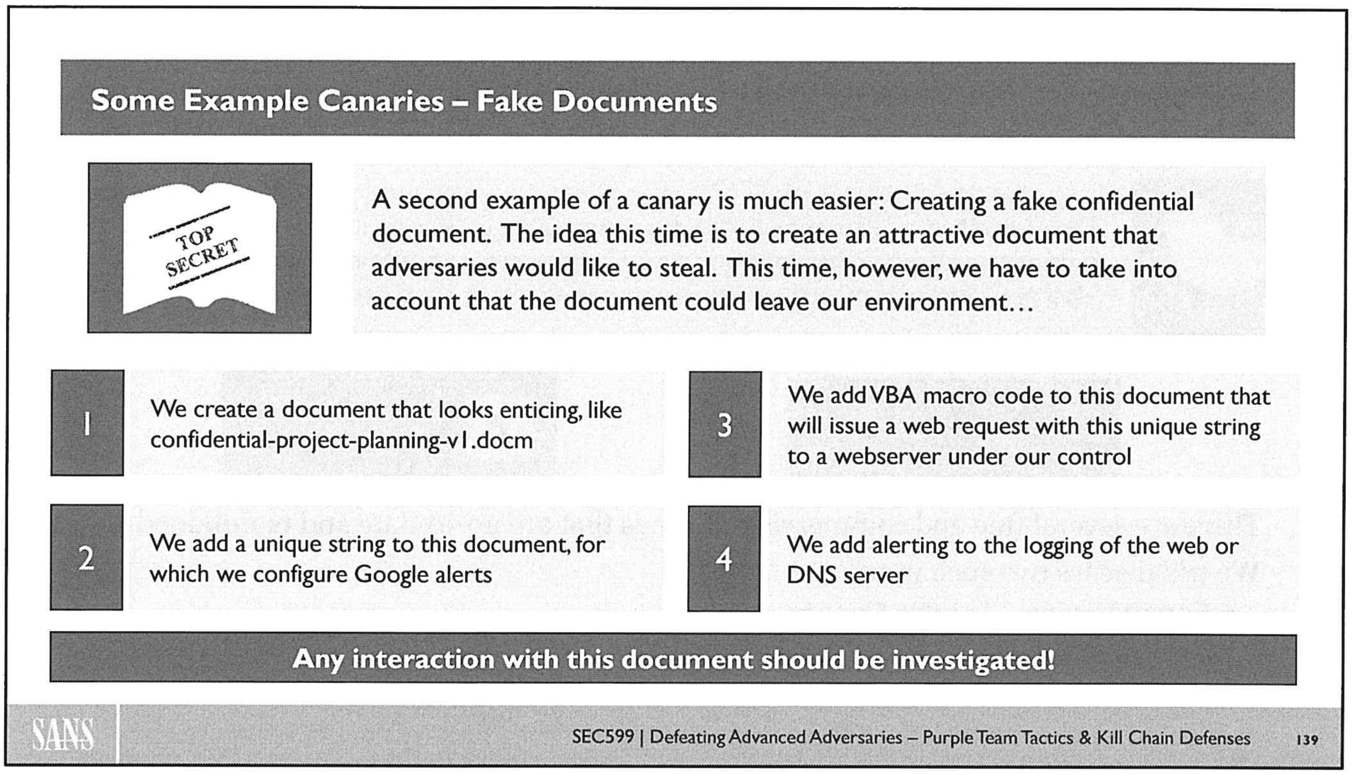
Responder is a tool initially developed by SpiderLabs (Laurent Gaffie). It is mainly focused on attacking NTLM authentication. Responder attempts to lure victims to authenticate, after which it collects NTLMv2 challenge responses that can be reused in various attacks.

But what if we could turn the game around and fool the adversary? This is where ResponderGuard comes in!

- ResponderGuard is a PowerShell-based script that sends out periodic LLMNR / NBT-NS requests for random hostnames.
- Whenever a system responds to these random hostnames (likely Responder-style attack tools) ResponderGuard detects and reports these hosts.
- Optionally, ResponderGuard can attempt a fake NTLM authentication against the Responder system!

ResponderGuard can be found here:
https://github.com/CredDefense/CredDefense/blob/master/scripts/ResponderGuard.ps1

© 2019 Erik Van Buggenhout & Stephen Sims



Some Example Canaries – Fake Documents

A second example of a canary is much easier: Creating a fake confidential document. The idea this time is to create an attractive document that adversaries would like to steal. This time, however, we have to take into account that the document could leave our environment…

1 We create a document that looks enticing, like confidential-project-planning-v1.docm

2 We add a unique string to this document, for which we configure Google alerts

3 We add VBA macro code to this document that will issue a web request with this unique string to a webserver under our control

4 We add alerting to the logging of the web or DNS server

Any interaction with this document should be investigated!

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 139

Some Example Canaries – Fake Documents
A second example of a canary is much easier: Creating a fake confidential document. The idea this time is to create an attractive document that adversaries would like to steal. This time, however, we have to take into account that the document could leave our environment…

In our example, we create a Word document, but canaries of this type can also be created with other formats, for example, PDF:

