Tenable Identity Exposure Indicators of Attack Reference Guide

Last Revised: March 11, 2025

Table of Contents

Indicators of Attack and the Active Directory **3**

Customize an Indicator of Attack 4

Advanced Audit Policies for Indicators of Attack 5

OS Credential Dumping: LSASS Memory 6

Suspicious DC Password Change 12

DCShadow 19

DCSync 23

DNSAdmins Exploitation 28

Domain Backup Key Extraction 34

Enumeration of Local Administrators 39

Golden Ticket 46

Kerberoasting 52

Massive Computers Reconnaissance 56

NTDS Extraction 65

Password Guessing 74

Password Spraying 78

PetitPotam 82

SAM Name Impersonation 86

Unauthenticated Kerberoasting 92

Zerologon Exploitation 97

Indicators of Attack and the Active Directory

Required license: Indicators of Attack

Tenable Identity Exposure's Indicators of Attack provide a reactive approach to detect an attack in real time. Tenable Identity Exposure leverages three sources of information to detect security incidents:

- Your Active Directory (AD) database
- The SYSVOL shared folder
- Windows Event Logs

Tenable Identity Exposure collects the insertion strings associated with the event IDs and processes them to determine whether or not the events represent an attack.

Indicators of Attack (IoAs) help identify potential threats. When they trigger alerts, you must investigate to determine whether the alerts indicate real attacks or if they are triggered by specific activities in your AD environment. Tenable Identity Exposure may flag certain routine tasks, such as backups or domain synchronization, due to how they operate in your environment. Tenable Identity Exposure's approach is to flag potential attacks while allowing you to fine-tune your security profile to filter out legitimate activities, such as backup tools, ensuring accurate threat detection.

For information, see Install Indicators of Attack.

Indicators of Attack

Each Indicator of Attack (IoA) requires specific audit policies that the installation script automatically enables.

Note: You must run the IoA installation script again if the configuration for the attack detection changes.

- OS Credential Dumping: LSASS Memory
- DCShadow
- DCSync
- DNSAdmins Exploitation

- Domain Backup Key Extraction
- Enumeration of Local Administrators
- Golden Ticket
- Kerberoasting
- Massive Computers Reconnaissance
- NTDS Extraction
- Password Guessing
- Password Spraying
- PetitPotam

Customize an Indicator of Attack

In Tenable Identity Exposure, you can customize an Indicator of Attack with available options. For best results, follow the recommended values for each option.

To customize an indicator:

1. In Tenable Identity Exposure, click **Accounts** > **Security profiles management**.

   The **Security profiles management** pane appears.

2. In the list of security profiles, hover over the security profile that contains the indicator you want to customize. Click on the ✎ icon at the end of the line where the security profile file name appears.

   The **Profile configuration** pane appears.

3. Select the tab for **Indicators of Exposure** or **Indicators of Attack**.
4. (Optional) In the **Search an indicator** box, type an indicator name.
5. Click the name of the an indicator to customize.

   The **Indicator Customization** pane appears.

6. Select the options from the Options table.

Tip: To enable the **aggressive mode** for Indicators of Attack, click the toggle button for the option "Aggressive mode" to "Yes."

Tip: Certain indicator options require the use of regular expressions (regex). Regex are a 'contain' match instead of an 'equal' match.
- To get an exact match, you must use Regex special characters (`"^...$"`) syntax.
- You must also escape special characters with a backslash when using regex. Example: To declare `"domain\user"` and `"CN=Vincent C (Test),DC=tenable,DC=corp"`, you type `"domain\\user"` and `"CN=Vincent C. \(Test\),DC=tenable,DC=corp"`.

7. Click **Save as draft**.

   A message confirms that Tenable Identity Exposure saved the customization options.

Advanced Audit Policies for Indicators of Attack

This section outlines how to leverage audit policies to detect and analyze Indicators of Attack (IoA) within your network.

By implementing these advanced settings, organizations can enhance visibility, quickly identify malicious activity, and respond effectively to potential threats.

Audit Policy	Setting
Audit Credential Validation	Success, Failure
Audit Kerberos Authentication Service	Success, Failure
Audit Kerberos Service Ticket Operations	Success, Failure
Audit Computer Account Management	Success
Audit Security Group Management	Success
Audit Process Creation	Success
Audit Process Termination	Success
Audit Directory Service Access	Success
Audit Logoff	Success

Audit Policy	Setting
Audit Logon	Success, Failure
Audit Application Generated	Success
Audit Detailed File Share	Success
Audit Handle Manipulation	Success
Audit Other Object Access Events	Success
Audit SAM	Success
Audit Sensitive Privilege Use	Success, Failure

OS Credential Dumping: LSASS Memory

After a user logs on, attackers can attempt to access credential material stored in the process memory of the Local Security Authority Subsystem Service (LSASS). The NT AUTHORITY\SYSTEM account is a built-in local account with the highest privileges on a Windows system. When this account is identified as the origin of an attack, it indicates that the activity is occurring locally and directly on the machine itself, using the localhost interface. However, it may have been remotely triggered by another process, making the reported source IP potentially inaccurate.

Events Auditing Policy

Event IDs	Audit Policies	Value
4624	├ Category: Logon/Logoff └ – Sub-category: Logon	Success
1	Sysmon - Process creation	Sysmon - N/A
8	Sysmon - CreateRemoteThread	Sysmon - N/A
10	Sysmon - ProcessAccess	Sysmon - N/A

	Requires Sysmon extension For information on how to install and configure Sysmon, see Install Microsoft Sysmon.	Yes

Options

To select options for this Indicator of Attack, see Customize an Indicator of Attack.

Option Name (Type, Default Value)	
Aggressive mode (Boolean, False)	
Description	Recognizes the tool and considers only the predefined processes as non-legitimate. Otherwise, it considers all attack tools as non-legitimate unless they are in the allowlist.
Recommendation	Do not change this value unless you have a specific need.
Allowed processes in aggressive mode (List of strings, list of processes)	
Description	List of allowed processes used for malicious actions. This allowlist supports multiple inputs, such as: full process path, process name only, or the specific technique that the IoA identified.
Recommendation	Use the exact string indicated by an attack, but confirm beforehand that it's a false positive and the alert is linked to a genuine application in your environment. This list applies only with "Aggressive" mode enabled.
Basic mode — Deny list (List of strings, list of processes)	
Description	Specifies the tools that trigger attacks in basic mode: `mimikatz`, `taskmgr`, `ipconfig`, `arp`, `powershell`, `net`, `auditpol`, `whoami`, `cmd`, `route`, `processhacker`, `net1`, `csc`, `procdump`, `osqueryi`.
Recommendation	Adapt the list to your environment, especially if you're using administrative tools that attackers can abuse.
Allow unknown source (Boolean, True)	

Description	Allow the display of attacks when their sources only show an IP address and not a hostname.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted source hostnames (List of strings, Empty)	
Description	Allow attacks from these hostnames.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted source hostnames (List of strings, Empty)	
Description	Allow attacks from these hostnames.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted source IPs (List of strings, Empty)	
Description	Allow attacks from these IP addresses.
Recommendation	Do not change this value unless you have a specific need. This IoA can identify accounts that use security tools such as EDR or AV as malicious and trigger an attack. Add the source IP addresses of the machines running these tools to this list to classify them as legitimate. Use `WhiteListedProcesses` first in this IoA to allow only specific tools as they are more precise, ensuring efficient detection of true attacks.
Whitelisted target domain controllers (List of strings, Empty)	
Description	Whitelist domain controllers from detected attack results by providing their names in the LDAP Common Name (CN) format.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted usernames (List of strings, Empty)	
Description	Allow attacks associated with these usernames.
Recommendation	Do not change this value unless you have a specific need.

INDICATOR CUSTOMIZATION

Name of the indicator: **OS Credential Dumping: LSASS Memory**

Global customization

Search an option

Apply on: **On all directories**

Severity level: Critical

Indicates the importance and criticality of this indicator of attack.

Aggressive mode: No

When activated, the IoA engine will detect more events as attacks (broader attack detection, but with a risk of false positives).

Allowed processes in aggressive mode

List of allowed processes potentially used for malicious actions. This allow list supports multiple inputs: the full process path, just the process name, or the specific technique that the indicator identified. The safest option is to use the exact string indicated by an attack, but only after confirming it's a false positive and the alert is linked to a genuine application in your environment. This list is only used in aggressive mode.

Basic mode - Deny list

- mimikatz
- taskmgr
- ipconfig
- arp
- powershell
- net
- auditpol
- whoami
- cmd
- route
- processhacker
- net1
- csc
- procdump

On basic mode, only these tools will trigger attacks.

Allow unknown source: Yes

Allow the display of attacks when their sources only show an IP address and not a hostname.

Yara Detection Rules

YARA rules are malware detection patterns that you can customize to identify targeted attacks and security threats specific to your environment.

OS Credential Dumping Yara Detection Rules

```
rule mimikatz
{
    meta:
        description    = "mimikatz"
        author         = "Benjamin DELPY (gentilkiwi)"
        tool_author    = "Benjamin DELPY (gentilkiwi)"

    strings:
        $exe_x86_1     = { 89 71 04 89 [0-3] 30 8d 04 bd }
        $exe_x86_2     = { 8b 4d e? 8b 45 f4 89 75 e? 89 01 85 ff 74 }

        $exe_x64_1     = { 33 ff 4? 89 37 4? 8b f3 45 85 c? 74}
        $exe_x64_2     = { 4c 8b df 49 [0-3] c1 e3 04 48 [0-3] 8b cb 4c 03 [0-3] d8 }

        $dll_1         = { c7 0? 00 00 01 00 [4-14] c7 0? 01 00 00 00 }
        $dll_2         = { c7 0? 10 02 00 00 ?? 89 4? }

        $sys_x86       = { a0 00 00 00 24 02 00 00 40 00 00 00 [0-4] b8 00 00 00 6c 02 00 00 40 00 00
00 }
        $sys_x64       = { 88 01 00 00 3c 04 00 00 40 00 00 00 [0-4] e8 02 00 00 f8 02 00 00 40 00 00
00 }

    condition:
        (all of ($exe_x86_*)) or (all of ($exe_x64_*)) or (all of ($dll_*)) or (any of ($sys_*))
}

rule invoke_mimikatz
{
    meta:
        description = "Detects Invoke-Mimikatz"
        author      = "Tenable.AD"
        reference1  = "https://github.com/EmpireProject/Empire/blob/master/data/module_
source/credentials/Invoke-Mimikatz.ps1"
        reference2  = "https://github.com/BC-SECURITY/Empire/blob/master/data/module_
source/credentials/Invoke-Mimikatz.ps1"
        date        = "2020-11-09"

    strings:
        $sekurlsa   = "sekurlsa" wide base64
        $lsadump    = "lsadump" wide base64

    condition:
      all of them
}

rule pypykatz
{
    meta:
        description = "Detects Pypykatz"
        author      = "Tenable.AD"
        reference1  = "https://github.com/skelsec/pypykatz"
```

```
        date        = "2020-11-10"

    strings:
        $pypykatz   = "pypykatz"

    condition:
        any of them
}

rule create_remote_thread
{
    meta:
        description             = "Detects CreateRemoteThread"
        author                  = "Tenable.AD"

    strings:
        $createremotethreadwide = "CreateRemoteThread" wide nocase
        $createremotethread     = "CreateRemoteThread"

    condition:
        any of them
}

rule mimikatz_lsass_mdmp
{
  meta:
    description    = "LSASS minidump file for mimikatz"
    author         = "Benjamin DELPY (gentilkiwi)"

  strings:
    $lsass         = "System32\\lsass.exe" wide nocase

  condition:
    (uint32(0) == 0x504d444d) and $lsass
}
```

See also

- Suspicious DC Password Change
- DCShadow
- DCSync
- OS Credential Dumping: LSASS Memory
- Domain Backup Key Extraction
- Enumeration of Local Administrators
- Golden Ticket
- Kerberoasting

- Massive Computers Reconnaissance
- NTDS Extraction
- Password Guessing
- Password Spraying
- PetitPotam
- SAM Name Impersonation
- Unauthenticated Kerberoasting
- Zerologon Exploitation

Suspicious DC Password Change

Related to the Zerologon Exploitation Indicator of Attack, this IoA focuses on a **specific** post-exploitation activity that attackers commonly use in conjunction with the Netlogon vulnerability: the modification of the Domain Controller machine account password.

Detection Type	Related to a Common Vulnerabilities and Exposures (CVE)	Available from Tenable Identity Exposure version
Generic IOC	Yes (CVE-2020-1472)	3.46

How the attack works

The attack exploits a cryptographic flaw in the function used to initiate the Netlogon secure channel.

By setting certain fields to null bytes (0), an attacker can quickly make multiple requests (in a few seconds) to a server and bypass authentication without any prior valid identifiers. Once authenticated, the attacker can modify passwords for computer accounts or domain controllers and gain high privilege and persistence on the domain.

After resetting the domain controller machine account, the attacker can perform a DCSync attack to obtain privileged hashes.

The password reset affects domain controller authentication. Existing Kerberos tickets continue to work until they expire, but new ones will not because of their encryption with the new empty password. Similarly, the NTLM protocol cannot establish a secure channel.

This is due to the storage of the domain controller password in two different places:

- HKLM\SECURITY\Policy\Secrets$machine.ACC
- NTDS.DIT database on the DCs

However, the attack only resets the password stored in the NTDS.DIT database and not the one in the Registry. To resolve the inconsistency, you can either restore the previous password or set a new one in both locations after the attack.

How the IoA works

The Suspicious DC Password Change IoA identifies a post-exploitation activity related to the Netlogon vulnerability, namely the modification of the password for the domain controller machine account. To do this, the IoA examines event log 4742 for any anomalous circumstances surrounding a password reset, as well as Netlogon event 5823 to determine the reason for the reset. It also searches remote authentication attempts that use the reset machine account for any indication of suspicious activity.

Specific modifications to the environment

None. Tenable Identity Exposure adapts the audit policy to meet the needs of the required Windows event logs.

