CYBERARK®

LABS RESEARCH

# PREDICTING RISK: CREDENTIAL THEFT FORESIGHT


## Abstract

What do the infamous attacks on the SWIFT banking network and the attack that took down Ukraine's power grid have in common? The exploitation of privileged credentials sit at the center of these, and nearly all other, successful attacks.

While it sounds like something out of a movie, credential theft foresight - or precognition - is entirely possible and effective in identifying network weak spots likely to expose privileged credentials to compromise. With credential theft precognition, organizations can identify specific accounts and machines that are likely to be involved in an attack, and minimize their attack surface.

This whitepaper examines the research behind credential theft precognition. We look at how it's a significant defensive advantage over traditional security tools, the multi-step process for identifying network weak spots, use cases and more.



**Authors:** Lavi Lazarovitz, Asaf Hecht

# Table of Contents

# Introduction

Attackers and red teamers attempting to circumvent organizational security controls often adopt the principle of 'living off the land' by using whatever the endpoint or the network have to offer without introducing any external tools or crafted malware. Operations guided by this principle rely heavily on compromised credentials that allow the attackers to disguise themselves as legitimate users with flexible privileged access.

This realization follows a simple conclusion: the shortest path to privileged account compromise is the shortest path to network compromise. Several tools and frameworks were built around this idea to allow red teamers to take advantage of this. BloodHound, created by Andy Robins, Will Schroder and Justin Warner, is one example of such a framework.

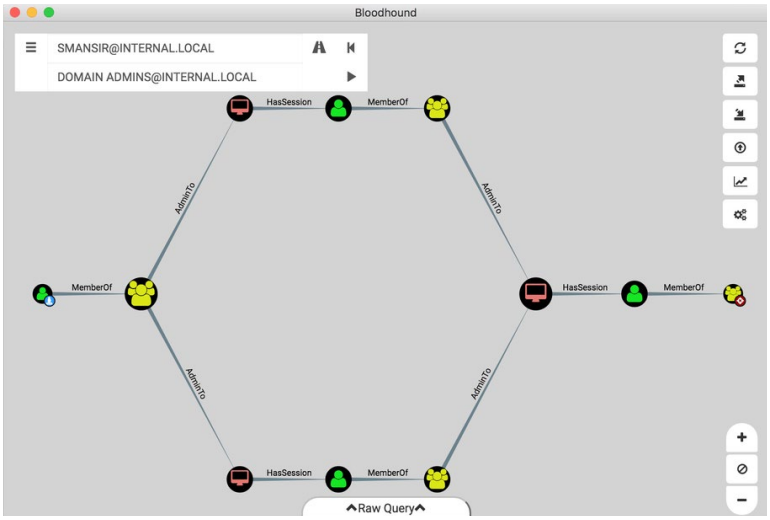



***Figure 1*** *- BloodHound framework showing potential attack vectors*

As can be seen in Figure 1, BloodHound allows red and blue teamers to visualize the shortest path to privileged account compromise. The figure shows two potential attack vectors starting with an account with admin privileges on two machines where there is another account connected, which has admin access on another machine where a domain admin is currently connected. Hence this domain admin account is currently at risk and vulnerable to credential theft.

Frameworks like BloodHound are very useful for red teamers who only need one viable attack vector to escalate privilege and take control over the domain. Blue teamers, however, must cover and mitigate all possible attack vectors to protect the network. Aside from the network complexity that enterprise networks might present, the major difference for blue teams is that it can be impossible for organizations to protect their networks and their privileged accounts. For example, in a small network with 100 machines, the number of attack vectors might reach $100^2$ in a worst case scenario. In such cases, it is not trivial to spot what link or asset should be handled first to maximize the number of neutralized vectors. Alternatively, the identification of a HotSpot will allow organizations to foresee and prevent future attack vectors. Simply put, they can achieve precognition of credential theft.

After identifying the involved accounts on a newly created HotSpot, the organization can investigate the reason for the initiation of connection and prevent such future connections. As a result, organizations can prevent future HotSpots from being created, therefore minimizing the attack surface and enhancing the security posture of the organization.

Another thing to consider when observing attack vectors based on credential theft is the fact that credentials might be exposed in limited time frames. Hence a short-term attack won't necessarily be identified by blue teamers taking a snapshot of the network using BloodHound or similar frameworks.

This is where our credential theft precognition research was born – to help defenders close this gap.

## HotSpots and ColdSpots

In order to understand how credential theft precognition works, it's important to know what HotSpots and ColdSpots are. Essentially, they're key predictors that make precognition possible.

HotSpots are areas within a network that are predictably involved with an attack. As machines that expose privileged accounts to credential theft, they're bottlenecks for dozens of potential attack vectors.

ColdSpots, on the other hand, are machines hosting privileged accounts that could be targeted by attackers in an attempt to escalate privileges. If that happens, the ColdSpot will be transformed into a HotSpot.

Figure 2 below depicts examples of HotSpots and ColdSpots.