1. We create a Word document with a name that looks enticing, like confidential-project-planning-v1.docm.

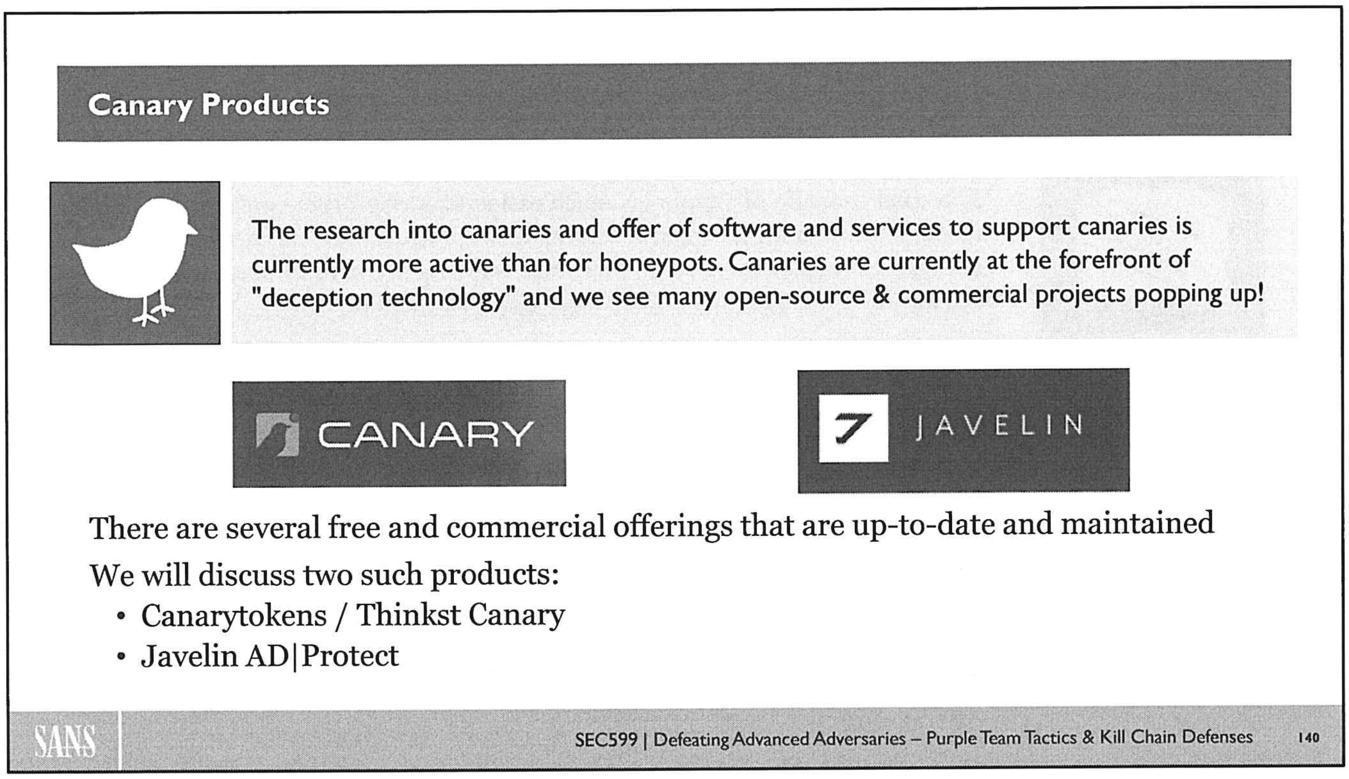
2. Next, we want to generate alerts whenever this document is copied or accessed (for example, when it is opened). We can achieve this by adding a unique string to this document (for example, corporate asset JDE0345HND), and then we configure alerts when we see content with this string moving inside our network or even appearing on the internet.

   Alerts for movements inside our network can be done with IDS for example, we create a simple rule to alert whenever the string JDE0345HND is found. This rule should certainly be implemented in the IDSs on the perimeter of our corporate network so that we are alerted when this canary document is exfiltrated.

   We can also configure Google alerts (or other alerting services) whenever this string is found on a page on the internet (and of course, indexed by Google). For example, if the content of this document would appear on pastebin we want an alert.

3. Finally, we add VBA macro code to this document that will issue a web request with this unique string to a webserver under our control whenever the document is opened. We add alerting to the logging of the webserver so that we are alerted whenever this request is made to the webserver.

Other methods besides VBA macros exist, for example, templates or embedded, linked objects.



Canary Products

The research into canaries and offer of software and services to support canaries is currently more active than for honeypots. Canaries are currently at the forefront of "deception technology" and we see many open-source & commercial projects popping up!

CANARY

7 JAVELIN

There are several free and commercial offerings that are up-to-date and maintained

We will discuss two such products:

- Canarytokens / Thinkst Canary
- Javelin AD|Protect

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 140

Canary Products

The research into canaries and offer of software and services to support canaries is currently more active than for honeypots. Canaries are currently at the forefront of "deception technology" and we see many open-source and commercial projects popping up!

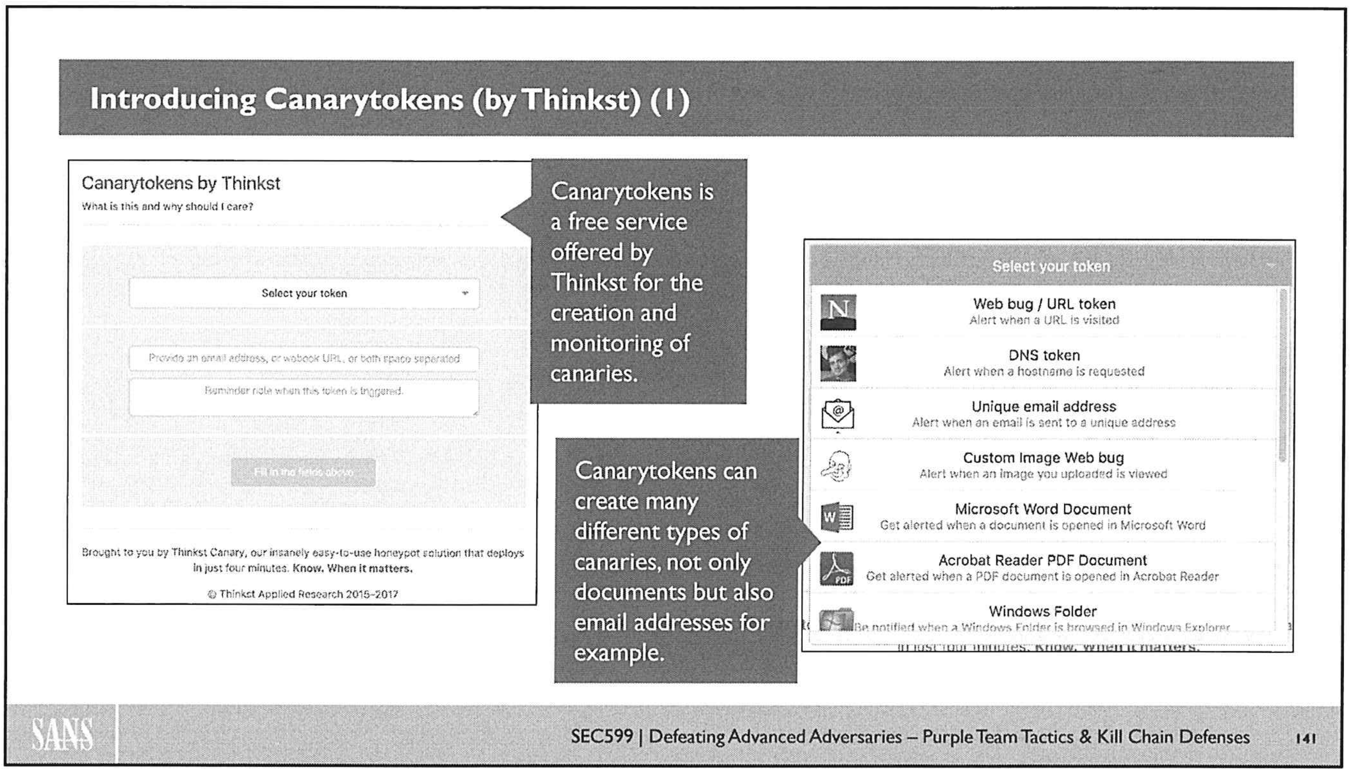
There are several free and commercial offerings for canaries that are up-to-date and actively maintained.