Events Auditing Policy

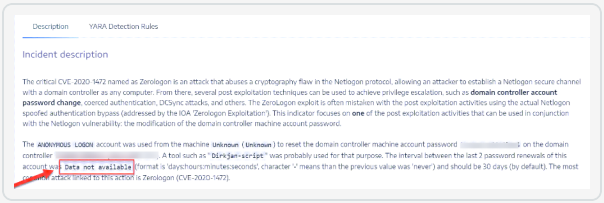
Provider Name	Channel	Event IDs	Audit Policies	Value
NETLOGON	System	5823	N/A	Success
Microsoft-Windows-Security-Auditing	Security	4742	├ Category: Account Management └ — Sub-category: Computer Account Management	Success

Microsoft-Windows-Security-Auditing	Security	4624	├ Category: Logon/Logoff └ – Sub-category: Audit Logon	Success
Other requirements				
Sysmon extension	No			
Honey Account	No			

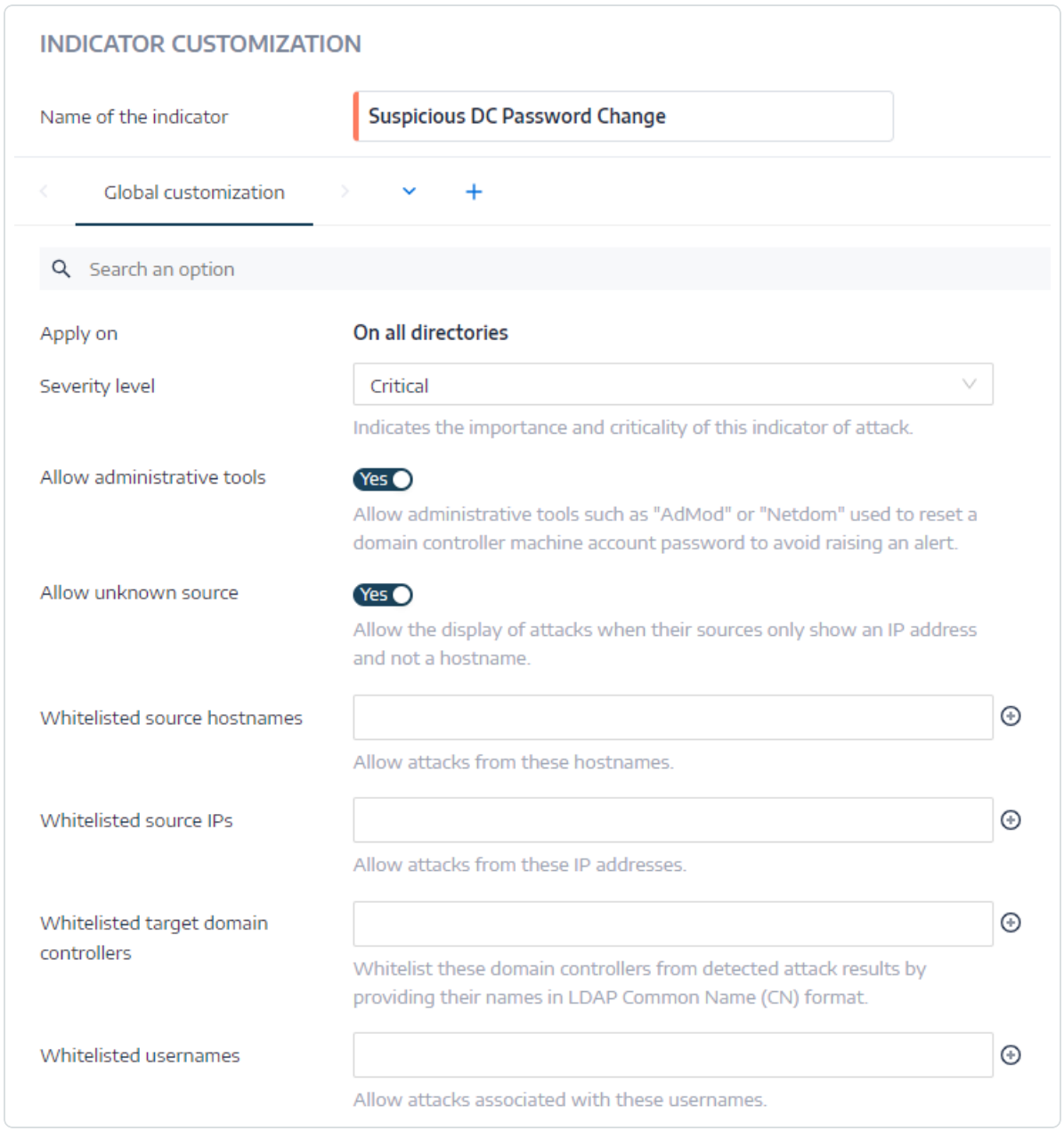
Options

To select options for this Indicator of Attack, see Customize an Indicator of Attack.

Option Name (Type, Default Value)	
Aggressive mode (Boolean, False)	
Description	Detects the attack whether or not the user is authenticated or not, whereas the basic mode (default) only detects authenticated users.
Recommendation	Do not change this value unless you have a specific need. Enabling this option allows a more aggressive detection, but can also lead to false positives.
Password Change Interval (Integer, 30)	
Description	Specifies the period between two password changes (30 days by default). This option is only useful when you enable the aggressive mode. A **password interval** represents the time gap between two consecutive password changes on a system. For example, if a password was changed on Monday and again on Wednesday, the password interval would be 2 days. In the context of the Suspicious DC Password Change detection in Tenable Identity Exposure, this interval is critical for identifying abnormal or frequent password changes, which might indicate a security issue.

	However, in certain cases, Tenable Identity Exposure shows "Data not available" for the password interval. This happens due to the following reasons: 1. During initial startup: When Tenable Identity Exposure starts for the first time, it only has access to the most recent password change event, but lacks data on previous password changes. Since calculating the interval requires at least two password change records, Tenable Identity Exposure cannot determine the interval, resulting in "Data not available". 2. When password changes occur while Tenable Identity Exposure is not running: If multiple password changes happen while Tenable Identity Exposure is offline, it only records the most recent event upon resuming operation. Without earlier change events, Tenable Identity Exposure cannot compute the interval, and thus, displays "Data not available" in the incident description. 
Recommendation	Do not change this value unless you have a specific need. Enabling this option allows a more aggressive detection, but can also lead to false positives.
Allow administrative tools (Boolean, True)	
Description	Allows administrative tools such as "AdMod" or "Netdom" used to reset a domain controller machine account password to avoid raising an alert.
Recommendation	Do not change this value unless you have a specific need. Enabling this option allows a more aggressive detection, but can also lead to false positives.

Allow unknown source (Boolean, True)	
Description	Allow the display of attacks when their sources only show an IP address and not a hostname.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted source hostnames (List of strings, Empty)	
Description	Allow attacks from these hostnames.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted source IPs (List of strings, Empty)	
Description	Allow attacks from these IP addresses.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted target domain controllers (List of strings, Empty)	
Description	Whitelist domain controllers from detected attack results by providing their names in LDAP Common Name (CN) format.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted usernames (List of strings, Empty)	
Description	Allow attacks associated with these usernames.
Recommendation	Do not change this value unless you have a specific need.



Yara Detection Rules

YARA rules are malware detection patterns that you can customize to identify targeted attacks and security threats specific to your environment.

```
rule mimikatz
{
    meta:
        description     = "mimikatz"
        author          = "Benjamin DELPY (gentilkiwi)"
```

```
        tool_author    = "Benjamin DELPY (gentilkiwi)"

    strings:
        $exe_x86_1     = { 89 71 04 89 [0-3] 30 8d 04 bd }
        $exe_x86_2     = { 8b 4d e? 8b 45 f4 89 75 e? 89 01 85 ff 74 }

        $exe_x64_1     = { 33 ff 4? 89 37 4? 8b f3 45 85 c? 74}
        $exe_x64_2     = { 4c 8b df 49 [0-3] c1 e3 04 48 [0-3] 8b cb 4c 03 [0-3] d8 }

        $dll_1         = { c7 0? 00 00 01 00 [4-14] c7 0? 01 00 00 00 }
        $dll_2         = { c7 0? 10 02 00 00 ?? 89 4? }

        $sys_x86       = { a0 00 00 00 24 02 00 00 40 00 00 00 [0-4] b8 00 00 00 6c 02 00 00 40 00 00
00 }
        $sys_x64       = { 88 01 00 00 3c 04 00 00 40 00 00 00 [0-4] e8 02 00 00 f8 02 00 00 40 00 00
00 }

    condition:
        (all of ($exe_x86_*)) or (all of ($exe_x64_*)) or (all of ($dll_*)) or (any of ($sys_*))
}

rule cve_2020_1472
{
   meta:
      description = "Detects Dirkjan cve-2020-1472.py script"
      author = "Tenable.AD"
      reference1 = "https://github.com/dirkjanm/CVE-2020-1472/blob/master/cve-2020-1472-exploit.py"
      date = "2023-03-22"
      hash1 = "50AF4367EADD55236D085D8221815EA06992D6C0E1AB3ED6848DC3BDACA6F7DD"
   strings:
      $x1 = "0x212fffff" fullword
      $x2 = "NETLOGON_SECURE_CHANNEL_TYPE.ServerSecureChannel" fullword
      $x3 = "NetrServerPasswordSet2" fullword
   condition:
      $x1 and $x2 and $x3
}
```

See also

- OS Credential Dumping: LSASS Memory
- DCShadow
- DCSync
- Suspicious DC Password Change
- Domain Backup Key Extraction
- Enumeration of Local Administrators
- Golden Ticket

- Kerberoasting
- Massive Computers Reconnaissance
- NTDS Extraction
- Password Guessing
- Password Spraying
- PetitPotam
- SAM Name Impersonation
- Unauthenticated Kerberoasting
- Zerologon Exploitation

DCShadow

DCShadow is another late-stage kill chain attack that allows an attacker with privileged credentials to register a rogue domain controller in order to push changes to a domain via domain replication.

Events Auditing Policy

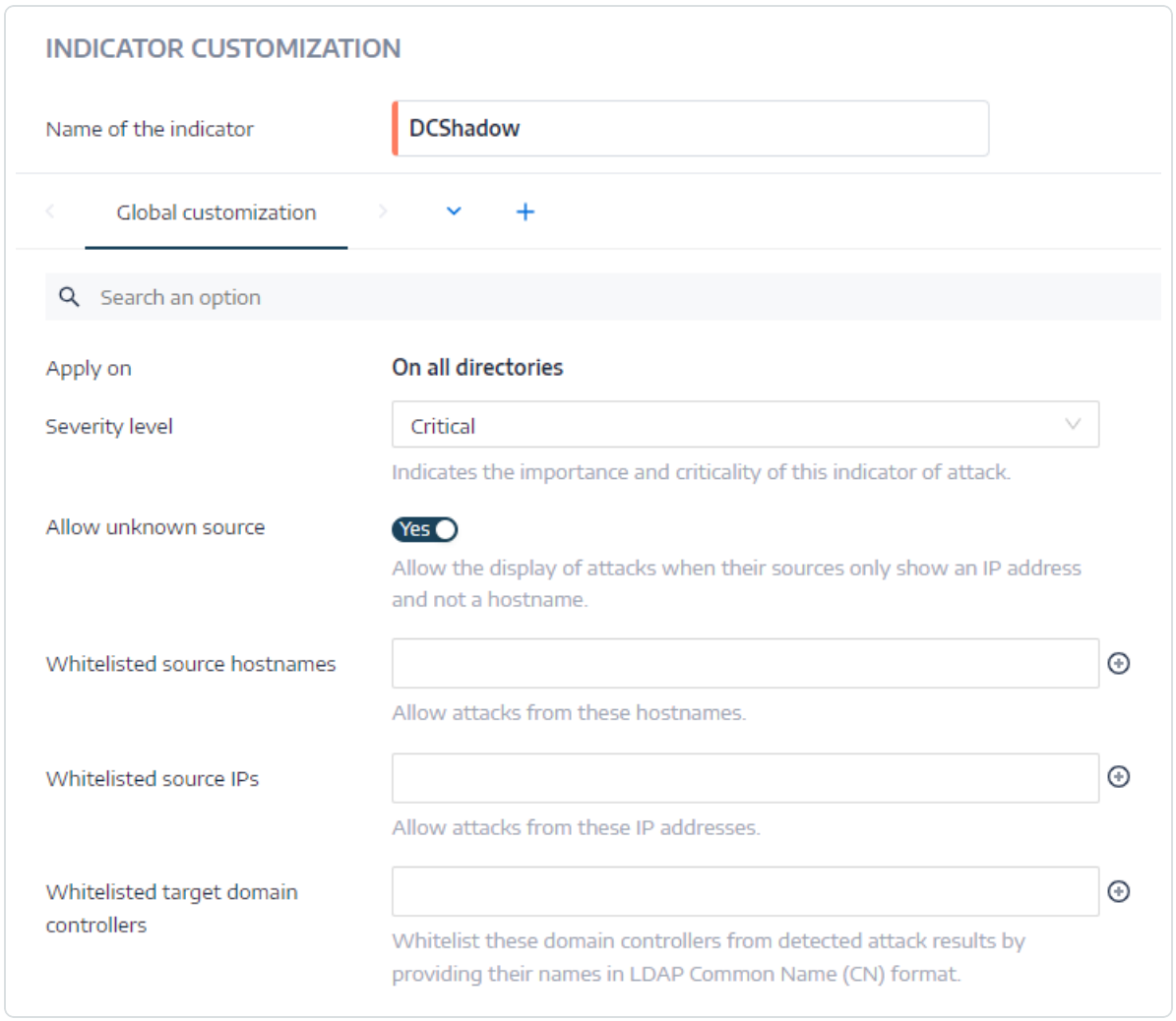
Event IDs	Audit Policies	Value
4624	├ Category: Logon/Logoff └ – Sub-category: Logon	Success
4662	├ Category: Directory Service (DS Access) └ – Sub-category: Directory Service Access	Success
	Requires Sysmon extension	No
	Requires honey account	No

Options

To select options for this Indicator of Attack, see Customize an Indicator of Attack.

Option Name (Type, Default Value)

Allow unknown source (Boolean, True)	
Description	Allow the display of attacks when their sources only show an IP address and not a hostname.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted source hostnames (List of strings, Empty)	
Description	Allow attacks from these hostnames.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted source IPs (List of strings, Empty)	
Description	Allow attacks from these IP addresses.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted target domain controllers (List of strings, Empty)	
Description	Whitelist domain controllers from detected attack results by providing their names in the LDAP Common Name (CN) format.
Recommendation	Do not change this value unless you have a specific need.



Yara Detection Rules

YARA rules are malware detection patterns that you can customize to identify targeted attacks and security threats specific to your environment.

DC Shadow Yara Detection Rules

```
rule mimikatz
{
    meta:
        description    = "mimikatz"
        author         = "Benjamin DELPY (gentilkiwi)"
        tool_author    = "Benjamin DELPY (gentilkiwi)"

    strings:
        $exe_x86_1     = { 89 71 04 89 [0-3] 30 8d 04 bd }
        $exe_x86_2     = { 8b 4d e? 8b 45 f4 89 75 e? 89 01 85 ff 74 }
```

```
        $exe_x64_1      = { 33 ff 4? 89 37 4? 8b f3 45 85 c? 74}
        $exe_x64_2      = { 4c 8b df 49 [0-3] c1 e3 04 48 [0-3] 8b cb 4c 03 [0-3] d8 }

        $dll_1          = { c7 0? 00 00 01 00 [4-14] c7 0? 01 00 00 00 }
        $dll_2          = { c7 0? 10 02 00 00 ?? 89 4? }

        $sys_x86        = { a0 00 00 00 24 02 00 00 40 00 00 00 [0-4] b8 00 00 00 6c 02 00 00 40 00 00
00 }
        $sys_x64        = { 88 01 00 00 3c 04 00 00 40 00 00 00 [0-4] e8 02 00 00 f8 02 00 00 40 00 00
00 }

    condition:
        (all of ($exe_x86_*)) or (all of ($exe_x64_*)) or (all of ($dll_*)) or (any of ($sys_*))
}

rule invoke_mimikatz
{
    meta:
        description = "Detects Invoke-Mimikatz"
        author      = "Tenable.AD"
        reference1  = "https://github.com/EmpireProject/Empire/blob/master/data/module_
source/credentials/Invoke-Mimikatz.ps1"
        reference2  = "https://github.com/BC-SECURITY/Empire/blob/master/data/module_
source/credentials/Invoke-Mimikatz.ps1"
        date        = "2020-11-09"

    strings:
        $sekurlsa   = "sekurlsa" wide base64
        $lsadump    = "lsadump" wide base64

    condition:
        all of them
}
```

See also

- OS Credential Dumping: LSASS Memory
- Suspicious DC Password Change
- DCSync
- DCShadow
- Domain Backup Key Extraction
- Enumeration of Local Administrators
- Golden Ticket
- Kerberoasting
- Massive Computers Reconnaissance

- NTDS Extraction
- Password Guessing
- Password Spraying
- PetitPotam
- SAM Name Impersonation
- Unauthenticated Kerberoasting
- Zerologon Exploitation

DCSync

The DCSync command in Mimikatz allows an attacker to pretend to be a domain controller and retrieve password hashes from other domain controllers, without executing any code on the target.