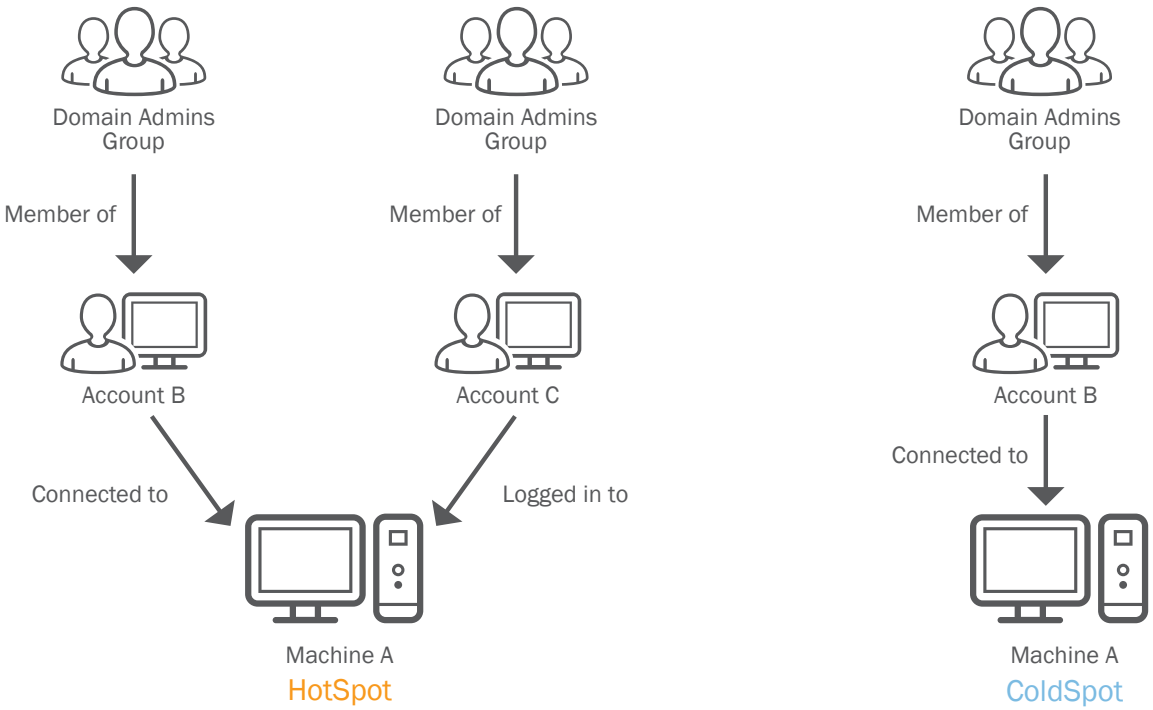



***Figure 2** – HotSpot scenario (left), ColdSpot scenario (right)*

As new attack techniques and variants are continuously introduced, credential theft precognition using HotSpot detection presents a significant defensive advantage over traditional security controls, like vulnerability scanners and Intrusion Detection Systems (IDS). These controls usually look for known malicious activities and indicators of compromise that might be absent when new techniques or variants are used.

## The Tier Doctrine

There are two significant advantages in a defensive strategy focusing on HotSpots. The first advantage is the granular perspective rather than a technique-specific perspective it enables. Attackers can choose from dozens of techniques to exploit a machine hosting a privileged account in the attempt of compromising the credentials. Moreover, obfuscation and encryption could be utilized to make it more difficult to identify malicious or suspicious activity.

A technique-specific approach attempts to identify a file hash or specific behavior that might fail in identifying the risk while hunting for HotSpots, allowing organizations to identify and react before the malicious activity has taken place.

The second advantage and benefit in focusing on HotSpots is the elimination of future credentials risk. Organizations that hunt for HotSpots will be able to identify workflows that do not comply with Microsoft's tier doctrine.