We will discuss 2 products:

- Canarytokens / Thinkst Canary
- Javelin AD|Protect

Canarytokens is a free service offered by Thinkst to create canaries and be alerted whenever these canaries are used. Several types of canaries can be created. Thinkst Canary is Thinkst's commercial offering for canaries. Not only do they provide canaries, but also honeypots.

Javelin AD|Protect is a commercial canary application for Active Directory. When deployed, it will create different types of canaries inside your corporate Active Directory infrastructure and alert you when these canaries are used by attackers.



Introducing Canarytokens (by Thinkst) (1)

Canarytokens by Thinkst

What is this and why should I care?

Select your token

Provide an email address, or webook URL, or both space separated

Reminder note when this token is triggered.

Fill in the fields above

Brought to you by Thinkst Canary, our insanely easy-to-use honeypot solution that deploys in just four minutes. **Know. When it matters.**

© Thinkst Applied Research 2015-2017

Canarytokens is a free service offered by Thinkst for the creation and monitoring of canaries.

Canarytokens can create many different types of canaries, not only documents but also email addresses for example.

Select your token

Web bug / URL token
Alert when a URL is visited

DNS token
Alert when a hostname is requested

Unique email address
Alert when an email is sent to a unique address

Custom Image Web bug
Alert when an image you uploaded is viewed

Microsoft Word Document
Get alerted when a document is opened in Microsoft Word

Acrobat Reader PDF Document
Get alerted when a PDF document is opened in Acrobat Reader

Windows Folder
Be notified when a Windows Folder is browsed in Windows Explorer

SANS

SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 141

Introducing Canarytokens (by Thinkst) (1)
The screenshot above (left) shows the free Canarytokens service as it is provided by Thinkst.

This page can be used to create your own, personalized canary together with the data necessary to alert you whenever this canary is used.

First, you have to select the type of canary you want to create. Many types are possible, varying from documents to email addresses. We will discuss this on the next page.

Then you have to provide an email address or URL. This will be used to alert you whenever your canary is accessed. When your canary is used by attackers, an event will be sent to the Canarytokens servers. This is typically a web request or a DNS name resolution. When the Canarytokens servers get this particular event for your canary, they will alert you via the email address or URL you provided.

As you can create many canaries with Canarytokens, you are also given the option to provide a reminder note that will help you remember why you created this particular canary. This reminder note will be included in the alert you receive from the Canarytokens servers.

Tricking the adversary
Here, we can see a partial list (screenshot right) of the type of canaries you can create using the free Canarytokens service. Canarytokens can create:

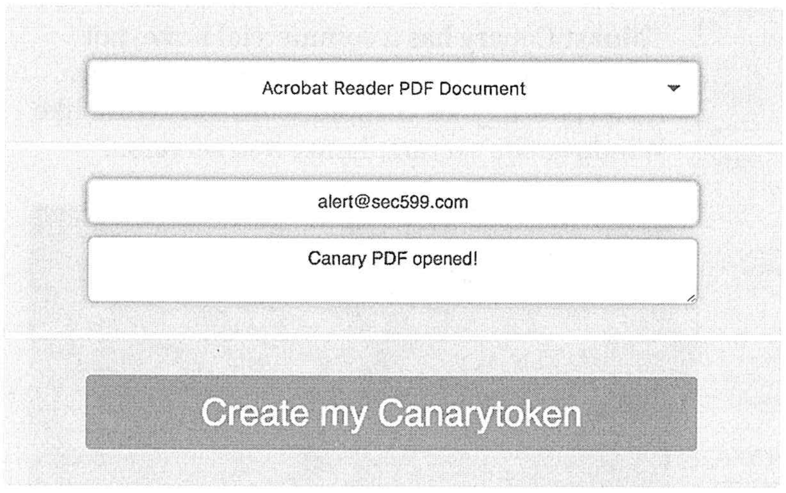
- Web bugs
- DNS tokens
- Email addresses
- Custom images
- Word documents
- PDF documents

- Windows folders
- Custom EXEs
- Canary for a cloned website
- SQL server
- QR code
- SVN
- AWS keys

Whenever one of these types is used, you will receive an alert.

© 2019 Erik Van Buggenhout & Stephen Sims

Introducing Canarytokens (by Thinkst) (2)



Here, we create a PDF document as canary.

We just have to provide and email address and a message, and then we can create the PDF document.

Whenever this document is opened, we will receive an email from Canarytokens.

Introducing Canarytokens (by Thinkst) (2)
This is the page you will see when you have completed the fields for the canary you want to create with Canarytokens.