Events Auditing Policy

Event IDs	Audit Policies	Value
4624	├ Category: Logon/Logoff └ – Sub-category: Logon	Success
4662	├ Category: Directory Service (DS Access) └ – Sub-category: Directory Service Access	Success
	Requires Sysmon extension	No
	Requires honey account	No

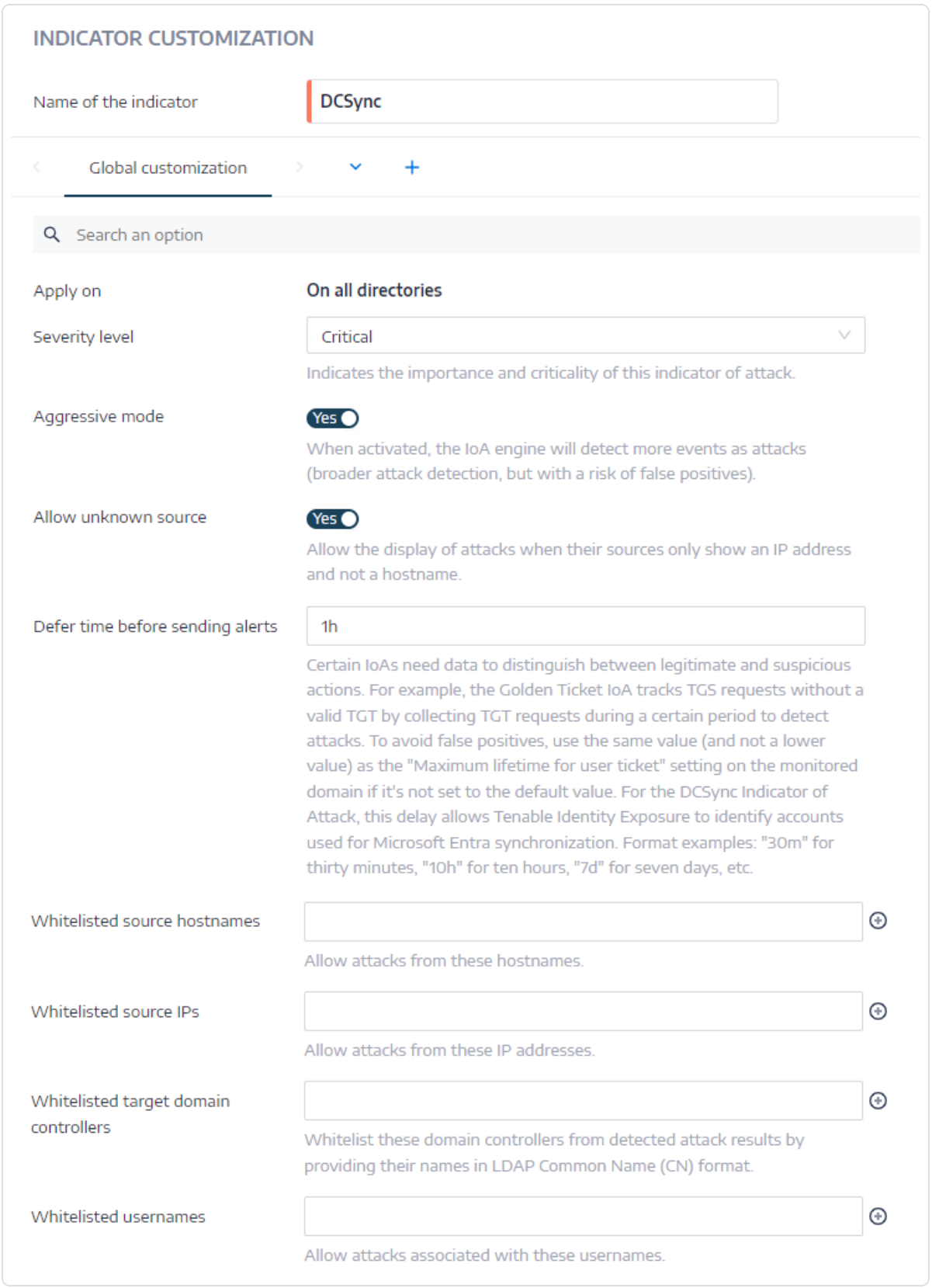
Options

To select options for this Indicator of Attack, see Customize an Indicator of Attack.

Option Name (Type, Default Value)
Aggressive mode (Boolean, False)

Description	If enabled, triggers an attack even if the machine is in the domain.
Recommendation	Do not change this value unless you have a specific need.
Allow unknown source (Boolean, True)	
Description	Allow the display of attacks when their sources only show an IP address and not a hostname.
Recommendation	Do not change this value unless you have a specific need.
Defer time before sending alerts (String, 12h)	
Description	Certain IoAs need data to distinguish between legitimate and suspicious actions. For example, the Golden Ticket IoA tracks TGS requests without a valid TGT by collecting TGT requests during a certain period to detect attacks. To avoid false positives, use the same value (and not a lower value) as the "Maximum lifetime for user ticket" setting on the monitored domain if it's not set to the default value. For the DCSync Indicator of Attack, this delay allows Tenable Identity Exposure to identify accounts used for AzureAD synchronization. Format examples: "30m" for thirty minutes, "10h" for ten hours, "7d" for seven days, etc.
Recommendation	10m Automatically whitelist accounts using DCSync-like methods. To whitelist attacks, use deterministic approaches such as "WhiteListedUserNames", "WhiteListedSourceIPs", or "WhiteListedSourceHostNames" options.
Whitelisted source hostnames (List of strings, Empty)	
Description	Allow attacks from these hostnames.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted source IPs (List of strings, Empty)	
Description	Allow attacks from these IP addresses.

Recommendation	Do not change this value unless you have a specific need.
Whitelisted target domain controllers (List of strings, Empty)	
Description	Whitelist domain controllers from detected attack results by providing their names in the LDAP Common Name (CN) format.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted usernames (List of strings, Empty)	
Description	Allow attacks associated with these usernames.
Recommendation	Do not change this value unless you have a specific need.



Yara Detection Rules

YARA rules are malware detection patterns that you can customize to identify targeted attacks and security threats specific to your environment.

DCSync Yara Detection Rules

```
rule mimikatz
{
    meta:
        description    = "mimikatz"
        author         = "Benjamin DELPY (gentilkiwi)"
        tool_author    = "Benjamin DELPY (gentilkiwi)"

    strings:
        $exe_x86_1     = { 89 71 04 89 [0-3] 30 8d 04 bd }
        $exe_x86_2     = { 8b 4d e? 8b 45 f4 89 75 e? 89 01 85 ff 74 }

        $exe_x64_1     = { 33 ff 4? 89 37 4? 8b f3 45 85 c? 74}
        $exe_x64_2     = { 4c 8b df 49 [0-3] c1 e3 04 48 [0-3] 8b cb 4c 03 [0-3] d8 }

        $dll_1         = { c7 0? 00 00 01 00 [4-14] c7 0? 01 00 00 00 }
        $dll_2         = { c7 0? 10 02 00 00 ?? 89 4? }

        $sys_x86       = { a0 00 00 00 24 02 00 00 40 00 00 00 [0-4] b8 00 00 00 6c 02 00 00 40 00 00
00 }
        $sys_x64       = { 88 01 00 00 3c 04 00 00 40 00 00 00 [0-4] e8 02 00 00 f8 02 00 00 40 00 00
00 }

    condition:
        (all of ($exe_x86_*)) or (all of ($exe_x64_*)) or (all of ($dll_*)) or (any of ($sys_*))
}

rule invoke_mimikatz
{
    meta:
        description = "Detects Invoke-Mimikatz"
        author      = "Tenable.AD"
        reference1  = "https://github.com/EmpireProject/Empire/blob/master/data/module_
source/credentials/Invoke-Mimikatz.ps1"
        reference2  = "https://github.com/BC-SECURITY/Empire/blob/master/data/module_
source/credentials/Invoke-Mimikatz.ps1"
        date        = "2020-11-09"

    strings:
        $sekurlsa   = "sekurlsa" wide base64
        $lsadump    = "lsadump" wide base64

    condition:
        all of them
}

rule secretsdump
{
    meta:
        description     = "Detects secretsdump"
        author          = "Tenable.AD"
        reference       =
"https://github.com/SecureAuthCorp/impacket/blob/master/examples/secretsdump.py"
        date            = "2020-11-09"
```

```
    strings:
        $impacket       = "impacket"
        $secretsdump    = "secretsdump"

    condition:
        all of them
}
```

See also

- OS Credential Dumping: LSASS Memory
- Suspicious DC Password Change
- DCShadow
- DCSync
- Domain Backup Key Extraction
- Enumeration of Local Administrators
- Golden Ticket
- Kerberoasting
- Massive Computers Reconnaissance
- NTDS Extraction
- Password Guessing
- Password Spraying
- PetitPotam
- SAM Name Impersonation
- Unauthenticated Kerberoasting
- Zerologon Exploitation

DNSAdmins Exploitation

DNSAdmins exploitation is a well-known attack that allows members of the DNSAdmins group to take over control of a Domain Controller running the Microsoft DNS service.

A member of the DNSAdmins group has the rights to perform administrative tasks on the Active Directory DNS service. Attackers can abuse these rights to execute malicious code in a highly privileged context.

Detection Type	Related to a Common Vulnerabilities and Exposures (CVE)	Available from Tenable Identity Exposure version
Generic IOC	No	3.21

How the attack works

The attacker must be a member of the DnsAdmins group or have write access to a DNS server object.

According to Microsoft protocol specifications, the attacker can load an arbitrary DLL — without a verification of the DLL path — by editing the `ServerLevelPluginDll` registry key.

The Microsoft administration tool *dnscmd.exe* then implements this option: `dnscmd.exe /config /serverlevelplugindll \\path\to\dll`.

When the system executes this command, it populates the following registry key: `HKLM\SYSTEM\CurrentControlSet\services\DNS\Parameters\ServerLevelPluginDll`.

Then, when the DNS service restarts, it loads the DLL provided and executes the malicious code.

How the IoA works

The DNSAdmins Indicator of Attack (IoA) can detect the successful editing of the dangerous registry key `ServerLevelPluginDll`. This IoA identifies this first step in the DNSAdmins exploitation attack before the system loads and executes the malicious DLL.

It provides the security team with the malicious DLL path to let the defending team perform further investigation.

Specific modifications to the environment

To have the required DNS audit logs, a Windows Server 2012 R2 domain controller must have the hotfix KB2956577 installed. There is nothing to do for Windows Server 2016 and later versions.

For this IoA to detect DNSAdmins exploitation, the IoA script automatically enables the "Microsoft-Windows-DNSServer/Audit" channel by adding the registry key `Microsoft-Windows-DNSServer/Audit` to `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog`.

Note: If you previously configured log retention for this specific channel, adding this registry key overrides the initial configuration. Previous events before this configuration are no longer visible.

Note: Tenable Identity Exposure supports all operating systems from Windows Server 2012 R2.

Events Auditing Policy

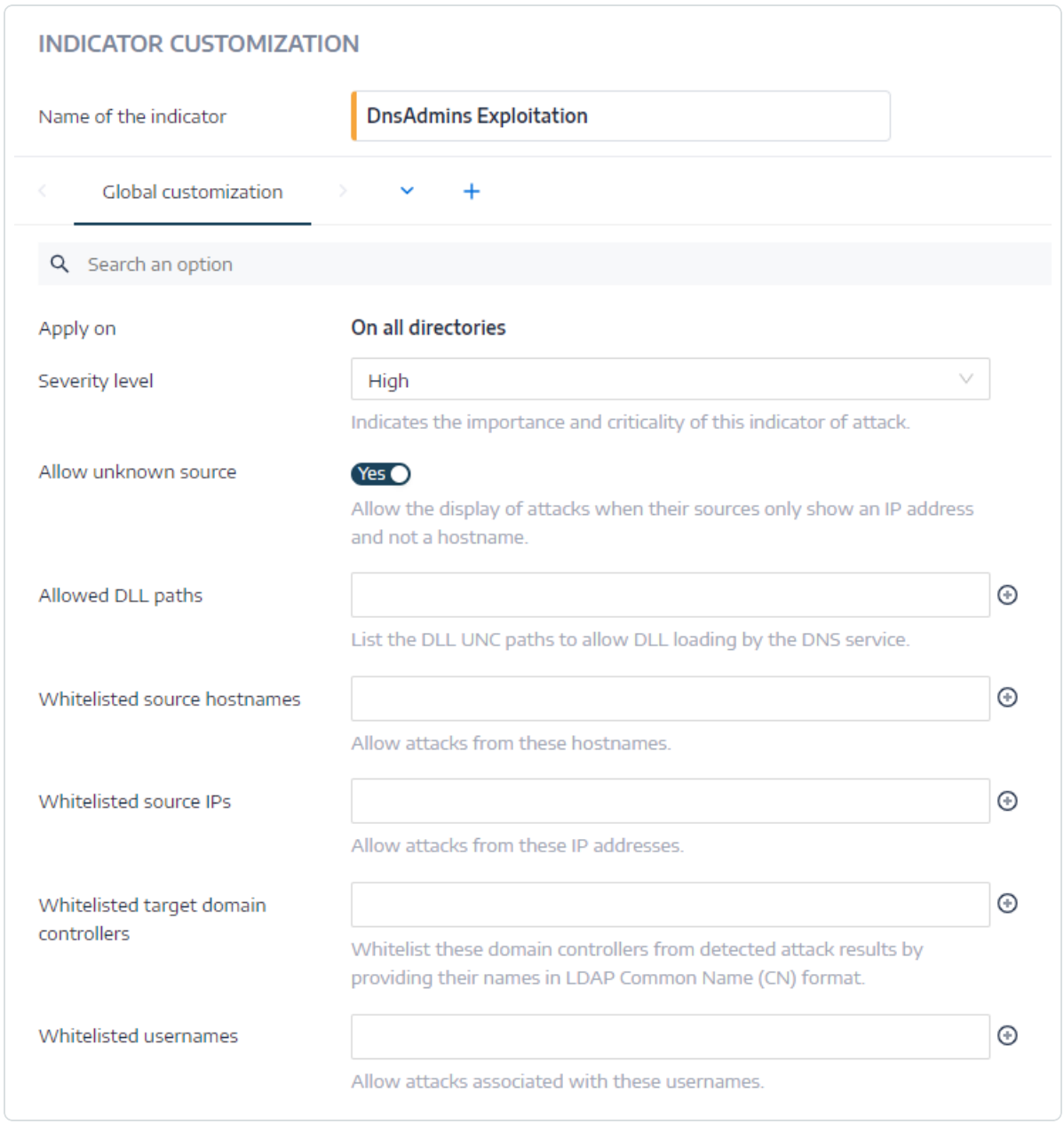
Provider Name	Channel	Event IDs	Audit Policies	Value
Microsoft-Windows-Security-Auditing	Security	4624	├ Category: Logon/Logoff └ – Sub-category: Audit Logon	Success
Microsoft-Windows-DNSServer	Microsoft-Windows-DNSServer/Audit	541		
Other requirements				
Sysmon extension	No			
Honey Account	No			

Options

To select options for this Indicator of Attack, see Customize an Indicator of Attack.

Option Name (Type, Default Value)	
Allow unknown source (Boolean, True)	
Description	Allow the display of attacks when their sources only show an IP address and not a hostname.

Recommendation	Do not change this value unless you have a specific need.
Allowed DLL paths (List of strings, Empty)	
Description	List the DLL UNC paths to allow DLL loading by the DNS service.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted source hostnames (List of strings, Empty)	
Description	Allow attacks from these hostnames.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted source IPs (List of strings, Empty)	
Description	Allow attacks from these IP addresses.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted target domain controllers (List of strings, Empty)	
Description	Whitelist domain controllers from detected attack results by providing their names in the LDAP Common Name (CN) format.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted usernames (List of strings, Empty)	
Description	Allow attacks associated with these usernames.
Recommendation	Do not change this value unless you have a specific need.



Yara Detection Rules

YARA rules are malware detection patterns that you can customize to identify targeted attacks and security threats specific to your environment.