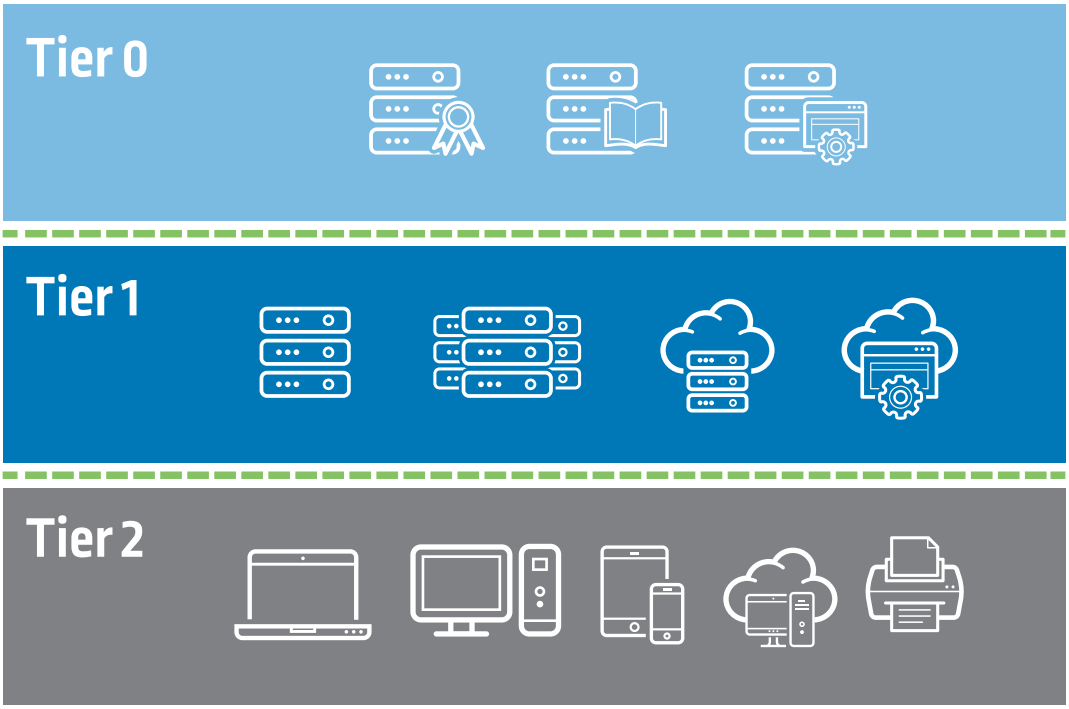



***Figure 3*** *– Microsoft tier doctrine diagram*

The tier doctrine guides organizations on segregating network access based on sensitivity or privileged tiers. For example, an administrative account with local administrative access to tier 1 assets should not have access to tier 0 or tier 2 assets. If such access is possible, a privileged account from another tier might be able to compromise it on a HotSpot and use it take over another network tier. By eliminating HotSpots and constituting secure workflows based on the tier doctrine, organizations can eliminate future risks and contain any machine compromise to tier compromise (and not network compromise) as the worst case scenario.

## Use Cases

1. **Machine HotSpots**: In the organizations below, all users use domain accounts with local admin privileges on their desktops and laptops. As soon as a privileged IT administrator or a privileged service connects to those machines, a HotSpot is created as the privileged credentials currently stored in the machine memory could be compromised by the domain user.

   By identifying the HotSpot and denying future connections of server admins to user endpoints, any possible attack vectors utilizing the HotSpot are neutralized, hence preventing future credential theft and privilege escalation opportunities.

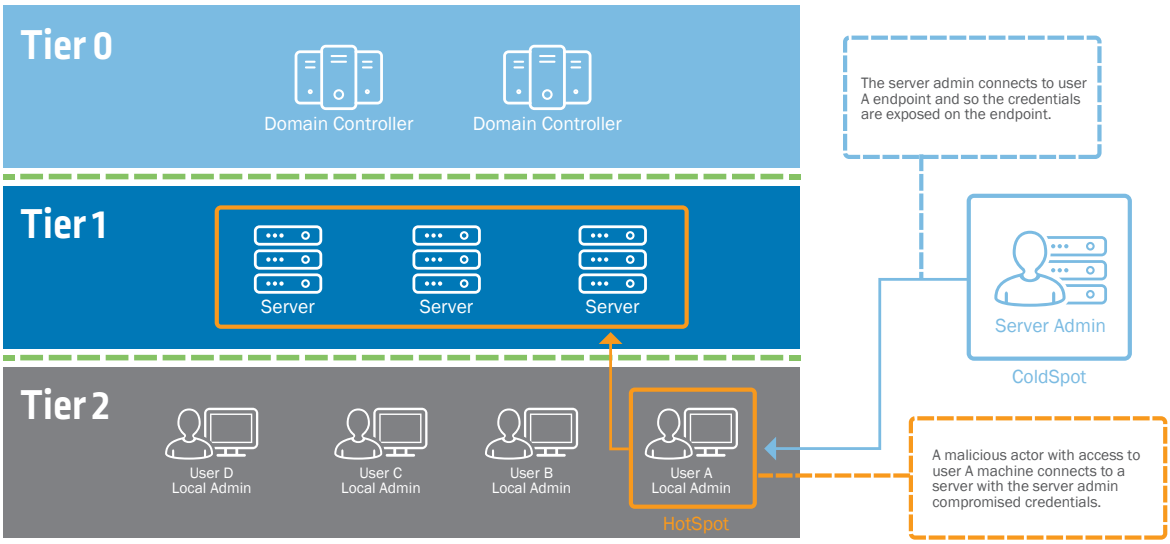



*Figure 4 – HotSpot in permissive organization*

2. **Delegation HotSpots**: In the organization below, users are restricted and do not have local admin rights on their laptops. Moreover, the organization adheres adequately to privileged account security principals and all users are granted privileges based on the least privilege principal. At some point, an administrator installed a service that requires impersonation privileges – the privilege to impersonate users in the domain. This action immediately transformed the service account to be highly privileged, as highly privileged accounts can be impersonated (in case the account is not protected or disabled for delegation). The machine hosting the service, and the new trusted for delegation account, is now considered a ColdSpot. As soon as a less-privileged account connects to that machine, the machine will then immediately transform into a HotSpot.