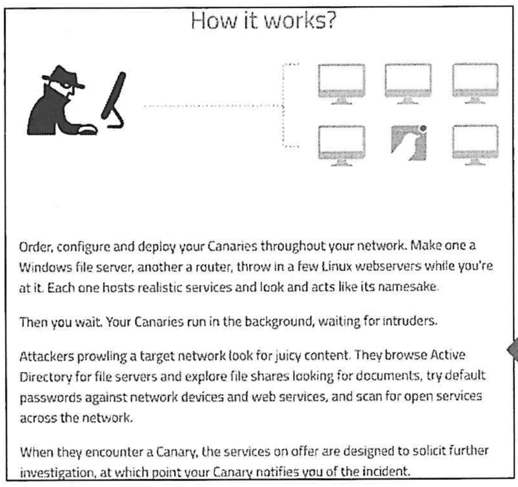
In this example, we create an Acrobat Reader PDF document with a canarytoken that will send an email to address alert@sec599.com whenever the PDF document is opened.

The reminder note "Canary PDF opened!" will be included in the email when an alert is triggered.

By clicking "Create my canarytoken", the PDF is created and available for download, and entries will be created in Canarytokens database to alert you.

The method for PDFs is the following: The PDF that is created by Canarytokens contains a unique URL that is automatically requested whenever the PDF is opened with Adobe Reader. The user will be warned and asked for authorization to execute the web request; however, the DNS request for the domain included in the URL will be issued without any warning. Canarytokens servers will receive this unique DNS request and alert you via email.

Introducing Canarytokens (by Thinkst) (3)


How it works?

Order, configure and deploy your Canaries throughout your network. Make one a Windows file server, another a router, throw in a few Linux webservers while you're at it. Each one hosts realistic services and look and acts like its namesake.

Then you wait. Your Canaries run in the background, waiting for intruders.

Attackers prowling a target network look for juicy content. They browse Active Directory for file servers and explore file shares looking for documents, try default passwords against network devices and web services, and scan for open services across the network.

When they encounter a Canary, the services on offer are designed to solicit further investigation, at which point your Canary notifies you of the incident.

Thinkst Canary has a commercial honeypot solution that allows you to create and deploy honeypots in your corporate infrastructure like Windows file servers, Linux web servers...

Thinkst Canary is the commercial offering of Thinkst. Next to honeypots described above, another advantage it has when compared to the free Canarytokens service is that it allows you to configure a canary architecture that does not rely on the infrastructure set up by Thinkst, thus offering increased confidentiality!

Introducing Canarytokens (by Thinkst) (3)
Canarytokens is a free service and as such has limitations that can impair its use by a large corporation.

For example, corporate clients will want:

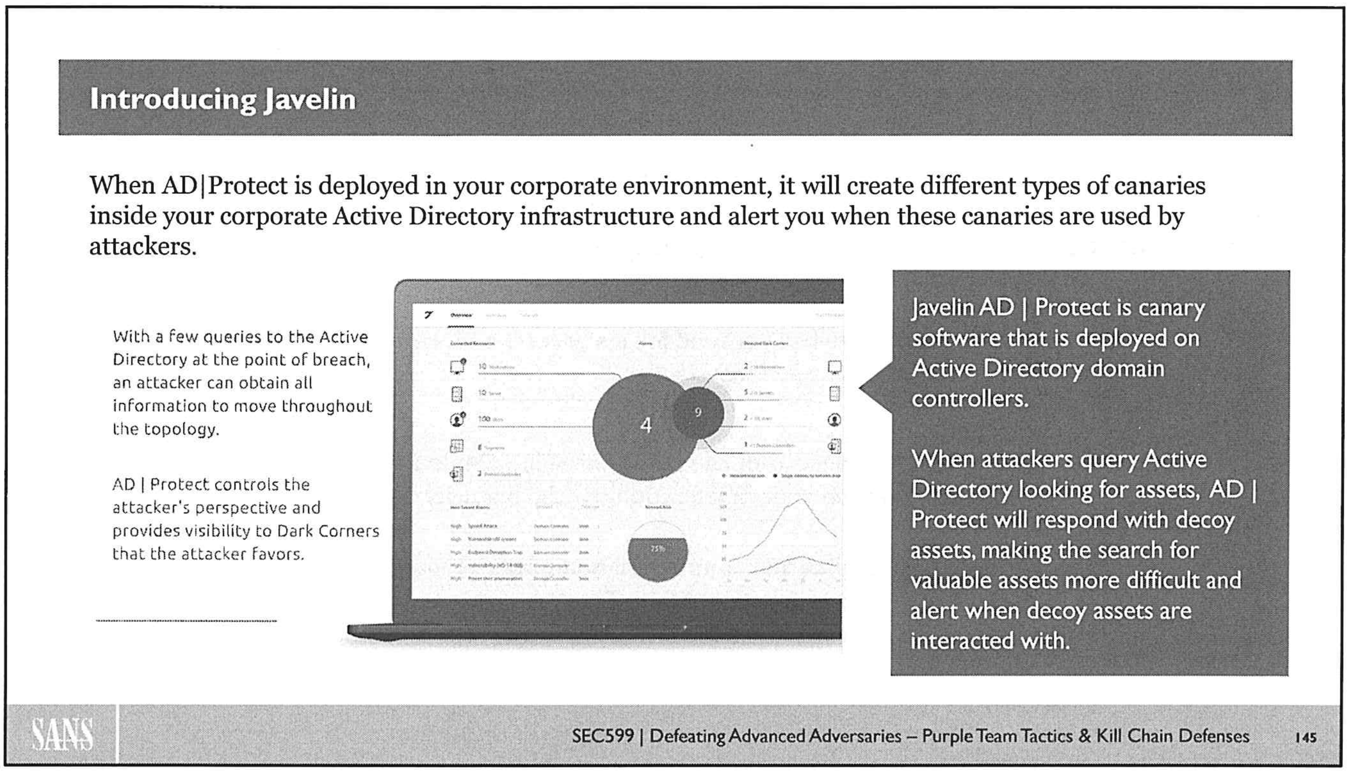
- More options for alerting than email and URL.
- SLAs to guarantee a certain level of service.
- Strict confidentiality: with Canarytokens, the alerts are processed by Thinkst's servers .
- ...