DnsAdmins Exploitation Yara Detection Rules

```
rule dnscmd
{
    meta:
```

```
        description = "Detects DnsCmd use to load a DLL on Domain Controllers (Yara format for ELAT)"
        author      = "Tenable.AD"
        date        = "2022-03-18"

    strings:
        $provider = "Microsoft-Windows-DNSServer" nocase
        $eventid  = "541"
        $setting  = "serverlevelplugindll" nocase

    condition:
        ($provider and $eventid and $setting)
}
```

See also

- OS Credential Dumping: LSASS Memory
- Suspicious DC Password Change
- DCShadow
- DCSync
- Domain Backup Key Extraction
- Enumeration of Local Administrators
- Golden Ticket
- Kerberoasting
- Massive Computers Reconnaissance
- NTDS Extraction
- Password Guessing
- Password Spraying
- PetitPotam
- SAM Name Impersonation
- Unauthenticated Kerberoasting
- Zerologon Exploitation

Domain Backup Key Extraction

The Data Protection API (DPAPI) helps to protect data in operating systems running Windows 2000 and later. Operating systems and applications use DPAPI to protect private keys, stored credentials in Windows XP and later, and other information that they want to keep confidential.

DPAPI uses a mechanism involving several keys created for users and computers objects. For example, when a user logs on to a computer the first time, the system creates a key derived from the user's password that encrypts the first copy of the user's DPAPI master key. The system stores these master keys in the directory `C:\Users\<USER>\AppData\Roaming\Microsoft\Protect\<SID>\<GUID>`, where `<SID>` is the user's security identifier and `<GUID>` is the name of the master key.

> **Note**: The system stores DPAPI computers keys in the system profile directory.

The decrypting of the master key requires the user or computer password. In order to support password resets, the system encrypts a secondary copy of the master key with a DPAPI Domain Backup Key and replicates it on all writable domain controllers. Since this DPAPI Domain Backup Key is global to the domain, any compromise by an attacker can give access to any encrypted confidential data using the DPAPI mechanism.

Examples of DPAPI secrets:

- User
    - Windows "Credentials" (like saved RDP credentials)
    - Windows Vaults
    - Saved IE and Chrome logins/cookies
    - Remote Desktop Connection Manager files with passwords
    - Dropbox syncs
- System

- Scheduled tasks credentials
- Azure sync accounts
- Wi-Fi credentials

Detection Type	Related to a Common Vulnerabilities and Exposures (CVE)	Available from Tenable Identity Exposure version
Generic IOC	No	3.17

How the attack works

The attacker must be a privileged user on the domain to perform the correct Local Security Authority (LSA) RPC calls to extract the DPAPI Domain Backup Key. The system stores this domain backup key using an LSA secret object and generates audit events when accessed.

To achieve this extraction, common attack tools like *mimikatz* carry out the following steps:

1. Logs on to the Domain Controller (DC).
2. Accesses the named pipe "lsarpc" to communicate with the LSA functions.
3. Gets a handle on the global LSA secret object named `G$BCKUPKEY_PREFERRED` and reads it. It contains the GUID of the actual used Domain Backup Key.
4. Gets a handle on the global LSA secret object named `G$BCKUPKEY_<GUID>` (e.g `G$BCKUPKEY_935e526e-e44b-4032-9355-265b57c7dea2`) and reads it. It contains the actual Domain Backup Key used.
5. Returns the private key of the Domain Backup Key pair.

Once attackers obtain this private key, they can decrypt any DPAPI secrets on the domain.

How the IoA works

The DPAPI Domain Backup Key Extraction Indicator of Attack can detect a wide variety of attack tools that use LSA RPC calls to access backup keys.

The strategy is to identify specific patterns in Active Directory events that relate to the access of LSA secret objects containing sensitive data in DPAPI keys. Since certain tools access both current

modern and legacy keys, the Indicator of Attack does not raise multiple deviances for the same attack scenario.

By using enrichment events, the IoA can also provide a detailed description of a DPAPI Domain Backup Key extraction attack.

Specific modifications to the environment

None. Tenable Identity Exposure adapts the audit policy to meet the needs of the required Windows event logs.

Events Auditing Policy

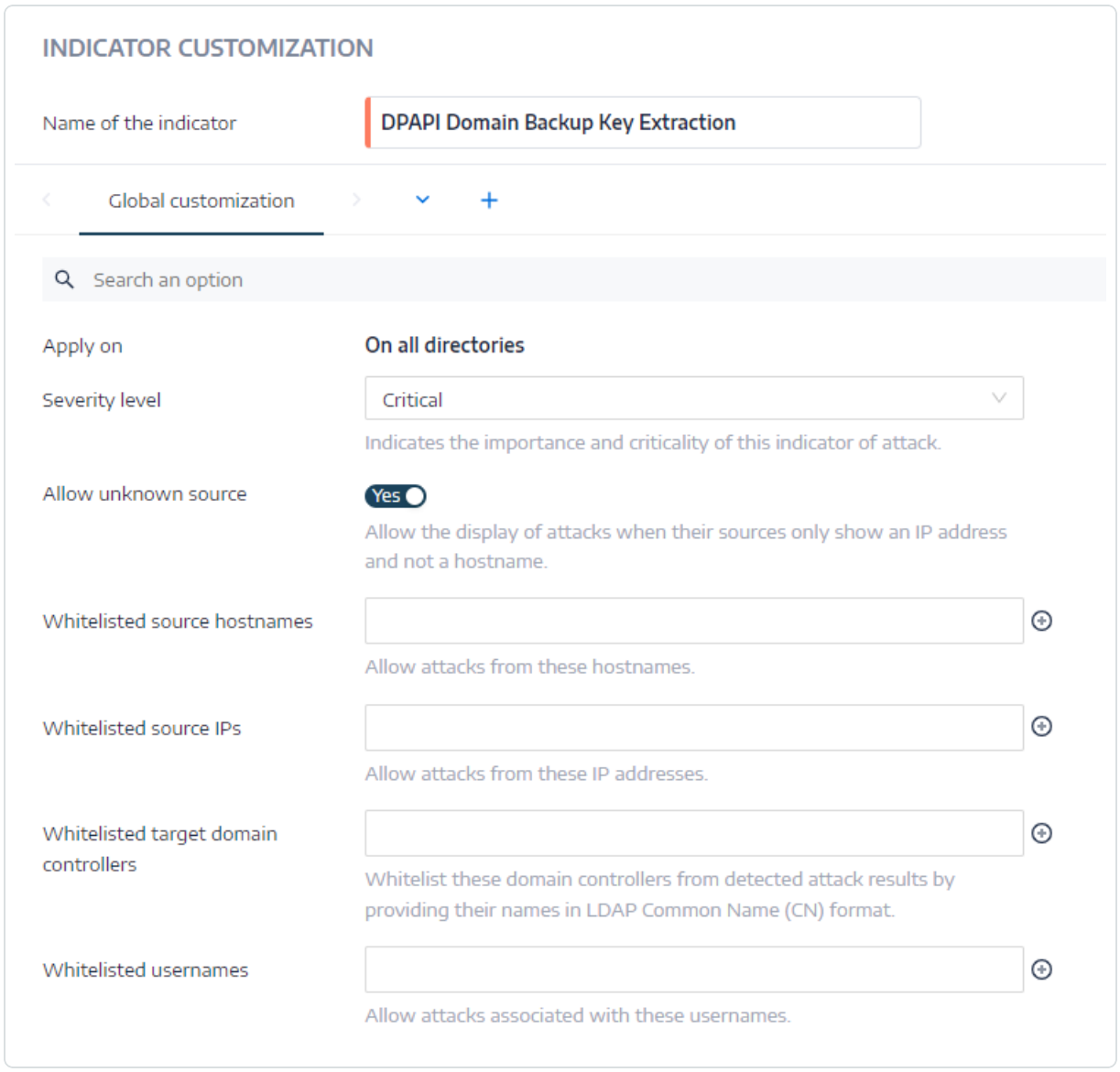
Provider Name	Channel	Event IDs	Audit Policies	Value
Microsoft-Windows-Security-Auditing	Security	4662	├ Category: Object Access └ – Sub-category:Audit Other Object Access Events	Success
Microsoft-Windows-Security-Auditing	Security	4624	├ Category: Logon/Logoff └ – Sub-category: Audit Logon	Success
Other requirements				
Sysmon extension	No			
Honey Account	No			

Options

To select options for this Indicator of Attack, see Customize an Indicator of Attack.

Option Name (Type, Default Value)
Allow unknown source (Boolean, True)

Description	Allow the display of attacks when their sources only show an IP address and not a hostname.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted source hostnames (List of strings, Empty)	
Description	Allow attacks from these hostnames.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted source IPs (List of strings, Empty)	
Description	Allow attacks from these IP addresses.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted target domain controllers (List of strings, Empty)	
Description	Whitelist domain controllers from detected attack results by providing their names in LDAP Common Name (CN) format.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted usernames (List of strings, Empty)	
Description	Allow attacks associated with these usernames.
Recommendation	Do not change this value unless you have a specific need.



Yara Detection Rules

YARA rules are malware detection patterns that you can customize to identify targeted attacks and security threats specific to your environment.

Domain Backup Key Yara Detection Rules

```
rule mimikatz_dpapi
{
    meta:
        description = "Detects Mimikatz use to extract DPAPI master keys (Yara format for ELAT)"
        author      = "Tenable.AD"
        reference   = "https://github.com/gentilkiwi/mimikatz/"
```

```
        date        = "2022-03-08"

    strings:
        $provider      = "Microsoft-Windows-Security-Auditing"
        $eventid       = "4662"
        $objectname    = "BCKUPKEY" nocase
        $operationtype = "Query" nocase
        $accessmask    = "0x2"

    condition:
        ($provider and $eventid and $objectname and $operationtype and $accessmask)
}
```

See also

- OS Credential Dumping: LSASS Memory
- Suspicious DC Password Change
- DCShadow
- DCSync
- Domain Backup Key Extraction
- Enumeration of Local Administrators
- Golden Ticket
- Kerberoasting
- Massive Computers Reconnaissance
- NTDS Extraction
- Password Guessing
- Password Spraying
- PetitPotam
- SAM Name Impersonation
- Unauthenticated Kerberoasting
- Zerologon Exploitation

Enumeration of Local Administrators

This Indicator of Attack (IoA) detects reconnaissance attacks that enumerate the members of the Local Administrator group on domain controllers. A common attack tool that attackers use is BloodHound, which this IoA can detect in BloodHound's default configuration.

Tenable Identity Exposure supports two methods in this IoA:

- Targeted systems for Windows versions 2016 or later.
- Targeted systems for Windows versions 2012 R2 or earlier.

Detection Type	Related to a Common Vulnerabilities and Exposures (CVE)	Available from Tenable Identity Exposure version
Generic IOC	No	3.14

How the attack works

The attacker uses the SAMR RPC (Remote Procedure Call) interface to list the members of the local Administrators group (not a domain group) of some domain controllers.

How the IoA works

This IoA can detect this technique, which SharpHound3 (the crawler part of the BloodHound tool) uses when it is launched through the following configurations:

- Using the default configuration.
- Enabling all collection methods.
- Enabling only the **LocalAdmin** collection method.

In addition to BloodHound, this IoA can detect other attack tools that use the same technique.

You should not have false positives (especially for Windows versions 2016+) because the IoA detection relies on the Sharphound implementation, which differs from the Microsoft library. For this reason, the IoA does not consider as an attack such normal behaviors as the Microsoft Management Console (MMC) and command line tools that remotely list the members of the local Administrators group.

The IoA's detection technique is different for systems running Windows versions earlier than 2012 R2, because Microsoft does not provide the required event for older systems. Tenable Identity

Exposure provides another less robust algorithm and enables it by default for older systems. If required, you can disable this option in Tenable Identity Exposure.

Note: In most situations, this IoA triggers at the same time as the Massive Computers Reconnaissance IoA. This is expected because they do not cover exactly the same cases.

Specific modifications to the environment

None. Tenable Identity Exposure adapts the audit policy to meet the needs of the required Windows event logs.

Events auditing policy				
Provider Name	Channel	Event IDs	Audit Policies	Value
Microsoft-Windows-Security-Auditing	Security	4799	├ Category: Account Management └ – Sub-category: Security Group Management	Success
Microsoft-Windows-Security-Auditing	Security	5145	├ Category: Object Access └ – Sub-category: Detailed File Share	Success
Microsoft-Windows-Security-Auditing	Security	4661	├ Category: Object Access └ – Sub-category: SAM + ├ Category: Object Access └ – Sub-category: Handle Manipulation	Success
Other requirements				

Sysmon extension	No
Honey Account	No

Options

To select options for this Indicator of Attack, see Customize an Indicator of Attack.

Option Name (Type, Default Value)	
Filtered access rights in aggressive mode (List of strings, Empty)	
Description	Considers only specified access rights fetched from the event 'A network share object was checked' to classify a potential attack in progress. This list only applies in aggressive mode.
Recommendation	Do not change this value unless you have a specific need.
Aggressive mode (Boolean, False)	
Description	When activated, the IoA engine triggers an alert if the user attempting to retrieve or enumerate the list of local administrator accounts is not privileged.
Recommendation	Do not change this value unless you have a specific need.
Allow unknown source (Boolean, True)	
Description	Allow the display of attacks when their sources only show an IP address and not a hostname.
Recommendation	Do not change this value unless you have a specific need.
Heuristic for Domain Controllers using versions before Windows Server 2016 (Boolean, False)	
Description	Enable detection for domain controllers running Windows 2012 R2 or earlier versions.
Recommendation	Disable by default to avoid too many false positives.
Whitelisted source hostnames (List of strings, Empty)	

Description	Allow attacks from these hostnames.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted source IPs (List of strings, Empty)	
Description	Allow attacks from these IP addresses.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted target domain controllers (List of strings, Empty)	
Description	Whitelist domain controllers from detected attack results by providing their names in the LDAP Common Name (CN) format. The IoA does not trigger an alert if the target IP is unknown. The behavior applies to both **Basic** and **Aggressive** modes. This is because the attack logic requires confirming that the destination IP is different from the source, which is not possible without the Target IP.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted usernames (List of strings, Empty)	
Description	Allow attacks associated with these usernames.
Recommendation	Do not change this value unless you have a specific need.

INDICATOR CUSTOMIZATION

Name of the indicator: **Enumeration of Local Administrators**

Global customization

Search an option

Apply on: **On all directories**

Severity level: Low

Indicates the importance and criticality of this indicator of attack.

Allow unknown source: Yes

Allow the display of attacks when their sources only show an IP address and not a hostname.

Filtered access rights in agressive mode

List of access rights fetched from the event 'A network share object was checked' that allows an attack to be triggered. This list is only used in aggressive mode.

Aggressive mode: No

When activated, the IoA engine will detect more events as attacks (broader attack detection, but with a risk of false positives).

Heuristic for Domain Controllers using versions prior to Windows Server 2016: No

Enable detection for domain controllers running Windows Server 2012 R2 or earlier versions. The primary and best detection method for this attack requires that domain controllers run Windows Server 2016 or later, but this option offers an alternative although less precise heuristic for detecting the attack on those running older versions.

Whitelisted source hostnames

Allow attacks from these hostnames.

Whitelisted source IPs

Allow attacks from these IP addresses.

Whitelisted target domain controllers

Whitelist these domain controllers from detected attack results by providing their names in LDAP Common Name (CN) format.

Whitelisted usernames

Allow attacks associated with these usernames.

Yara Detection Rules

YARA rules are malware detection patterns that you can customize to identify targeted attacks and security threats specific to your environment.