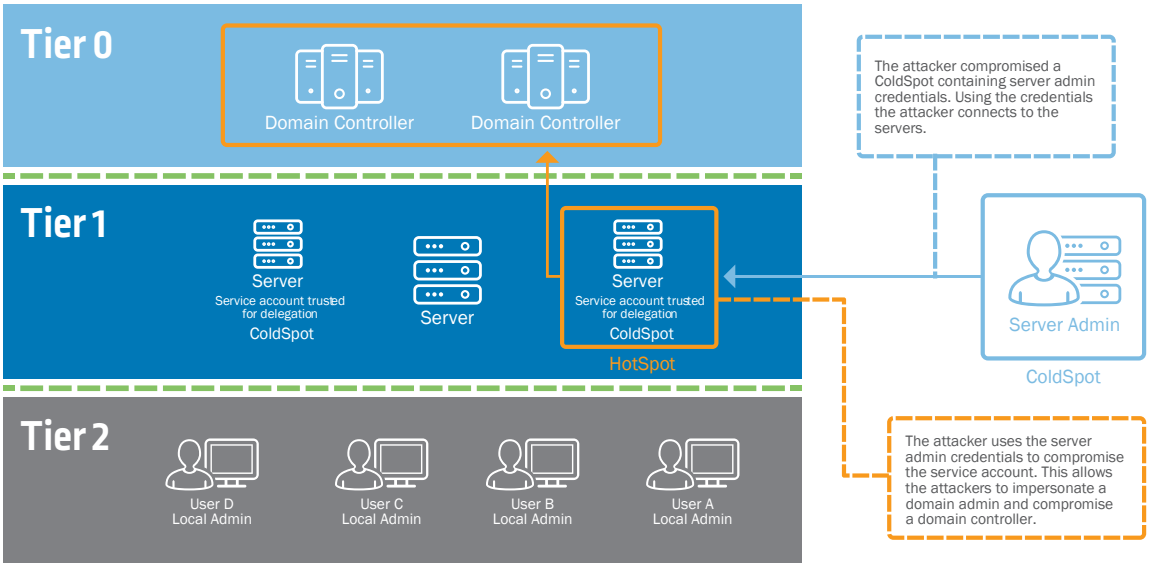
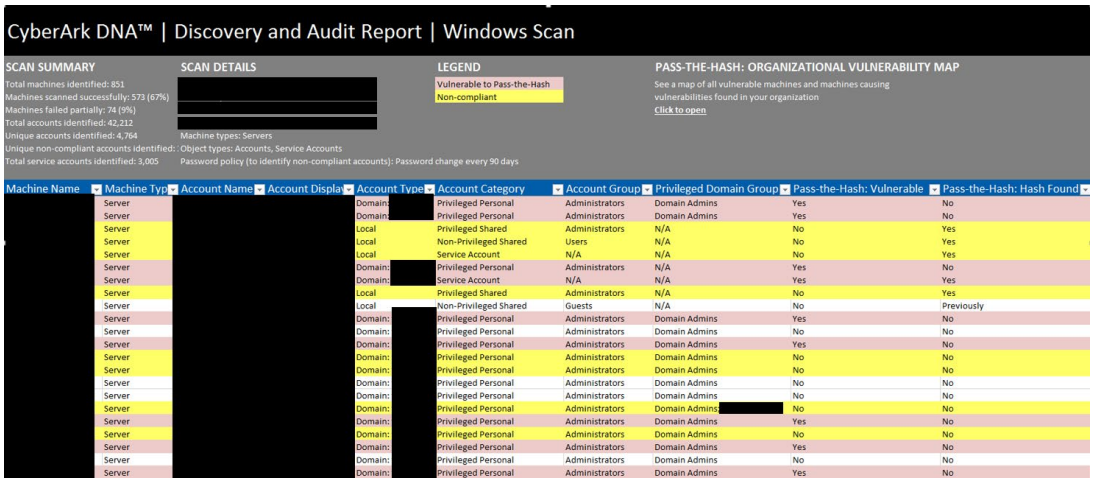



*Figure 5 – Delegation HotSpot creation*

# Statistics

To better visualize what HotSpots and ColdSpots look like in real organizations, we turned to CyberArk DNA™, which is used by many organizations to map privileged accounts across their networks. Below there is an obfuscated example of one such DNA scan we analyzed to identify HotSpots and ColdSpots and determine which machines and accounts were involved.



CyberArk DNA™ | Discovery and Audit Report | Windows Scan

SCAN SUMMARY
Total machines identified: 851
Machines scanned successfully: 573 (67%)
Machines failed partially: 74 (9%)
Total accounts identified: 42,212
Unique accounts identified: 4,764
Unique non-compliant accounts identified: :
Total service accounts identified: 3,005

SCAN DETAILS
[redacted]
Machine types: Servers
Object types: Accounts, Service Accounts
Password policy (to identify non-compliant accounts): Password change every 90 days

LEGEND
Vulnerable to Pass-the-Hash
Non-compliant

PASS-THE-HASH: ORGANIZATIONAL VULNERABILITY MAP
See a map of all vulnerable machines and machines causing vulnerabilities found in your organization
Click to open

| Machine Name | Machine Typ | Account Name | Account Displa | Account Type | Account Category | Account Group | Privileged Domain Group | Pass-the-Hash: Vulnerable | Pass-the-Hash: Hash Found |
|---|---|---|---|---|---|---|---|---|---|
| [redacted] | Server | [redacted] | [redacted] | Domain: | Privileged Personal | Administrators | Domain Admins | Yes | No |
| | Server | | | Domain: | Privileged Personal | Administrators | Domain Admins | Yes | No |
| | Server | | | Local | Privileged Shared | Administrators | N/A | No | Yes |
| | Server | | | Local | Non-Privileged Shared | Users | N/A | No | Yes |
| | Server | | | Local | Service Account | N/A | N/A | No | Yes |
| | Server | | | Domain: | Privileged Personal | Administrators | N/A | Yes | No |
| | Server | | | Domain: | Service Account | N/A | N/A | Yes | Yes |
| | Server | | | Local | Privileged Shared | Administrators | N/A | No | Yes |
| | Server | | | Local | Non-Privileged Shared | Guests | N/A | No | Previously |
| | Server | | | Domain: | Privileged Personal | Administrators | Domain Admins | Yes | No |
| | Server | | | Domain: | Privileged Personal | Administrators | Domain Admins | No | No |
| | Server | | | Domain: | Privileged Personal | Administrators | Domain Admins | Yes | No |
| | Server | | | Domain: | Privileged Personal | Administrators | Domain Admins | No | No |
| | Server | | | Domain: | Privileged Personal | Administrators | Domain Admins | No | No |
| | Server | | | Domain: | Privileged Personal | Administrators | Domain Admins | No | No |
| | Server | | | Domain: | Privileged Personal | Administrators | Domain Admins [redacted] | No | No |
| | Server | | | Domain: | Privileged Personal | Administrators | Domain Admins | Yes | No |
| | Server | | | Domain: | Privileged Personal | Administrators | Domain Admins | No | No |
| | Server | | | Domain: | Privileged Personal | Administrators | Domain Admins | Yes | No |
| | Server | | | Domain: | Privileged Personal | Administrators | Domain Admins | No | No |
| | Server | | | Domain: | Privileged Personal | Administrators | Domain Admins | Yes | No |