These requirements can be accommodated by opting for a commercial solution. Thinkst Canary is the commercial offering of Canarytokens, and this will, for example, allow you to host your own alerting servers so that you don't have to rely on Thinkst and that they are also not aware of potential breaches.

Thinkst Canary not only offers canaries like Canarytokens but also honeypots.

Thinkst Canary has a commercial honeypot solution that allows you to create and deploy honeypots in your corporate infrastructure like Windows file servers, Linux web servers, ...

© 2019 Erik Van Buggenhout & Stephen Sims

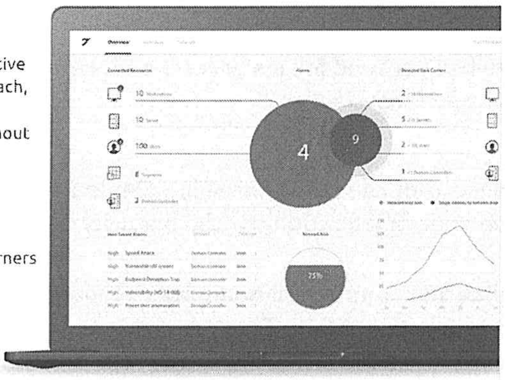


Introducing Javelin

When AD|Protect is deployed in your corporate environment, it will create different types of canaries inside your corporate Active Directory infrastructure and alert you when these canaries are used by attackers.

With a few queries to the Active Directory at the point of breach, an attacker can obtain all information to move throughout the topology.

AD | Protect controls the attacker's perspective and provides visibility to Dark Corners that the attacker favors.



Javelin AD | Protect is canary software that is deployed on Active Directory domain controllers.

When attackers query Active Directory looking for assets, AD | Protect will respond with decoy assets, making the search for valuable assets more difficult and alert when decoy assets are interacted with.

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 145

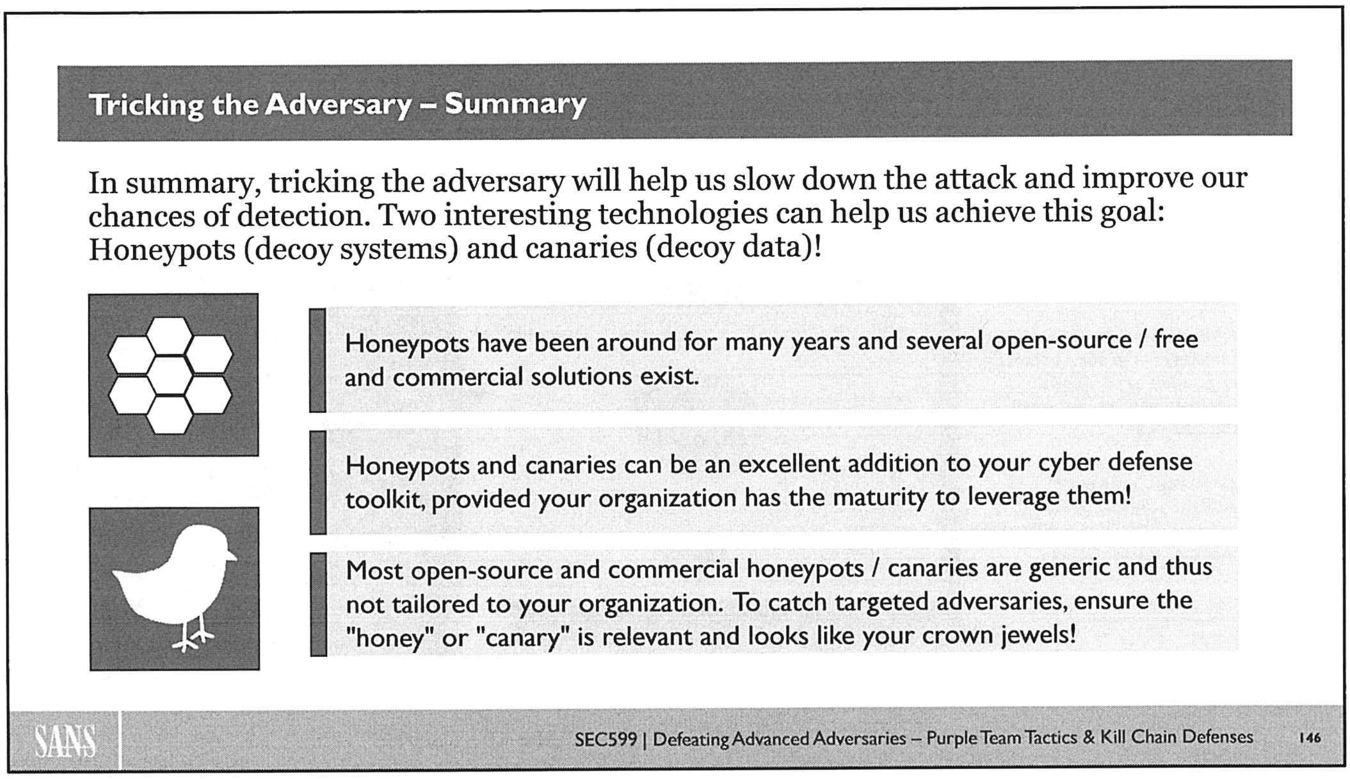
Introducing Javelin
Another commercial canary solution for Active Directory we want to mention is Javelin AD|Protect.

When AD|Protect is deployed in your corporate environment, it will create different types of canaries inside your corporate Active Directory infrastructure and alert you when these canaries are used by attackers.

AD|Protect will not only create canaries like fake administrative accounts and fake Windows servers, but it will offer several other deception technologies.

Active Directory is a repository (directory) for all your corporate (Windows) assets and can thus, be queried to obtain a list of assets. For example, any domain member (a user or a machine like a workstation) can query an Active Directory domain controller to obtain a list of all computers that are members of an Active Directory domain. This function does not require special privileges; all members can query information like this. For example, the Petya/Notpetya ransomware worm would use this functionality to obtain a list of all computers on the network to attack.