Enumeration of Local Administrators Yara Detection Rules

```
rule reconadminsenum
{
    meta:
        description = "Detects an attacker which is using the SAMR RPC interface to list the members
of the local Administrators group (Yara format for ELAT)"
        author      = "Tenable.AD"
        reference   = "https://bloodhound.readthedocs.io/en/latest/data-collection/sharphound.html"
        date        = "2021-09-08"

    strings:
        $provider    = "Microsoft-Windows-Security-Auditing"
        $eventid     = "4799"
        $targetsid   = "S-1-5-32-544"

    condition:
        ($provider and $eventid and $target)
}
```

See also

- OS Credential Dumping: LSASS Memory
- Suspicious DC Password Change
- DCShadow
- DCSync
- Enumeration of Local Administrators
- Domain Backup Key Extraction
- Golden Ticket
- Kerberoasting
- Massive Computers Reconnaissance
- NTDS Extraction
- Password Guessing

- Password Spraying
- PetitPotam
- SAM Name Impersonation
- Unauthenticated Kerberoasting
- Zerologon Exploitation

Golden Ticket

A Golden Ticket attack is a type of attack in which an adversary gains control over an Active Directory Key Distribution Service Account (KRBTGT), and uses that account to create valid Kerberos Ticket Granting Tickets (TGTs).

Detection of a Golden Ticket attack occurs only under two conditions:

- In **basic mode** — When the attack targets a privileged identity and the forged TGT is used to impersonate this identity.
- In **aggressive mode** — When the attack targets a privileged identity and the forged TGT is used to impersonate this identity even if the correlation events are not found.

  > **Note**: This indicator raises an alert when the attack is crafted using Rubeus. It does not support other tools and may not generate an alert for them.

  See Privileged Entity Definitions in the Tenable Identity Exposure Indicators of Attack Reference Guide User Guide for more information.

Events Auditing Policy

Event IDs	Audit Policies	Value
4768	├ Category: Account Logon └ — Sub-category: Kerberos Authentication Service	Success and Failure
4769	├ Category: Account Logon └ — Sub-category: Kerberos Service Ticket Operations	Success and Failure
4770	├ Category: Account Logon	Success

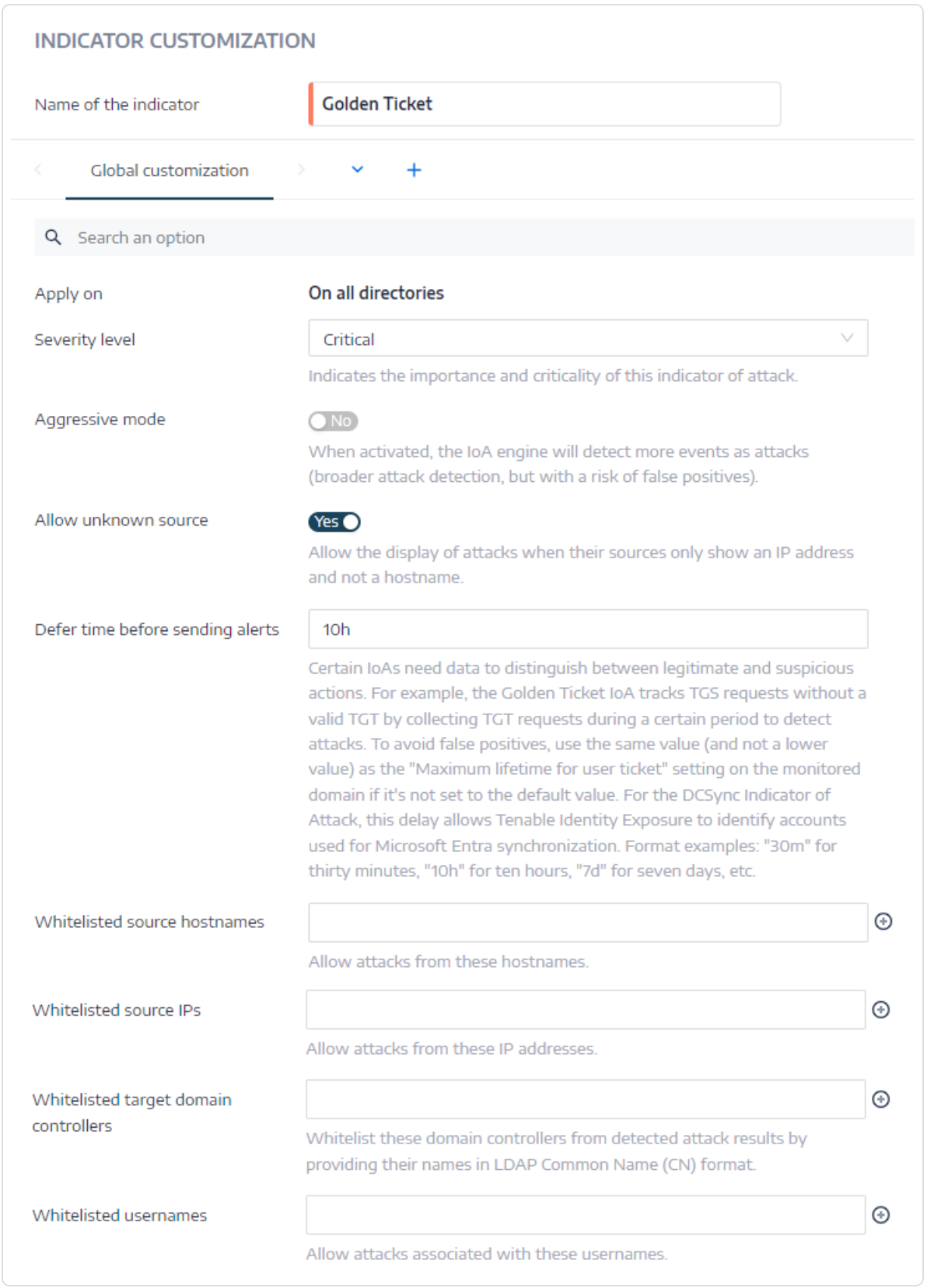
	└ – Sub-category: Kerberos Service Ticket Operations	
4624	├ Category: Logon/Logoff └ – Sub-category: Logon	Success
	Requires Sysmon extension	No
	Requires honey account	No

Options

To select options for this Indicator of Attack, see Customize an Indicator of Attack.

Option Name (Type, Default Value)	
Aggressive mode (Boolean, False)	
Description	Allow attacks even if the correlation events are not found. Also, allow attacks even if some domain controllers are not monitored (in other words, they do not emit any Windows Event Log).
Recommendation	Do not change this value unless you have a specific need.
Allow unknown source (Boolean, True)	
Description	Allow the display of attacks when their sources only show an IP address and not a hostname.
Recommendation	Do not change this value unless you have a specific need.
Defer time before sending alerts (String, 10h)	
Description	Certain IoAs need data to distinguish between legitimate and suspicious actions. For example, the Golden Ticket IoA tracks TGS requests without a valid TGT by collecting TGT requests during a certain period to detect attacks. To avoid false positives, use the same value (and not a lower value) as the "Maximum lifetime for user ticket" setting on the monitored domain if it's not set to the default value. Format examples: "30m" for thirty minutes, "10h" for ten hours, "7d" for seven days, etc.

Recommendation	Do not modify this value unless you've changed the default value for the "Maximum lifetime for user ticket" policy, which is unusual. If you have, update it with the correct value.
Whitelisted source hostnames (List of strings, Empty)	
Description	Allow attacks from these hostnames.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted source IPs (List of strings, Empty)	
Description	Allow attacks from these IP addresses.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted target domain controllers (List of strings, Empty)	
Description	Whitelist domain controllers from detected attack results by providing their names in LDAP Common Name (CN) format.
Recommendation	Do not change this value unless you have a specific need. Certain backup tools can run periodically to back up the NTDS.DIT database and cause this IoA to trigger an attack. Prioritize the use of other options in this IoA to authorize specific tools on certain time slots as they are more precise and help to maintain a high level of detection to identify real attacks.
Whitelisted usernames (List of strings, Empty)	
Description	Allow attacks associated with these usernames.
Recommendation	Do not change this value unless you have a specific need.



Yara Detection Rules

YARA rules are malware detection patterns that you can customize to identify targeted attacks and security threats specific to your environment.

Golden Ticket Yara Detection Rules

```
rule mimikatz
{
    meta:
        description    = "mimikatz"
        author         = "Benjamin DELPY (gentilkiwi)"
        tool_author    = "Benjamin DELPY (gentilkiwi)"

    strings:
        $exe_x86_1     = { 89 71 04 89 [0-3] 30 8d 04 bd }
        $exe_x86_2     = { 8b 4d e? 8b 45 f4 89 75 e? 89 01 85 ff 74 }

        $exe_x64_1     = { 33 ff 4? 89 37 4? 8b f3 45 85 c? 74}
        $exe_x64_2     = { 4c 8b df 49 [0-3] c1 e3 04 48 [0-3] 8b cb 4c 03 [0-3] d8 }

        $dll_1         = { c7 0? 00 00 01 00 [4-14] c7 0? 01 00 00 00 }
        $dll_2         = { c7 0? 10 02 00 00 ?? 89 4? }

        $sys_x86       = { a0 00 00 00 24 02 00 00 40 00 00 00 [0-4] b8 00 00 00 6c 02 00 00 40 00 00
00 }
        $sys_x64       = { 88 01 00 00 3c 04 00 00 40 00 00 00 [0-4] e8 02 00 00 f8 02 00 00 40 00 00
00 }

    condition:
        (all of ($exe_x86_*)) or (all of ($exe_x64_*)) or (all of ($dll_*)) or (any of ($sys_*))
}

rule invoke_mimikatz
{
    meta:
        description = "Detects Invoke-Mimikatz"
        author      = "Tenable.AD"
        reference1  = "https://github.com/EmpireProject/Empire/blob/master/data/module_
source/credentials/Invoke-Mimikatz.ps1"
        reference2  = "https://github.com/BC-SECURITY/Empire/blob/master/data/module_
source/credentials/Invoke-Mimikatz.ps1"
        date = "2020-11-09"

    strings:
        $sekurlsa   = "sekurlsa" wide base64
        $lsadump    = "lsadump" wide base64

    condition:
        all of them
}

rule secretsdump
{
    meta:
        description     = "Detects secretsdump"
        author          = "Tenable.AD"
        reference       =
"https://github.com/SecureAuthCorp/impacket/blob/master/examples/secretsdump.py"
        date            = "2020-11-09"
```

```
    strings:
        $impacket       = "impacket"
        $secretsdump    = "secretsdump"

    condition:
        all of them
}

rule ticketer
{
    meta:
        description = "Detects ticketer"
        author      = "Tenable.AD"
        reference   = "https://github.com/SecureAuthCorp/impacket/blob/master/examples/ticketer.py"
        date        = "2020-11-13"

    strings:
        $impacket   = "impacket"
        $ticketer   = "ticketer"

    condition:
        all of them
}
```

See also

- OS Credential Dumping: LSASS Memory
- Suspicious DC Password Change
- DCShadow
- DCSync
- Golden Ticket
- Domain Backup Key Extraction
- Enumeration of Local Administrators
- Kerberoasting
- Massive Computers Reconnaissance
- NTDS Extraction
- Password Guessing
- Password Spraying

- PetitPotam
- SAM Name Impersonation
- Unauthenticated Kerberoasting
- Zerologon Exploitation

Kerberoasting

Kerberoasting is a type of attack that targets Active Directory service account credentials for offline password cracking.

This attack seeks to gain access to service accounts by requesting and extracting service tickets and then cracking the service account's credentials offline.

The Kerberoasting Indicator of Attack requires the activation of Tenable Identity Exposure's Honey Account feature to send out an alert when there is a login attempt on the Honey Account or if this account receives a ticket request.

Events Auditing Policy

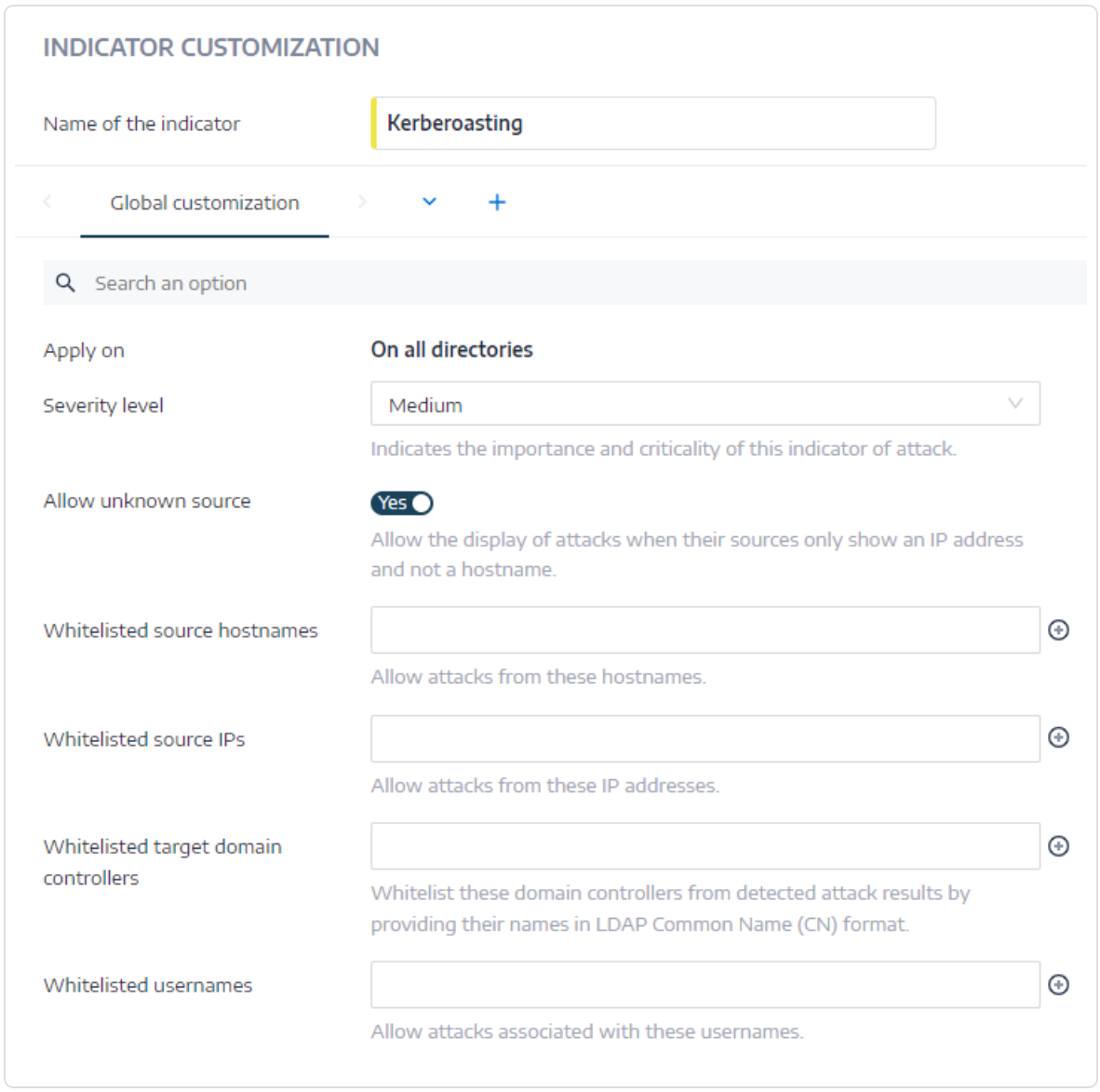
Provider Name	Channel	Event ID	Audit Policies	Value
Microsoft-Windows-Security-Auditing	Security	4769	├ Category: Account Logon └ – Sub-category: Kerberos Service ticket operations	Success
Requires Sysmon extension			No	
Requires honey account			Yes	

Options

To select options for this Indicator of Attack, see Customize an Indicator of Attack.