***Figure 6*** – *CyberArk DNA scan sample data*

The CyberArk DNA report lists the machines on the network and all of the accounts that have some privileges over those machines. The report includes details about each account, whether the account's hash is present on the machine and the type of presence the account has on the machine (RDP, scheduled task and so on).

Using data from CyberArk DNA scans, we are able to extract a snapshot of HotSpots and ColdSpots present in each organization at the time of the scan. As mentioned above, HotSpot and ColdSpot lifecycles might be short, so the analysis here is only a lower bound on the numbers of those spots.

## A Look at the Numbers

- Average number of ColdSpots per organizations: 37
- Average number of HotSpots per organization: 5.5
- Average number of HotSpots exposing domain admin accounts: 3.3

The numbers above describe the attack surface of an average organization with respect to the HotSpot notion. The good news is there are on average at least 3-4 HotSpots that attackers must go through to compromise the entire network. These spots are bottlenecks that organizations can focus on and neutralize. The bad news is there are also at least 3-4 attack vectors that lead to domain compromise. Through those bottlenecks there could be numerous vectors. Therefore, organizations must monitor and closely control the privileged accounts associated with those HotSpots.

# HotSpot Live Detection & Mitigation

The process of identifying HotSpots requires several steps:

- The first step in the process is defining the privileged groups and accounts that will produce a ColdSpot where credentials are exposed.
- The second step is identifying these ColdSpots. This can be accomplished by using Windows Event Forwarding (WEF) to continuously monitor authentication attempts. The relevant security events that indicate authentication attempts are discussed below (under the PreCog tool section).

- A ColdSpot can also be created when a service like MySQL or SharePoint is associated with privileged accounts. In this case, the password will be stored in the host memory (LSASS) to allow the service continuous access to the credentials. Those hosts and services that are registered in ActiveDirectory with a Service Principal Name (SPN), which allows the service to rely on the Kerberos protocol, can be identified by an LDAP-SPN query (as is demonstrated in our RiskySPNs module).
- The third step focuses only on the ColdSpots. Once a ColdSpot is created, logs are pulled from the machine periodically to determine the following:
  - **Are the credentials exposed?** In some authentication scenarios, the password hash or Kerberos tokens are not forwarded to the machine, and therefore not exposed. For example, executing the Net Use command or accessing a network share does not expose the hash or Kerberos ticket on the target machine. The table below describes different connection scenarios and specifies whether the credentials are exposed.

| Logon type | # | Authenticators accepted | Reusable credentials in LSA session | Examples |
|---|---|---|---|---|
| Interactive (a.k.a., Logon locally) | 2 | Password, Smartcard, other | **Yes** | Console logon; RUNAS;<br>Hardware remote control solutions (such as<br>Network<br>KVM or Remote<br>Access / Lights-Out<br>Card in server)<br>IIS Basic Authn<br>(before IIS 6.0) |
| Network | 3 | Password, NT Hash, Kerberos ticket | No (except if delegation is enabled, then Kerberos tickets present) | NET USE;<br>RPC calls;<br>Remote registry;<br>IIS integrated<br>Windows authn;<br>SQL Windows authn; |
| Batch | 4 | Password (usually stored as LSA secret) | **Yes** | Scheduled tasks |
| Service | 5 | Password (usually stored as LSA secret) | **Yes** | Windows services |
| NetworkCleartext | 8 | Password | **Yes** | IIS Basic Authn (IIS<br><br>6.0 and newer); Windows PowerShell with CredSSP |
| NewCredentials | 9 | Password | **Yes** | RUNAS /NETWORK |
| RemoteInteractive | 10 | Password, Smartcard, other | **Yes** | Remote Desktop<br>(formerly known as “Terminal Services”) |

*Mitigating Pass-the-Hash (PtH) attacks and other credential theft techniques by Microsoft*

  - **Is it a HotSpot?** If credentials are exposed on the ColdSpot, we need to determine if there is another account that has admin privileges on the machine (but does not possess the same level of domain privileges as the account whose credentials are exposed) and is currently connected or connects while the credentials are exposed. To do this, we utilize WEF to monitor security event ID 4762 on the ColdSpot. If such an event is detected, a HotSpot is created as the exposed credentials are at imminent risk.