For example, AD|Protect can be configured to interfere with this functionality: It can inject a huge number of fake computers in the query results. When attackers query Active Directory domain controllers to obtain a list of all computers, AD|Protect will inject fake computers so that the attackers don't know what results are real and fake, and thus slow down their attack. For example, if you have 1,000 workstations in your Active Directory domain, AD|Protect can make it look like there are 10,000 workstations in your domain. Not only does this confuse the attackers, but it increases the chances that attackers will try to access a fake workstation significantly, and then AD|Protect will generate alerts.



Tricking the Adversary – Summary

In summary, tricking the adversary will help us slow down the attack and improve our chances of detection. Two interesting technologies can help us achieve this goal: Honeypots (decoy systems) and canaries (decoy data)!

Honeypots have been around for many years and several open-source / free and commercial solutions exist.

Honeypots and canaries can be an excellent addition to your cyber defense toolkit, provided your organization has the maturity to leverage them!

Most open-source and commercial honeypots / canaries are generic and thus not tailored to your organization. To catch targeted adversaries, ensure the "honey" or "canary" is relevant and looks like your crown jewels!

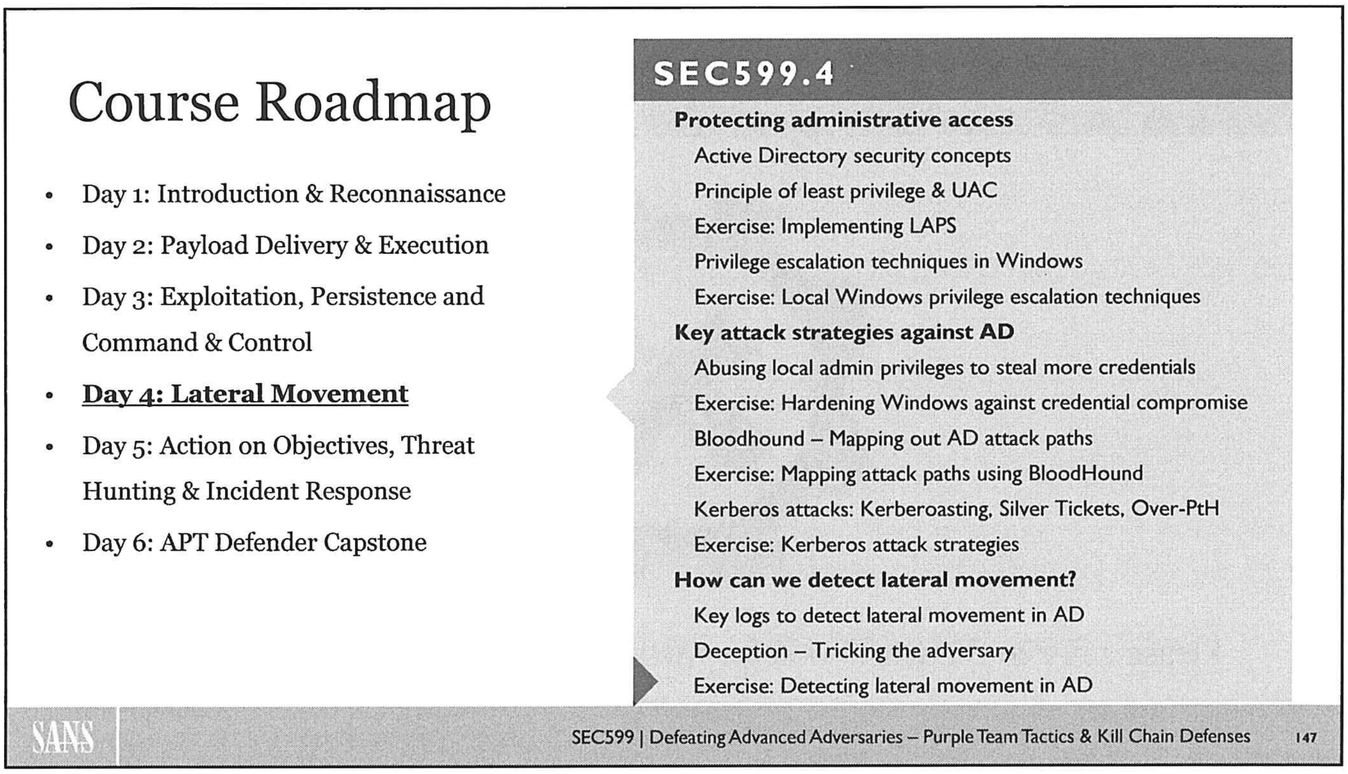
SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 146

Tricking the Adversary – Summary
In summary, tricking the adversary will help us slow down the attack and improve our chances of detection. Two interesting technologies can help us achieve this goal: Honeypots (decoy systems) & canaries (decoy data)! Honeypots are decoy systems: They simulate systems or services and are designed to try to attract the attention of our adversaries. Canaries are decoy data: They simulate corporate data and documents that look enticing to our adversaries and are designed to alert us when this data is consumed.

Here are a few closing thoughts:

- Honeypots have been around for many years and several open-source / free and commercial solutions exist.
- Honeypots and canaries can be an excellent addition to your cyber defense toolkit, provided your organization has the maturity to leverage them!
- Most open-source and commercial honeypots / canaries are generic and thus not tailored to your organization. To catch targeted adversaries, ensure the "honey" or "canary" is relevant and looks like your crown jewels!