Option Name (Type, Default Value)

Allow unknown source (Boolean, True)	
Description	Allow the display of attacks when their sources only show an IP address and not a hostname.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted source hostnames (List of strings, Empty)	
Description	Allow attacks from these hostnames.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted source IPs (List of strings, Empty)	
Description	Allow attacks from these IP addresses.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted target domain controllers (List of strings, Empty)	
Description	Whitelist domain controllers from detected attack results by providing their names in the LDAP Common Name (CN) format.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted usernames (List of strings, Empty)	
Description	Allow attacks associated with these usernames.
Recommendation	Do not change this value unless you have a specific need.



Yara Detection Rules

YARA rules are malware detection patterns that you can customize to identify targeted attacks and security threats specific to your environment.

Kerberoasting Yara Detection Rules

```
rule invoke_kerberoast
{
    meta:
        description             = "Detects Invoke-Kerberoast"
        author                  = "Tenable.AD"
        comment                 = "It is used to perform a Kerberoasting attack against the domain
(PowerSploit and Empire tools)."
        reference1              = "https://github.com/EmpireProject/Empire/blob/master/data/module_
```

```
source/credentials/Invoke-Kerberoast.ps1"
        date = "2019-05-30"

    strings:
        $authors                = "Author: Will Schroeder (@harmj0y), @machosec" nocase
        $heuristic              = "function Invoke-Kerberoast {" nocase

    condition:
        all of them
}

rule skelsec_kerberoast
{
    meta:
        description             = "Detects SkelSec Kerberoasting tool and library"
        author                  = "Tenable.AD"
        comment                 = "It is used to perform a Kerberoasting attack against the domain."
        reference1              = "https://github.com/skelsec/kerberoast"
        date = "2021-03-27"

    strings:
        $authors                 = "Tamas Jos (@skelsec)" nocase
        $minikerberos            = "class Kerberoast:" nocase
        $importMiniKerberos      = /from minikerberos.security import .*?Kerberoast/ nocase

    condition:
        2 of them
}

rule impacket_getuserspns
{
    meta:
        description             = "Detects Impacket GetUserSPNs.py module"
        author                  = "Tenable.AD"
        comment                 = "It is used to perform a Kerberoasting attack against the domain."
        reference1              =
"https://github.com/SecureAuthCorp/impacket/blob/master/examples/GetUserSPNs.py"
        date = "2021-07-20"

    strings:
        $authors                 = "SECUREAUTH LABS. Copyright (C)" nocase
        $heuristic               = "class GetUserSPNs:" nocase

    condition:
        all of them
}
```

See also

- OS Credential Dumping: LSASS Memory
- Suspicious DC Password Change
- DCShadow
- DCSync

- Kerberoasting
- Domain Backup Key Extraction
- Enumeration of Local Administrators
- Golden Ticket
- Massive Computers Reconnaissance
- NTDS Extraction
- Password Guessing
- Password Spraying
- PetitPotam
- SAM Name Impersonation
- Unauthenticated Kerberoasting
- Zerologon Exploitation

Massive Computers Reconnaissance

This Indicator of Attack (IoA) detects reconnaissance attacks that generate a massive number of authentication requests to Active Directory (AD) computers. A common attack tool that attackers use is BloodHound, which this IoA can detect in most scenarios.

This IoA supports the following two cases:

- An attacker using a domain-joined computer (for example a compromised machine after a phishing attack).
- An attacker using a computer outside of the domain (for example a rogue computer connected to the network).

Detection Type	Related to a Common Vulnerabilities and Exposures (CVE)	Available from Tenable Identity Exposure version
Behavioral	No	3.14

How the attack works

This IoA focuses on massive authentication requests originating from specific attack tools. In particular, when an attacker uses SharpHound3 (the crawler part of BloodHound), this tool calls some Remote Procedure Call (RPC) functions on all domain machines with a DNS name that resolves and which it can reach *via* SMB on TCP/445. As a result, the attacker account must authenticate to these computers before it can proceed. This leads to a large number of authentication requests in a short period of time, which triggers this IoA.

In addition to BloodHound, this IoA can detect other attack tools that exhibit a similar behavior.

How the IoA works

Tenable Identity Exposure triggers this IoA when it finds a dedicated pattern in a **combination** of the following conditions: (Default behavior that you can modify through the IoA options.)

- **Volumetry**: During a **1-hour window**, if there are authentication requests **for more than 10% of the total number of computers** in the AD (with a **fixed limit of 300** computers).
- **Source**: The requests all come from the same machine IP and domain account.
- **Diversity**: The requests target different domain computers.

> **Note**: Because various domain controllers can answer authentication requests, Tenable Identity Exposure aggregates the events from all domain controllers and does the calculation on the sum.

Tenable Identity Exposure filters out the same attack during a 15-minute period to limit the number of security alerts. Examples:

- If an attacker launches the same attack multiple times during those 15 minutes, Tenable Identity Exposure only raises one alert with this IoA.
- If an attack takes one hour to complete, Tenable Identity Exposure triggers four alerts to remind you that the attack is still in progress.

> **Note**: Tenable Identity Exposure offers several configuration options for this IoA. You may need to adapt them depending on the size of each monitored domain (the number of domain-joined computers) to have the fastest possible detection without getting false positives.

> **Note**: In some situations, this IoA triggers at the same time as the Enumeration of Local Administrators IoA. This is expected because they do not cover exactly the same cases.

Specific modifications to the environment

To analyze NTLM authentication requests, the IoA script **automatically** configures the policy settings on your domain controllers through the Tenable Identity Exposure Group Policy Object (GPO), as follows:

Location of the setting	Security policy setting	Value
Computer Configuration > Policies > Windows Settings > Security Settings > Local Policies > Security Options	Network security: Restrict NTLM: Outgoing NTLM traffic to remote servers	Audit all
	Network security: Restrict NTLM: Audit NTLM authentication in this domain	Enable all
	Network security: Restrict NTLM: Audit Incoming NTLM Traffic	Enable auditing for all accounts

Events Auditing Policy

Provider Name	Channel	Event IDs	Audit Policies	Value
Microsoft-Windows-Security-Netlogon	Microsoft-Windows-NTLM/Operational	8004	Configuration through a dedicated log, enabled by security policy settings.	N/A
Microsoft-Windows-Security-Auditing	Security	4624	├ Category: Logon/Logoff └ – Sub-category: Logon	Success
Microsoft-Windows-Security-Auditing	Security	4769	├ Category: Account Logon └ – Sub-category: Kerberos Service Ticket Operations	Success

Sysmon extension	No
Honey Account	No

Options

To select options for this Indicator of Attack, see Customize an Indicator of Attack.

Due to its dependence on volumetric calculations and the distinctiveness of each environment, modify these options in the IoA to adapt it effectively to your environment: **Number of computers**, **Percentage of computers**, **Sliding window**, and **Waiting time between attacks**.

Option Name (Type, Default Value)	
Allow unknown source (Boolean, True)	
Description	Allow the display of attacks when their sources only show an IP address and not a hostname.
Recommendation	Do not change this value unless you have a specific need.
Number of computers (Integer, 5000)	
Description	Indicate the number of computers that must be in a specific state to classify as an attack. Lower numbers allow faster detection but may lead to false positives; higher numbers may result in false negatives if attackers target only a small number of computers.
Recommendation	• Small environments: 100 • Large environments: 50 000
Minimum number of computers (internal option) (Integer, 100)	
Description	Define the minimum number of computers required to identify an attack (volumetry heuristic). This number sets a hard limit if the percentage of affected computers in a domain is lower than this limit. This option reduces false negatives in demo AD environments with

	fewer than 10 computers. Do not modify it unless you are aware of the impact of lower values (<10) resulting in many false positives in production environments.
Recommendation	Do not change this value unless you have a specific need. Option used primarily for QA testing and small environments such as demos or POV. This option sets a hard limit below which no attack triggers, regardless of the computed threshold. If the computed threshold is less than the option value, the algorithm uses the option value as the threshold. In an AD environment with 20 active computers, the default threshold for triggering an attack is 2 (10% of 20). This means that if the same user with the same machine triggers two NTLM/Kerberos events for two different machines, it could trigger a false positive attack alert. To prevent false positives, the IOA only triggers an attack if it receives 10 (default) or more events from the same source that target different machines. For larger environments, a low value (such as 1 for this option) could result in too many false positives. In production environments, it is not necessary to change the value for this option. However, for small QA/POV AD environments, set the value lower than the number of actual computers in the domain to enable triggering the attack.
Percentage of computers (Integer, 95)	
Description	Indicate the percentage of domain computers that must be in a specific state to classify as an attack.
Recommendation	Do not change this value. Use the **Number Of Computers** option instead.
Sliding window (Integer, 240)	
Description	Specify the duration in minutes to do an analysis, as an attack on

	multiple systems requires time to complete. During this period, Tenable Identity Exposure computes the sum of authentication attempts. If the duration is set too low, a slow attack may go undetected. However, if the duration is set too high, performance suffers and the analysis may become inaccurate.
Recommendation	• Small environments: 10m • Large environments: 120m
Waiting time between attacks (Integer, 240)	
Description	Set a time period in minutes to limit the number of alerts generated by a prolonged attack. If an attacker launches the same attack multiple times from the same computer within this time period, it triggers a single alert.
Recommendation	• Small environments: 5m • Large environments: 20m
Whitelisted source hostnames (List of strings, Empty)	
Description	Allow attacks from these hostnames.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted source IPs (List of strings, Empty)	
Description	Allow attacks from these IP addresses.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted target domain controllers (List of strings, Empty)	
Description	Whitelist domain controllers from detected attack results by providing their names in the LDAP Common Name (CN) format.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted usernames (List of strings, Empty)	
Description	Allow attacks associated with these usernames.

Recommendation	Do not change this value unless you have a specific need.

INDICATOR CUSTOMIZATION

Name of the indicator: **Massive Computers Reconnaissance**

Global customization

Search an option

Apply on: **On all directories**

Severity level: Low

Indicates the importance and criticality of this indicator of attack.

Allow unknown source: Yes

Allow the display of attacks when their sources only show an IP address and not a hostname.

Number of computers: 5000

Indicate the number of computers that must be in a specific state to classify as an attack (volumetry heuristic). Lower numbers allow faster detection but may lead to false positives; higher numbers may result in false negatives if attackers target only a small number of computers. Recommended values: 100 for small environments ; 600 for larger ones.

Minimum number of computers (internal option): 100

Define the minimum number of computers required to identify an attack (volumetry heuristic). This number sets a hard limit if the percentage of affected computers in a domain is lower than this limit. This option reduces false negatives in demo AD environments with fewer than 10 computers. Do not modify it unless you are aware of the impact of lower values (<10) resulting in many false positives in production environments.

Percentage of computers: 95

Indicate the percentage of domain computers that must be in a specific state to classify as an attack (volumetry heuristic). Recommendation: Do not change this value; use the "Number of computers" option instead.

Sliding window: 240

Specify the study duration to take for the analysis in minutes. An attack on multiple systems takes time to complete. During this period, Tenable Identity Exposure calculates the sum of authentication attempts. If the duration is too low, a slow attack may go unnoticed. However, if the duration is too long, performance suffers and the analysis can become inaccurate. Recommended values: 10min for small environments; 120min for large environments.

Waiting time between attacks: 240

Set a time period in minutes to limit the number of alerts generated by a prolonged attack. If an attacker launches the same attack multiple times from the same computer within this time period, it triggers a single alert. Recommended values: 5min for small environments; 20min for larger ones.

Whitelisted source hostnames: - 63 -

Allow attacks from these hostnames.

Yara Detection Rules

YARA rules are malware detection patterns that you can customize to identify targeted attacks and security threats specific to your environment.

Massive Reconnaissance Yara Detection Rules

```
{
    meta:
        description = "Detects an attacker which is using SharpHound tool on RPC collection methods
(Yara format for ELAT)"
        author      = "Tenable.AD"
        reference   = "https://bloodhound.readthedocs.io/en/latest/data-collection/sharphound.html"
        date        = "2021-12-15"
    strings:
        $providerNtlm     = "Microsoft-Windows-Security-Netlogon"
        $eventidNtlm      = "8004"
        $providerKerberos = "Microsoft-Windows-Security-Auditing"
        $eventidKerberos  = "4769"
    condition:
        #($providerNtlm and $eventidNtlm) > 300 or #($providerKerberos and $eventidKerberos) > 300
}
```

See also

- OS Credential Dumping: LSASS Memory
- Suspicious DC Password Change
- DCShadow
- DCSync
- Massive Computers Reconnaissance
- Domain Backup Key Extraction
- Enumeration of Local Administrators
- Golden Ticket
- Kerberoasting
- NTDS Extraction
- Password Guessing
- Password Spraying

- PetitPotam
- SAM Name Impersonation
- Unauthenticated Kerberoasting
- Zerologon Exploitation

NTDS Extraction

NTDS exfiltration refers to the technique that attackers use to retrieve the NTDS.dit database that stores Active Directory secrets such as password hashes and Kerberos keys. Once retrieved, the attacker parses a copy of this file offline, providing an alternative to DCSync attacks for retrieval of the Active Directory's sensitive content.

This Indicator of Attack sends an alert when an event shows the creation of a shadow copy of the database file in an attempt to exfiltrate the `NTDS.dit` database.

Detection Type	Related to a Common Vulnerabilities and Exposures (CVE)	Available from Tenable Identity Exposure version
Generic IOC	No	3.15

How the attack works

Since the operating system constantly accesses the `NTDS.dit` file, an attacker cannot read this file while it's being modified. In order to retrieve the password hashes from the `NTDS.dit` file, an attacker must meet one of the following criteria;

- No shadow copy exists, so the attacker must create a new one to represent a backup or a snapshot of the "C:" volume to get access to the targeted `NTDS.dit` file.
- A shadow copy already exists, so the attacker has direct access to it.

Once the attacker creates a shadow copy, they only have to exfiltrate the `NTDS.dit` file from the shadow volume (for example `\\?\GLOBALROOT\Device\HarddiskVolumeShadowCopy1\windows\system32\ntds.dit`) to a location which they control.

A broad range of tools can carry out this type of attack, including legitimate administration Windows tools such as `vssadmin` or `esentutl`.

How the IoA works

The NTDS Extraction Indicator of Attack can detect a large variety of attack tools by correlating Windows events specific to each step of this attack. Two main events drive the detection algorithm: one specific to the creation of the shadow copy, and the other specific to the creation of a process on the domain controller. This second step allows the detection of malicious exfiltration activity independently of the creation of a shadow copy.

As a consequence, the IoA can detect at an early stage any suspicious patterns linked to an exfiltration attack. Also, using others relevant Windows events, the IoA can provide a detailed description of an NTDS exfiltration attack.

Specific modifications in the environment

To have access to the full command line in the event Microsoft-Windows-Security-Auditing/4688, the IoA script automatically configures the policy settings on your domain controllers through the Tenable Identity Exposure Group Policy Object (GPO), as follows:

Location of the setting	Security policy setting	Value
Computer Configuration > Administrative Templates > System > Audit Process Creation	Include command line in process creation events	Enabled

Events Auditing Policy

Provider Name	Channel	Event ID	Audit Policies	Value
VSSAudit	Security	8222	├ Category: Object Access └ – Sub-category: Audit Application Generated	Success
Microsoft-Windows-Security-Auditing	Security	4688	├ Category: Detailed Tracking └ – Sub-category: Audit Process	Success

			Creation	
Microsoft-Windows-Security-Auditing	Security	5145	├ Category: Object Access └ – Sub-category: Audit Detailed File Share	Success
ESENT	Application	325	N/A	N/A
Microsoft-Windows-WMI-Activity	Microsoft-Windows-WMI-Activity/Operational	5857	N/A	N/A
Microsoft-Windows-DNSServer		271	N/A	N/A
Microsoft-Windows-Security-Auditing	Security	4624	├ Category: Logon/Logoff └ – Sub-category: Audit Logon	Success
Microsoft-Windows-Security-Auditing	Security	4674	├ Category: Privilege Use └ – Sub-category: Audit Sensitive Privilege Use	Success / Failure
Microsoft-Windows-Security-Auditing	Security	4689	├ Category: Detailed Tracking └ – Sub-category: Audit Process Termination	Success

Other requirements	
Sysmon extension	No
Honey Account	No

Options

To select options for this Indicator of Attack, see Customize an Indicator of Attack.