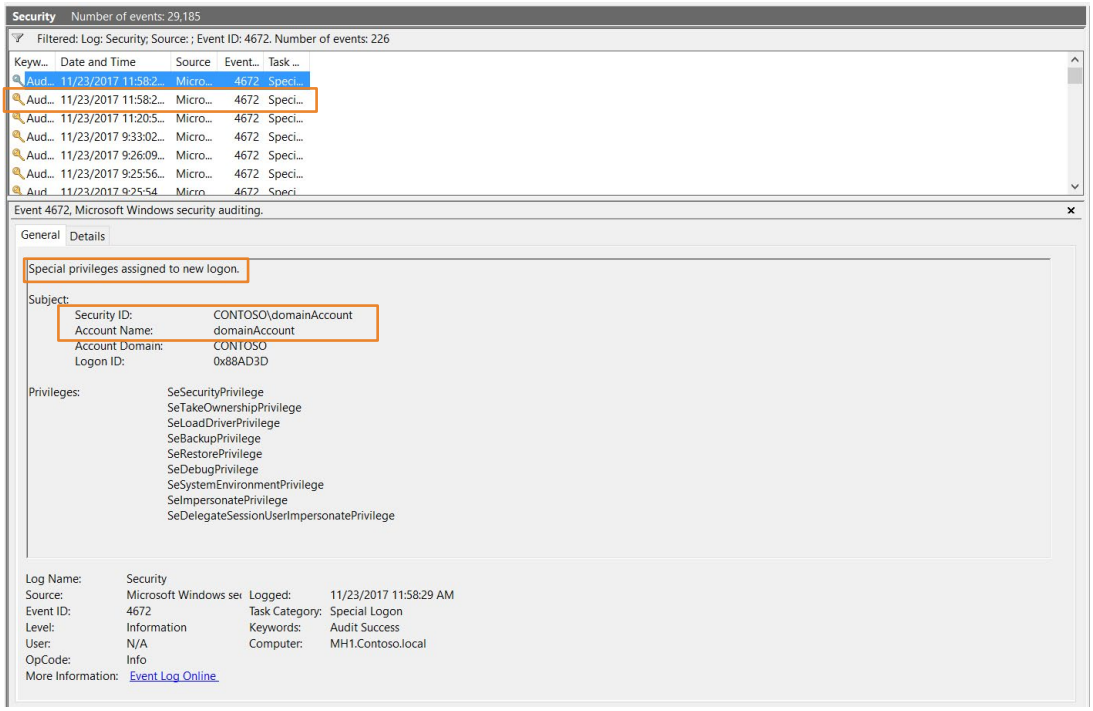



***Figure 7** – Event ID 4672 indicating a privileged access of domainAccount*

- **Are the sessions still alive?** After identifying the logon event, it is also crucial to identify when a session has ended - a HotSpot should be downgraded to ColdSpot after the less-privileged user has disconnected; or completely removed if the privileged user (whose credentials were exposed) has disconnected. This can be done by monitoring these logoff event logs: 4634 - "An account was logged off" and 4647 - "User initiated logoff." In addition, it is possible to monitor the machine's restarts with event logs like: 4608 - "Windows is starting up," 6005 - "Event Log service was started," 6006 - "The Event log service was stopped" and 6008 - "There was unexpected shutdown." Obviously when a machine restarts, all of its active logon sessions are terminated.

Upon detection of a HotSpot, there are two actions to take to mitigate current and future risk. Mitigation of current risk can be achieved by immediately disabling the privileged account or changing its password. This will obviously deny any potential compromise of the account or attempt to use it. It will also require system administrators to re-enable the account after initial investigation of the HotSpot incident.

Another action that should be considered is investigating the reason for the HotSpot creation. Networks that are segmented and limit the mixture of non-secure endpoints and privileged accounts should not have unintentional HotSpots. Hence, it is recommended to investigate the reason behind the mix of connections and take action accordingly. Either reduce the privileges of the privileged account if the ColdSpot is inevitable – necessary in cases where a machine must allow a mixture of privileged connections; reduce the privileges of the privileged accounts connecting to the previously identified HotSpot; or limit remote connections of the privileged / non-privileged accounts to the HotSpot and similar machines. In cases where those configuration changes cannot be made, apply targeted monitoring and targeted security on the host.

# The Tool: PreCog

## Background

To demonstrate the HotSpot concept, CyberArk Labs developed a tool called PreCog, which utilizes event logs from any monitored asset to identify HotSpots in real time as they are created. In identifying HotSpots, organizations can contain attackers and prevent future HotSpot creation.

It's available now on GitHub: https://github.com/cyberark/PreCog

The tool was designed and developed for security teams to improve the security posture of their networks. Aside from tracking live HotSpot creation and termination, we developed the past-HotSpots feature, which provides security teams with visibility into already terminated HotSpots; information that can be used to neutralize future HotSpots before they are created.

## Description

What is PreCog? PreCog is a PowerShell tool that identifies credential theft precognition by detecting HotSpots in the network. The tool analyzes event logs from domain-connected machines and follows the privileged account activity on those machines. The analysis identifies machine HotSpots that have open logon sessions from both tier 0 privileged accounts and accounts with local admin rights on the detected machine spot, which might be owned by a potential attacker. In neutralizing the HotSpot, the risk can be mitigated and future possible credential theft attempts prevented.

## PreCog in Action

After downloading the tool, you will get two folders and two scripts. The main script is "PreCog.ps1."