© 2019 Erik Van Buggenhout & Stephen Sims

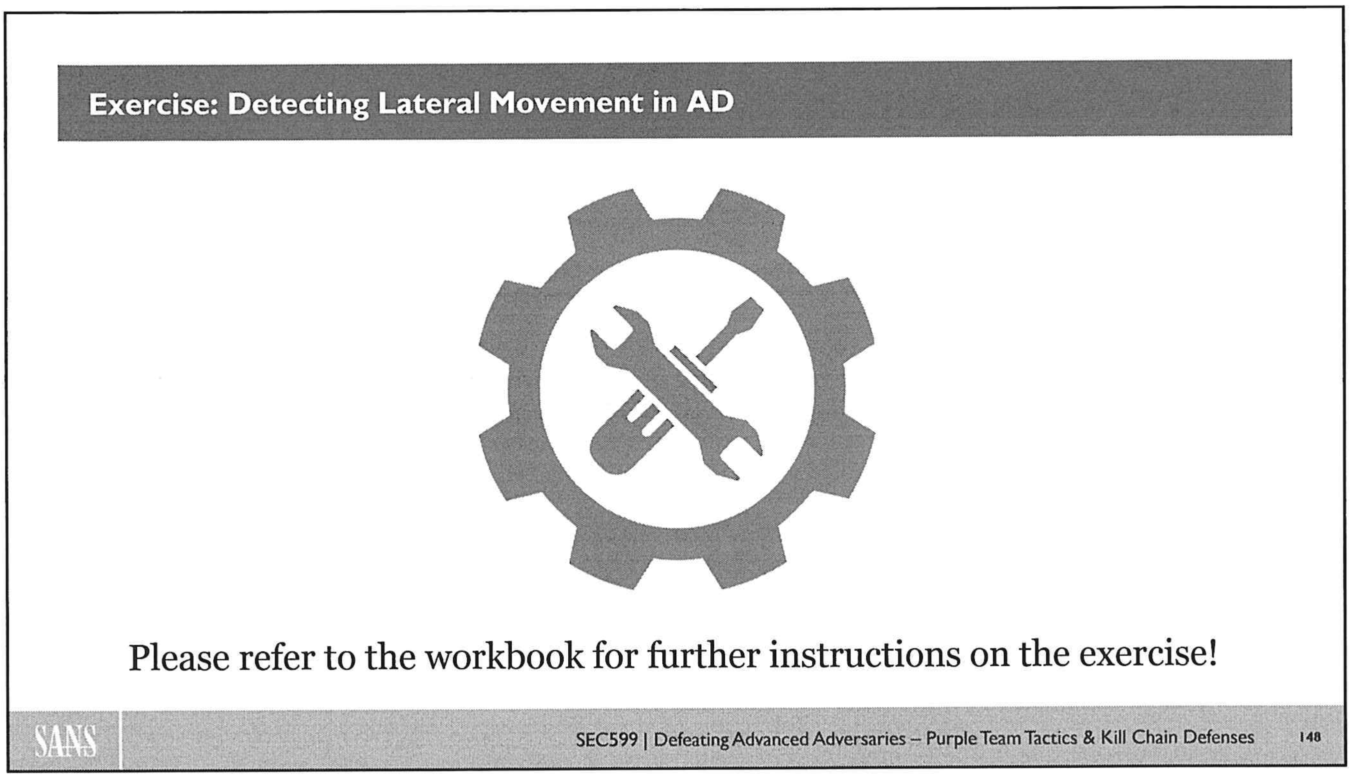

Course Roadmap

- Day 1: Introduction & Reconnaissance
- Day 2: Payload Delivery & Execution
- Day 3: Exploitation, Persistence and Command & Control
- **Day 4: Lateral Movement**
- Day 5: Action on Objectives, Threat Hunting & Incident Response
- Day 6: APT Defender Capstone

SEC599.4

This page intentionally left blank.

© 2019 Erik Van Buggenhout & Stephen Sims



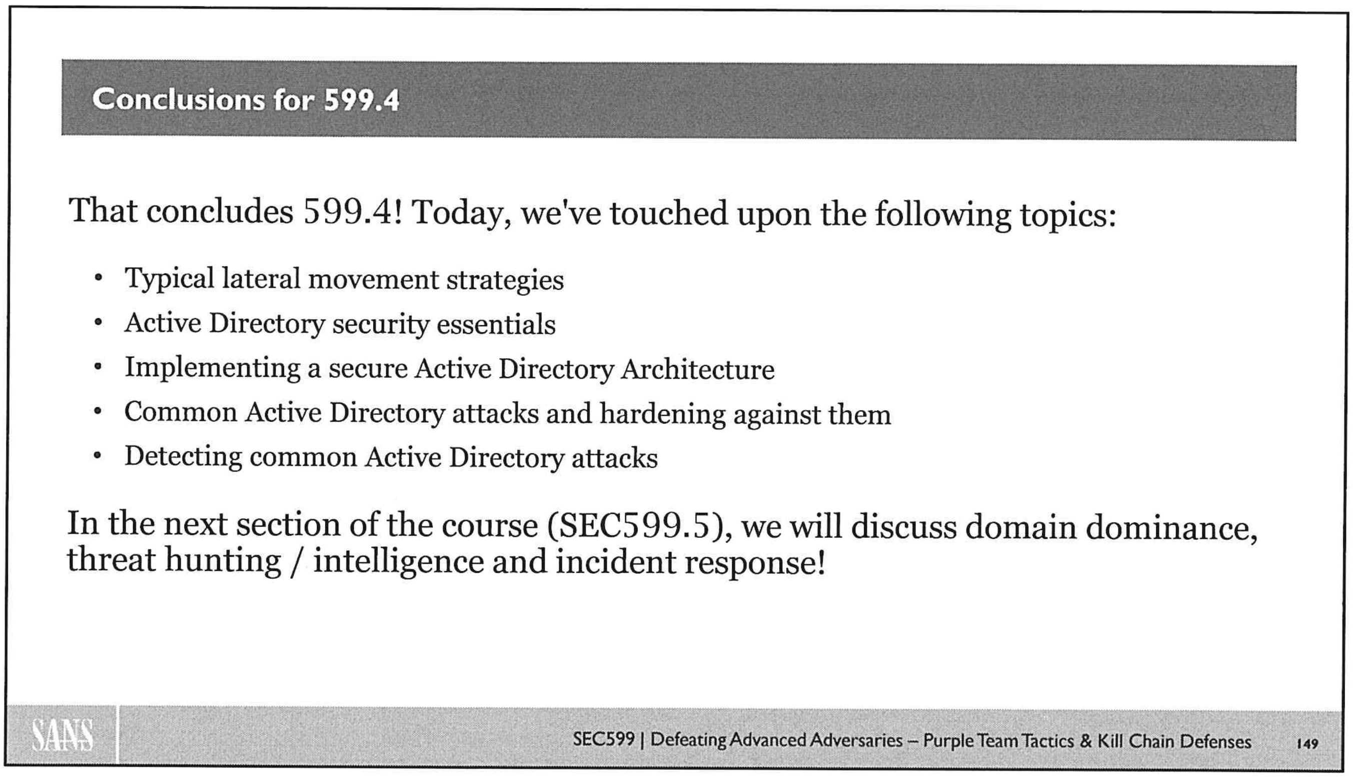
Exercise: Detecting Lateral Movement in AD

Please refer to the workbook for further instructions on the exercise!