Option Name (Type, Default Value)	
Aggressive mode (Boolean, False)	
Description	Detects more attacks but carries a risk of false positives.
Recommendation	Do not change this value unless you have a specific need.
Allowed processes in aggressive mode (List of strings, list of processes)	
Description	List of allowed processes used for malicious actions. This allowlist supports multiple inputs, such as: full process path, process name only, or the specific technique that the IoA identified.
Recommendation	Use the exact string indicated by an attack, but confirm beforehand that it's a false positive and the alert is linked to a genuine application in your environment. This list applies only with "Aggressive" mode enabled.
Basic mode -- Deny list (List of strings, list of processes)	
Description	Specifies the tools that trigger attacks in basic mode: `diskshadow`, `ntdsutil`, `esentutl`, `esentutldefrag mode`, `vssown, copy-vss`, wmi-based technique, `psexec_ntds_grab`, `wmiprvse`, `vssadmin`, `vss`, `impacket-secretsdump`, `vss_requestor`, `VeeamGuestHelper`, WMI-based technique.
Recommendation	Adapt the list to your environment, especially if you're using administrative tools that attackers can abuse.

Allow unknown source (Boolean, True)	
Description	Allow the display of attacks when their sources only show an IP address and not a hostname.
Recommendation	Do not change this value unless you have a specific need.
Allowed NTDS destination paths (List of strings, Empty)	
Description	List the authorized paths for NTDS.DIT destination.
Recommendation	If some backup tools run periodically to back up the NTDS.DIT database, add the destination to this option.
Whitelisted source hostnames (List of strings, Empty)	
Description	Allow attacks from these hostnames.
Recommendation	Do not change this value unless you have a specific need. Certain backup tools can run periodically to back up the NTDS.DIT database and cause this IoA to trigger an attack. Prioritize the use of other options in this IoA to authorize specific tools on certain time slots as they are more precise and help to maintain a high level of detection to identify real attacks.
Whitelisted source IPs (List of strings, Empty)	
Description	Allow attacks from these IP addresses.
Recommendation	Do not change this value unless you have a specific need. Certain backup tools can run periodically to back up the NTDS.DIT database and cause this IoA to trigger an attack. Prioritize the use of other options in this IoA to authorize specific tools on certain time slots as they are more precise and help to maintain a high level of detection to identify real attacks.
Whitelisted target domain controllers (List of strings, Empty)	
Description	Whitelist domain controllers from detected attack results by providing their names in LDAP Common Name (CN) format.

Recommendation	Do not change this value unless you have a specific need. Certain backup tools can run periodically to back up the NTDS.DIT database and cause this IoA to trigger an attack. Prioritize the use of other options in this IoA to authorize specific tools on certain time slots as they are more precise and help to maintain a high level of detection to identify real attacks.
Time slot durations (Timespan, 0)	
Description	Common duration of each of above time slots.
Recommendation	If some backup tools run periodically to back up the NTDS.DIT database, add the common duration for the backup task to this option.
Allowed time slot start times (List of strings, Empty)	
Description	Exclude start times of time slots from the IoA analysis. Combine them with **Time slots duration**. Consists of a CRON list in GMT. Each element of this option is a string matching the crontab format.
Recommendation	If some backup tools run periodically to back up the NTDS.DIT database, add the time interval to this option.
Whitelisted usernames (List of strings, Empty)	
Description	Allow attacks associated with these usernames.
Recommendation	Do not change this value unless you have a specific need. Certain backup tools can run periodically to back up the NTDS.DIT database and cause this IoA to trigger an attack. Prioritize the use of other options in this IoA to authorize specific tools on certain time slots as they are more precise and help to maintain a high level of detection to identify real attacks.

INDICATOR CUSTOMIZATION

Name of the indicator: **NTDS Extraction**

Global customization

Search an option

Apply on: **On all directories**

Severity level: Critical

Indicates the importance and criticality of this indicator of attack.

Aggressive mode: No

When activated, the IoA engine will detect more events as attacks (broader attack detection, but with a risk of false positives).

Allowed processes in agressive mode

List of allowed processes potentially used for malicious actions. This allow list supports multiple inputs: the full process path, just the process name, or the specific technique that the indicator identified. The safest option is to use the exact string indicated by an attack, but only after confirming it's a false positive and the alert is linked to a genuine application in your environment. This list is only used in aggressive mode.

Basic mode - Deny list

- diskshadow
- ntdsutil
- esentutl
- esentutldefrag mode
- vssown
- copy-vss
- wmi based technique
- psexec_ntds_grab
- wmiprvse
- vssadmin
- vss
- impacket-secretsdump

On basic mode, only these tools will trigger attacks.

Allow unknown source: Yes

Allow the display of attacks when their sources only show an IP address and not a hostname.

Allowed NTDS destination paths

List the destination paths authorized for copying the NTDS.dit file from a shadow copy.

Yara Detection Rules

YARA rules are malware detection patterns that you can customize to identify targeted attacks and security threats specific to your environment.

NTDS Extraction Yara Detection Rules

```
rule invoke_ninjacopy
{
    meta:
        description             = "Detects Invoke-NinjaCopy"
        author                  = "Tenable.AD"
        comment                 = "Copies a file from an NTFS partitioned volume by reading the raw
volume and parsing the NTFS structures."
        reference1              = "https://github.com/BC-
SECURITY/Empire/blob/master/empire/server/data/module_source/collection/Invoke-NinjaCopy.ps1"
        date                    = "2021-02-17"

    strings:
        $authors                = "Author: Joe Bialek, Twitter: @JosephBialek" nocase
        $heuristic              = "function Invoke-NinjaCopy" nocase

    condition:
        all of them
}

rule copy_vss
{
    meta:
        description             = "Detects Copy-VSS"
        author                  = "Tenable.AD"
        comment                 = "This payload uses the VSS service (starts it if not running),
creates a shadow of C: and copies the SAM file which could be used to dump password hashes from it."
        reference1              = "https://github.com/samratashok/nishang/blob/master/Gather/Copy-
VSS.ps1"
        date                    = "2017-12-18"

    strings:
        $heuristic1             = "function Copy-VSS" nocase
        $heuristic2             = "(Get-WmiObject -list win32_shadowcopy)" nocase

    condition:
        all of them
}

rule impacket_secretsdump
{
    meta:
        description             = "Detects Impacket secretsdump.py module"
        author                  = "Tenable.AD"
        comment                 = "Extract NTDS.dit via vssadmin executed  with the smbexec approach.
It's copied on the temp dir and parsed remotely."
        reference1              =
"https://github.com/SecureAuthCorp/impacket/blob/master/examples/secretsdump.py"
        date                    = "2021-07-07"

    strings:
```

```
        $authors                = "SECUREAUTH LABS. Copyright (C)" nocase
        $heuristic              = "class DumpSecrets:" nocase

    condition:
        all of them
}

rule impacket_compiled_secretsdump {
   meta:
      description               = "Detects compiled Impacket secretsdump module"
      license                   = "Detection Rule License 1.1 https://github.com/Neo23x0/signature-
base/blob/master/LICENSE"
      author                    = "Florian Roth"
      reference                 = "https://github.com/maaaaz/impacket-examples-windows"
      date                      = "2017-04-07"
      hash1                     = "47afa5fd954190df825924c55112e65fd8ed0f7e1d6fd403ede5209623534d7d"
   strings:
      $s1                       = "ssecretsdump" fullword ascii
      $s2                       = "impacket.ese(" fullword ascii
   condition:
      ( uint16(0) == 0x5a4d and filesize < 17000KB and all of them )
}
```

See also

- OS Credential Dumping: LSASS Memory
- Suspicious DC Password Change
- DCShadow
- DCSync
- NTDS Extraction
- Domain Backup Key Extraction
- Enumeration of Local Administrators
- Golden Ticket
- Kerberoasting
- Massive Computers Reconnaissance
- Password Guessing
- Password Spraying
- PetitPotam

- SAM Name Impersonation
- Unauthenticated Kerberoasting
- Zerologon Exploitation

Password Guessing

A brute-force password guessing attack consists of an attacker submitting many passwords or pass phrases and hoping to guess correctly eventually. The attacker systematically checks all possible passwords and pass phrases until it finds the correct one.

Events Auditing Policy

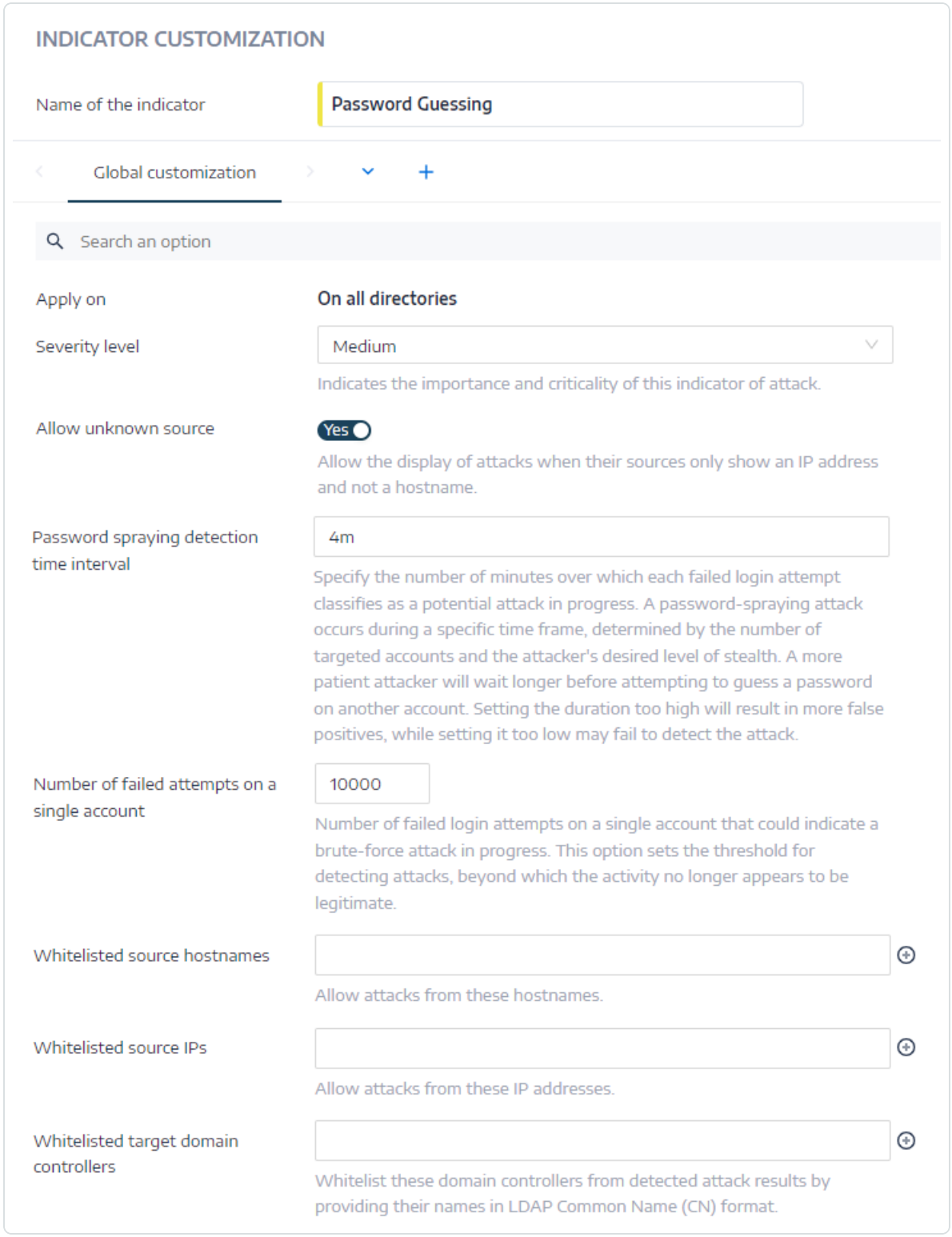
Event IDs	Audit Policies	Value
4625	├ Category: Logon/Logoff └ – Sub-category: Logoff	Failure
4771	├ Category: Account Logon └ – Sub-category: Kerberos Authentication Serviced	Failure
4624	├ Category: Logon/Logoff └ – Sub-category: Audit Logon	Success
4769	├ Category: Account Logon └ – Sub-category: Kerberos Service Ticket Operations	Success
4776	├ Category: Account Logon └ – Sub-category: Credential Validation	Success and Failure
	Requires Sysmon extension	No
	Honey Account	No

Options

To select options for this Indicator of Attack, see Customize an Indicator of Attack.

Option Name (Type, Default Value)	
Allow unknown source (Boolean, True)	
Description	Allow the display of attacks when their sources only show an IP address and not a hostname.
Recommendation	Do not change this value unless you have a specific need.
Password spraying detection time interval (String, 4m)	
Description	Specify the number of minutes over which each failed login attempt classifies as a potential attack in progress.
Recommendation	Do not change this value unless you have a specific need.
Number of failed attempts on a single account (Integer, 10 000)	
Description	The number of failed login attempts on a single account that could indicate an attack in progress. This option sets the threshold for detecting attacks, beyond which the activity no longer appears to be legitimate.
Recommendation	Do not change this value unless you have a specific need. Even for large environments, 500 is a good threshold to trigger a Password Guessing alert.
Whitelisted source hostnames (List of strings, Empty)	
Description	Allow attacks from these hostnames.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted source IPs (List of strings, Empty)	
Description	Allow attacks from these IP addresses.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted target domain controllers (List of strings, Empty)	
Description	Whitelist domain controllers from detected attack results by providing

	their names in the LDAP Common Name (CN) format.
Recommendation	Do not change this value unless you have a specific need.



Yara Detection Rules

YARA rules are malware detection patterns that you can customize to identify targeted attacks and security threats specific to your environment.

Password Guessing Yara Detection Rules

```
rule invoke_domain_passwordspray
{
    meta:
        description             = "Detects Invoke-DomainPasswordSpray"
        author                  = "Tenable.AD"
        comment                 = "It can be used to perform a spraying or a bruteforce."
        reference1              =
"https://github.com/dafthack/DomainPasswordSpray/blob/master/DomainPasswordSpray.ps1"
        date = "2020-11-13"

    strings:
        $fctname                = "Invoke-DomainPasswordSpray" nocase
        $domainpasswordspray    = "DomainPasswordSpray" nocase
        $heuristic              = "Using $UserList as userlist to spray with" nocase

    condition:
        all of them
}
```

See also

- OS Credential Dumping: LSASS Memory
- Suspicious DC Password Change
- DCShadow
- DCSync
- Password Guessing
- Domain Backup Key Extraction
- Enumeration of Local Administrators
- Golden Ticket
- Kerberoasting
- Massive Computers Reconnaissance

- NTDS Extraction
- Password Spraying
- PetitPotam
- SAM Name Impersonation
- Unauthenticated Kerberoasting
- Zerologon Exploitation

Password Spraying

Password spraying is an attack that attempts to access a large number of accounts with a single or a few commonly used passwords, also known as the low-and-slow method.