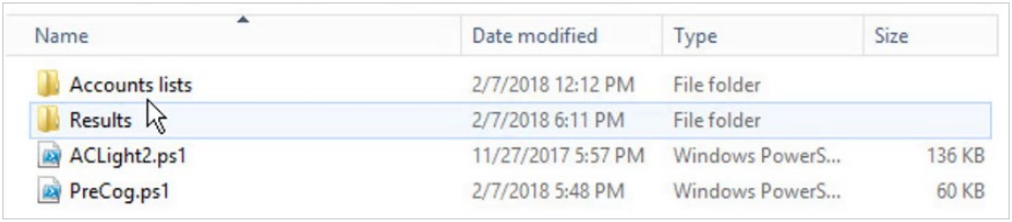



***Figure 8** – PreCog install structure*

PreCog queries the WEF (Windows Event Forwarding) server and analyzes four important events:

(i). 4624 - An account was successfully logged on.

(ii). 4672 - Special privileges assigned to new logon.

(iii). 4647 - User initiated logoff.

(iv). 4634 - An account was logged off.

These events provide PreCog with the ability to follow the logon sessions on each of the monitored machines. The tool also processes a few more events with the intention of detecting machines that were restarted, as their active logon sessions list should be reset. The event IDs that imply machine restart are: 4608 - "Windows is starting up," 6005 - "Event Log service was started," 6006 - "The Event log service was stopped" and 6008 - "There was unexpected shutdown."

## Running PreCog

You can easily run the tool with this default configuration from the PowerShell\CMD shell:

```
C:\HotSpots>. .\PreCog.ps1
```

***Figure 9** – Running PreCog*

When you start PreCog, it will show the privileged accounts that were loaded and will be monitored. It will look like this:

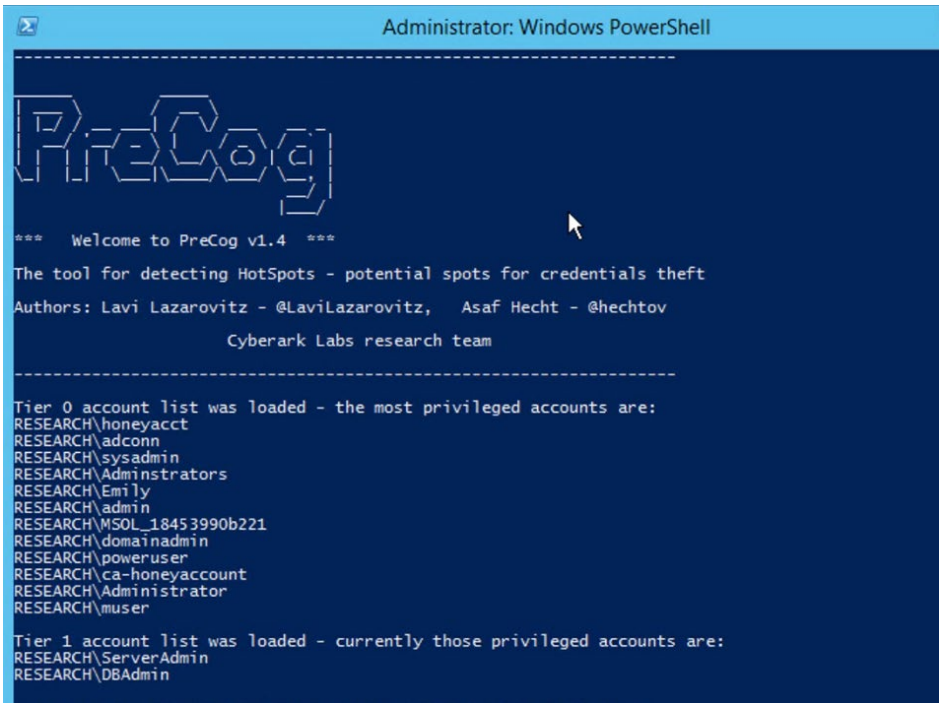



***Figure 10*** – *PreCog startup*

**Note** - on first execution of PreCog, the list of "Tier 0 - most privileged accounts.csv" will be created automatically. It will be done by running the "ACLight2" tool, which is a special discovery tool that will identify the network's most sensitive privileged accounts (more information on the ACLight tool could be seen in its official GitHub page: https://github.com/cyberark/ACLight and in the following blog post https://www.cyberark.com/threat-research-blog/shadow-admins-stealthy-accounts-fear/).

After PreCog loads the accounts, it goes on to the next step of analyzing the historic event logs. When past event log analysis is complete, it will continue to perform live monitoring of the logs.

## Detection

PreCog will automatically detect historic and live HotSpots. As you can see in our demo environment, PreCog pointed out different spots through the tool's command window:

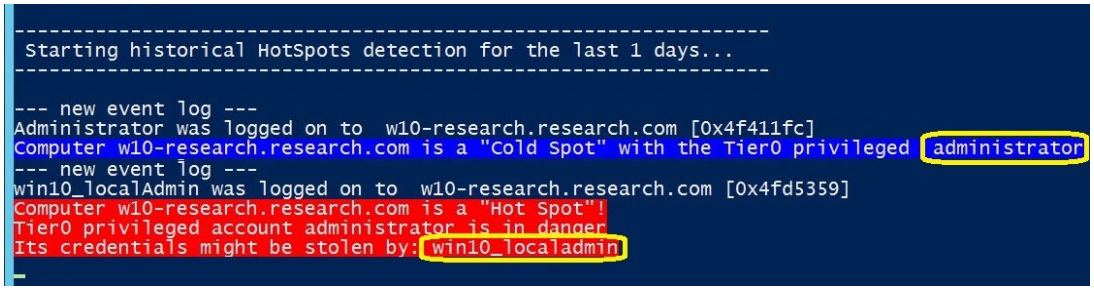



***Figure 11*** – *PreCog identifying HotSpots*

In addition, we can check the "Results" folder for details of the detected HotSpots:

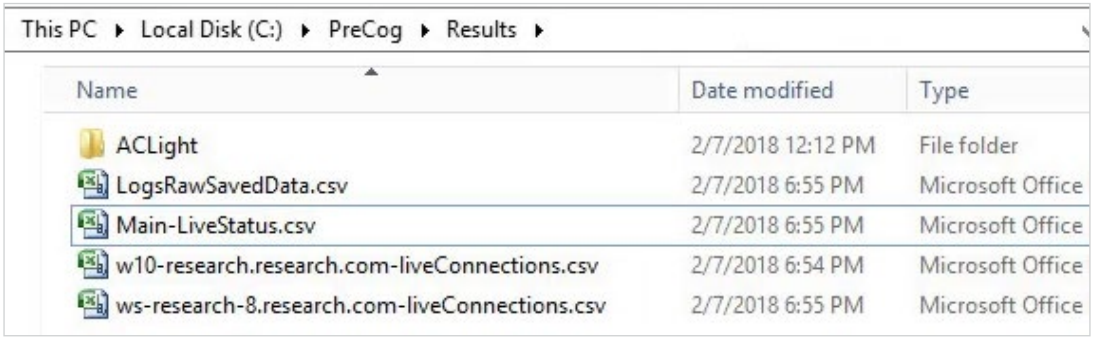



***Figure 12** – PreCog results folder*

The main results file is the "Main-LiveStatus.csv." In our demo environment, this file looks like this:


***Figure 13** – PreCog main results file*

From this file, the reader can see that PreCog discovered three interesting findings – including two benign ColdSpots and a very risky HotSpot. In that HotSpot, an attacker could have successfully stolen the higher tier 0 privileged account.

## Analysis

To better understand the results, let's review the findings in more details:

ColdSpots:

- The first ColdSpot is active on the "ws-research-8.research.com" machine. This machine is a ColdSpot, with the tier 0 privileged account "administrator" logged into it. But there is no other local admin connected to that machine at the same time, otherwise it would be elevated to a HotSpot – a potential machine for credential theft.
- The second, is a historical ColdSpot on the "w10-research.research.com" machine, as the new modified computer name implies "HISTORYspot->w10-research.research.com<-OLD." You can also see the exact time of creation and the existence duration of the ColdSpot.

HotSpot:

- There is a HotSpot on the "w10-research.research.com" machine.
- The result line of the HotSpot provides the rest of the details. We can see that on our windows 10 machine there are two sessions:
  - The first, is the privileged tier 0 "administrator" account.
  - The second, is non-tier 0 account - "win10_localadmin."
- In this situation, the "win10_localadmin" account can be used to compromise the other tier 0 administrator account as it has local admin rights on the windows 10 machine. This HotSpot put the whole network at risk because attackers could have previously compromised that endpoint, and now they might be able to escalate privileges and continue roaming the entire network.

This type of HotSpot should be avoided at all costs as it violates the network tiers defensive strategy. The isolation of the different tiers is intended to contain an attacker that somehow succeeded in compromising an account from one tier by preventing privilege escalation opportunities, which might allow access to other more sensitive tiers.

## Remediation

As a defensive measure, we should change the password of that "administrator" account because an attacker may have compromised it.

In addition, we should also take a look at the administrator's recent activities to make sure that in the time period before the password change, the attacker didn't perform other sensitive operations (like creating new admin accounts, adding permissions to other accounts and so on). For example, you can run the ACLight scan again - and make sure you recognize all the discovered tier 0 privileged accounts.

We encourage you to use the free PreCog tool to discover and eliminate spots of potential credential theft.

Full details and more explanations are available on PreCog are in the tool's GitHub page.

# Conclusion

The continuous race after new indicators of compromise, hashes of malicious files, known malicious code and scripts, and specific strings, is a race blue teamers are bound to lose. Attackers have the initiative in most cases, and defenders can usually only respond to those initiatives. After all this is why incident response teams are called "incident response."

Credential theft precognition brings to the table an effective approach that allows defenders to start that race with an advantage. This advantage, more specifically, is realized by detecting and neutralizing the existence of HotSpots and any future HotSpots that expose privileged credentials to potential compromise.

Attack techniques (TTP's) will continue to evolve and new tools will be created and added to the attacker arsenal. To deal with this ever-changing arsenal, defenders should adopt a TTP-agnostic approach such as credential theft precognition, to defend the most critical assets of the network, which are privileged accounts. The PreCog tool developed by CyberArk Labs demonstrates how this approach can be realized in Microsoft networked environments using standard monitoring and auditing tools. We are contributing this research and tool to the security community, and we welcome any feedback and contribution..

**References:**

Credential theft Precognition blog post

PreCog GitHub page: https://github.com/cyberark/PreCog

Use Windows Event Forwarding to help with intrusion detection:
https://docs.microsoft.com/en-us/windows/security/threat-protection/use-windows-event-forwarding-to-assist-in-intrusion-detection

Windows Event Collector:
https://msdn.microsoft.com/en-us/library/bb427443(v=vs.85).aspx

Windows Event Forwarding - resource center:
https://social.technet.microsoft.com/wiki/contents/articles/33895.windows-event-forwarding.aspx