SANS | SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 148

This page intentionally left blank.

© 2019 Erik Van Buggenhout & Stephen Sims



Conclusions for 599.4

That concludes 599.4! Today, we've touched upon the following topics:

- Typical lateral movement strategies
- Active Directory security essentials
- Implementing a secure Active Directory Architecture
- Common Active Directory attacks and hardening against them
- Detecting common Active Directory attacks

In the next section of the course (SEC599.5), we will discuss domain dominance, threat hunting / intelligence and incident response!

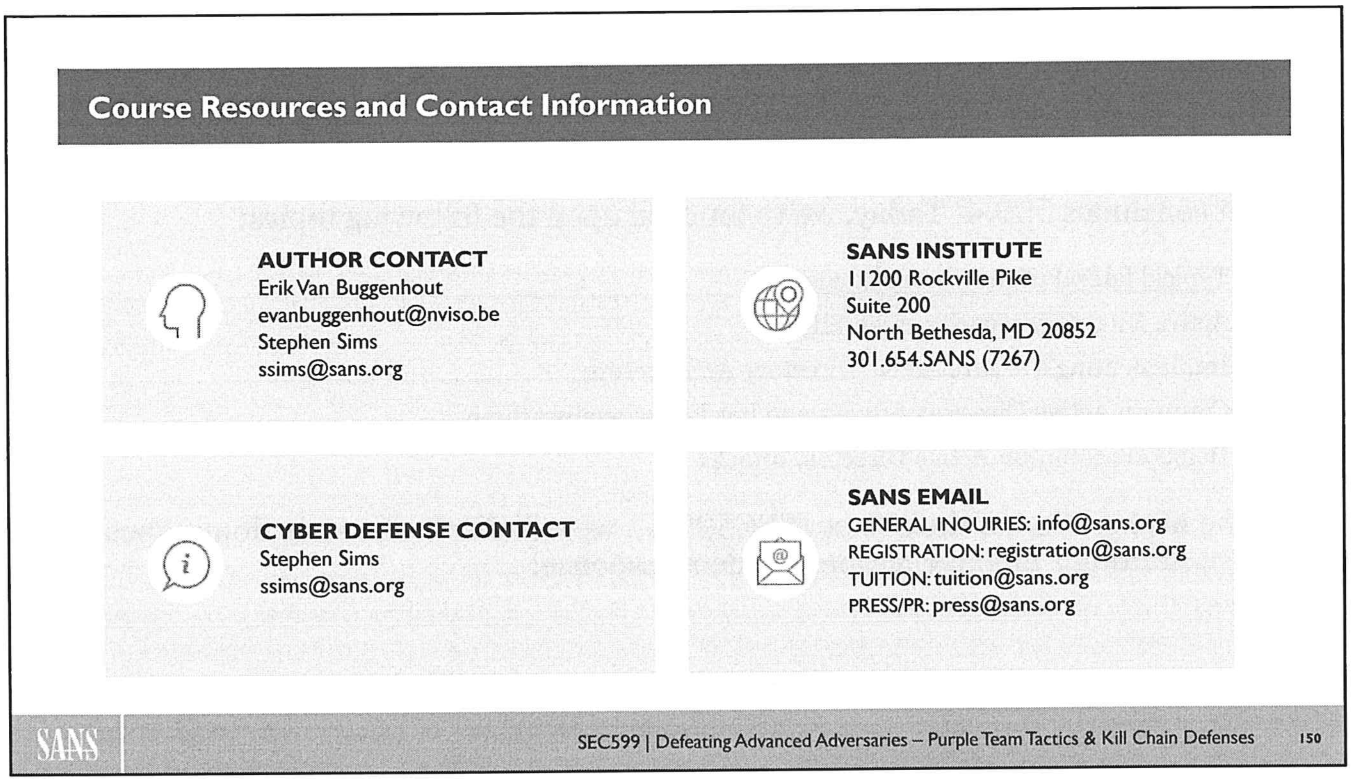
Conclusions for 599.4

Farewell, 599.4! Today, we focused on persistence, privilege escalation, command & control, and lateral movement. Among others, we touched upon the following topics:

- Typical lateral movement strategies
- Active Directory security essentials
- Implementing a secure Active Directory Architecture.
- Common Active Directory attacks and hardening against them.
- Detecting common Active Directory attacks.

In the next section of the course (SEC599.5), we will discuss domain dominance, threat hunting / intelligence and incident response!

© 2019 Erik Van Buggenhout & Stephen Sims



Course Resources and Contact Information

AUTHOR CONTACT
Erik Van Buggenhout
evanbuggenhout@nviso.be
Stephen Sims
ssims@sans.org

SANS INSTITUTE
11200 Rockville Pike
Suite 200
North Bethesda, MD 20852
301.654.SANS (7267)

CYBER DEFENSE CONTACT
Stephen Sims
ssims@sans.org

SANS EMAIL
GENERAL INQUIRIES: info@sans.org
REGISTRATION: registration@sans.org
TUITION: tuition@sans.org
PRESS/PR: press@sans.org

SANS SEC599 | Defeating Advanced Adversaries – Purple Team Tactics & Kill Chain Defenses 150

This page intentionally left blank.

© 2019 Erik Van Buggenhout & Stephen Sims