Events Auditing Policy

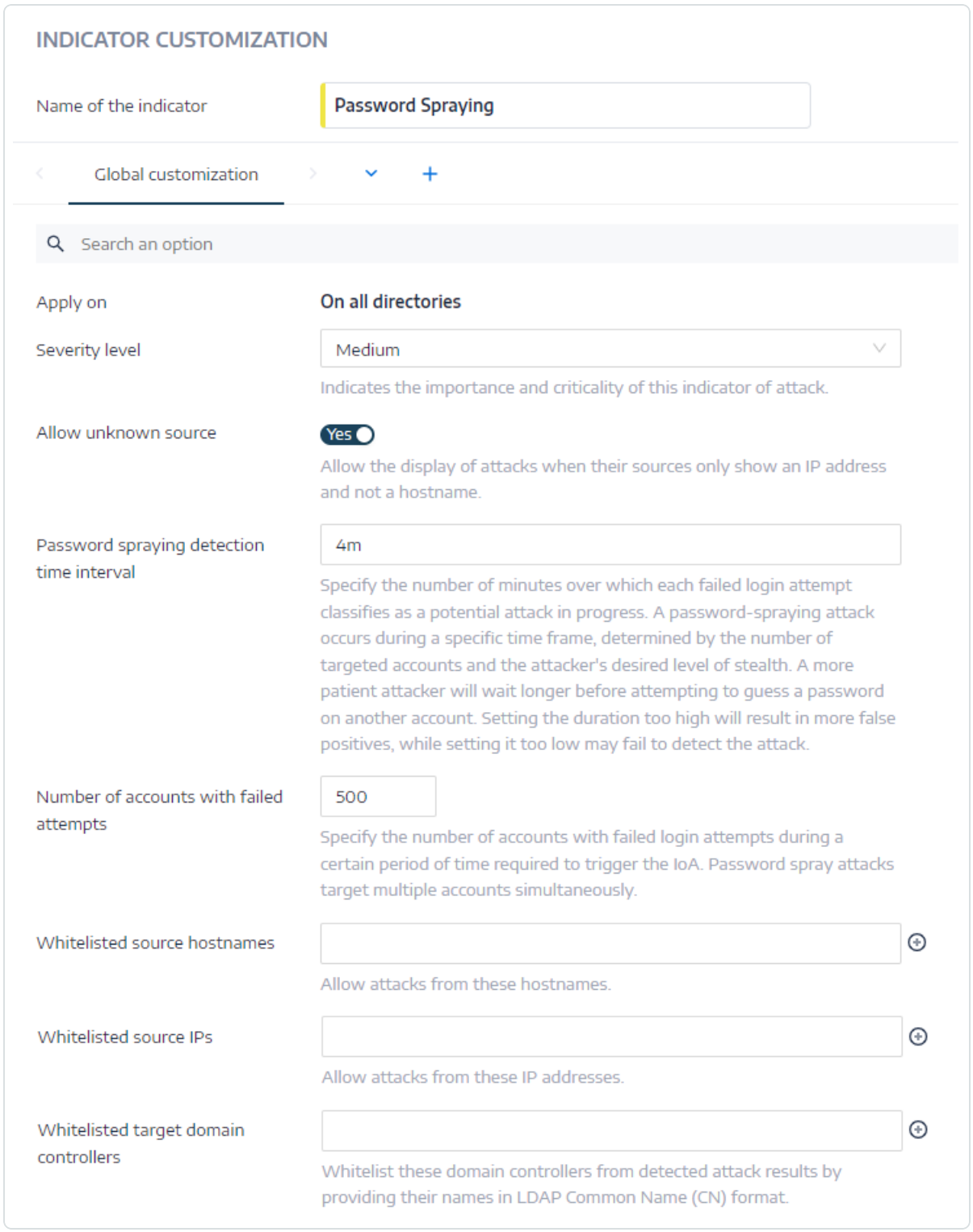
Event IDs	Audit Policies	Value
4624	├ Category: Logon/Logoff └ – Sub-category: Logon	Success
4625	├ Category: Logon/Logoff └ – Sub-category: Logoff	Failure
4771	├ Category: Account Logon └ – Sub-category: Kerberos Authentication Serviced	Failure
	Requires Sysmon extension	No
	Honey Account	No

Options

To select options for this Indicator of Attack, see Customize an Indicator of Attack.

Option Name (Type, Default Value)

Allow unknown source (Boolean, True)	
Description	Allow the display of attacks when their sources only show an IP address and not a hostname.
Recommendation	Do not change this value unless you have a specific need.
Password spraying detection time interval (String, 4m)	
Description	Specify the number of minutes over which each failed password login attempt classifies as a potential attack in progress.
Recommendation	Do not change this value unless you have a specific need.
Number of accounts with failed attempts (Integer, 500)	
Description	Specify the number of accounts with failed login attempts required to trigger the IoA.
Recommendation	For domains with fewer than 500 users, use a value smaller than the total number of users. Example: A domain with 350 users can have a value of 80.
Whitelisted source hostnames (List of strings, Empty)	
Description	Allow attacks from these hostnames.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted source IPs (List of strings, Empty)	
Description	Allow attacks from these IP addresses.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted target domain controllers (List of strings, Empty)	
Description	Whitelist domain controllers from detected attack results by providing their names in the LDAP Common Name (CN) format.
Recommendation	Do not change this value unless you have a specific need.



Yara Detection Rules

YARA rules are malware detection patterns that you can customize to identify targeted attacks and security threats specific to your environment.

Password Spraying Yara Detection Rules

```
rule invoke_domain_passwordspray
{
    meta:
        description             = "Detects Invoke-DomainPasswordSpray"
        author                  = "Tenable.AD"
        comment                 = "It can be used to perform a spraying or a bruteforce."
        reference1              =
"https://github.com/dafthack/DomainPasswordSpray/blob/master/DomainPasswordSpray.ps1"
        date = "2020-11-13"

    strings:
        $fctname                = "Invoke-DomainPasswordSpray" nocase
        $domainpasswordspray    = "DomainPasswordSpray" nocase
        $heuristic              = "Using $UserList as userlist to spray with" nocase

    condition:
        all of them
}
```

See also

- OS Credential Dumping: LSASS Memory
- Suspicious DC Password Change
- DCShadow
- DCSync
- Password Spraying
- Domain Backup Key Extraction
- Enumeration of Local Administrators
- Golden Ticket
- Kerberoasting
- Massive Computers Reconnaissance
- NTDS Extraction
- Password Guessing
- PetitPotam

- SAM Name Impersonation
- Unauthenticated Kerberoasting
- Zerologon Exploitation

PetitPotam

PetitPotam is a tool that coerces remote servers to authenticate to another machine on the network due to a Windows vulnerability. If PetitPotam targets a domain controller, an attacker can authenticate to another network machine using the domain controller's credentials.

An attacker can use PetitPotam in conjunction with PKI misconfigurations to generate a certificate to allow it to authenticate as the domain controller (such as when Active Directory Certificate Services (AD CS) web enrollments are available).

In order for this indicator-of-attack to detect PetitPotam, the IoA installation script enables automatically the Microsoft-Windows-EFS/Debug channel by adding the registry key `Microsoft-Windows-EFS/Debug` to "`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\EventLog`".

Note: If you previously set a configuration for the log retention for this specific channel, adding this registry key overrides the initial configuration, and events before this configuration are no longer visible.

Tip: Tenable recommends checking the targeted Domain Controller event logs to verify the source of the attack (account used and originating computer).

Events Auditing Policy

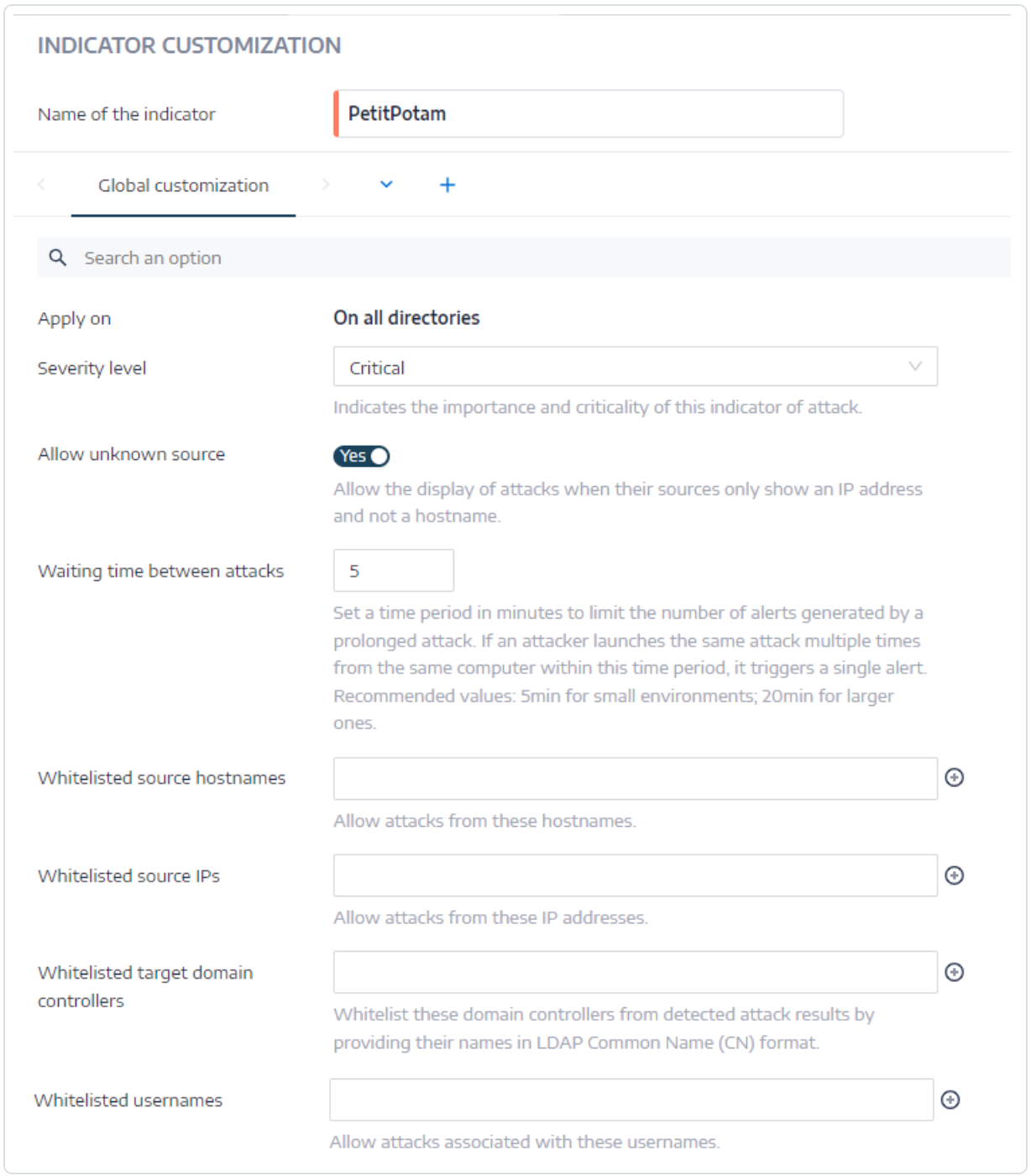
Event IDs	Audit Policies	Value
4624	├ Category: Logon/Logoff └ – Sub-category: Logon	Success
--	**Requires Sysmon extension**	No

Event IDs	Provider Name	Channel Enabled
1	Microsoft-Windows-EFS	Microsoft-Windows-EFS/Debug

Options

To select options for this Indicator of Attack, see Customize an Indicator of Attack.

Option Name (Type, Default Value)	
Allow unknown source (Boolean, True)	
Description	Allow the display of attacks when their sources only show an IP address and not a hostname.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted source hostnames (List of strings, Empty)	
Description	Allow attacks from these hostnames.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted source IPs (List of strings, Empty)	
Description	Allow attacks from these IP addresses.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted target domain controllers (List of strings, Empty)	
Description	Whitelist domain controllers from detected attack results by providing their names in the LDAP Common Name (CN) format.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted usernames (List of strings, Empty)	
Description	Allow attacks associated with these usernames.
Recommendation	Do not change this value unless you have a specific need.



Yara Detection Rules

YARA rules are malware detection patterns that you can customize to identify targeted attacks and security threats specific to your environment.

PetitPotam Yara Detection Rules

```
rule petitpotam
{
    meta:
        description = "Detects PetitPotam coerce authentication attack (Yara format for ELAT)"
        author      = "Tenable.AD"
        reference   = "https://github.com/topotam/PetitPotam"
        date        = "2021-08-01"

    strings:
        $provider    = "Microsoft-Windows-EFS"
        $eventid     = "1"
        $filenumber  = 21
        $linenumber1 = 274
        $linenumber2 = 309

    condition:
        ($provider and $eventid and $filenumber and ($linenumber1 or $linenumber2)))
}

SamAccountName Impersonation

rule samnameimpersonation
{
    meta:
        description = "Detects an attacker which is trying to exploit the vulnerability related to
sAMAccountName impersonation: CVE-2021-42287/CVE-2021-42278 (Yara format for ELAT)"
        author      = "Tenable.AD"
        reference   = "https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-42287"
        date        = "2022-01-27"

    strings:
        $provider         = "Microsoft-Windows-Security-Auditing"
        $eventid          = "4768"
        $targetusername   != "*$"

    condition:
        ($provider and $eventid and $target)
}
```

See also

- OS Credential Dumping: LSASS Memory
- Suspicious DC Password Change
- DCShadow
- DCSync
- PetitPotam

- Domain Backup Key Extraction
- Enumeration of Local Administrators
- Golden Ticket
- Kerberoasting
- Massive Computers Reconnaissance
- NTDS Extraction
- Password Guessing
- Password Spraying
- SAM Name Impersonation
- Unauthenticated Kerberoasting
- Zerologon Exploitation

SAM Name Impersonation

The **SAM Name Impersonation** Indicator of Attack (IoA) detects an attacker who tries to exploit two vulnerabilities (CVEs) that Microsoft patched silently in November 2021: CVE-2021-42278 and CVE-2021-42287. CVE-2021-42287 is critical and can lead to an elevation of privileges on the domain from a standard account without any security skills.

This IoA detects both fully automated tools such as noPac and sam-the-admin, as well as manual attacks using Rubeus and Impacket tools.

In cases where you encounter issues with the naming conventions of their computers and users, particularly instances of identical names, it may trigger alerts in the product. However, these alerts are false positives which you should disregard.

After you patch all Domain Controllers (DCs), it is safe to disable this IoA as the vulnerability it detects becomes non-exploitable.

Detection Type	Related to a Common Vulnerabilities and Exposures (CVE)	Available from Tenable Identity Exposure version
Generic IOC	Yes (CVE-2021-42278 / CVE-2021-42287)	3.15

How the attack works

An attacker can exploit the CVE-2021-42287 vulnerability by sending a Kerberos service ticket request using the S4U2self-mechanism and providing a spoofing account that does not currently exist in the Active Directory (AD). This prompts the domain controller to search whether or not a similar account name ending with `$` exists. If an account with such a `sAMAccountName` attribute exists, the attacker can compromise this account instead of the one they provided initially.

So by controlling an account that looks like a DC (that is the `sAMAccountName` = the DC name, without the ending `$`), the attacker can pretend to be this DC and elevate privileges on the domain.

In most scenarios, the attacker follows this process using a standard user account:

1. Creates a new computer account, using the AD misconfiguration of the ms-DS-MachineAccountQuota attribute.
2. Removes the Service Principal Names (SPNs) added to this newly created computer account.
3. Renames this computer to the name of a DC, dropping the ending `$`.
4. Requests a Kerberos TGT for this computer account using the password from the account creation.
5. Renames this computer to its original name by adding back the ending `$`.
6. Requests a Kerberos service ticket using the S4U2self-mechanism by presenting the previously obtained TGT, to target a service on a DC.

By default, in unhardened environments, any domain user can create up to 10 computer accounts in AD, which is the recommended quota for the ms-DS-MachineAccountQuota attribute. Even though this would harden the global AD configuration, **it is not enough** to protect the AD from this attack. In fact, this attack process can take place using a spoofing user account instead of a computer account.

Automated attack tools apply exactly the same process. But it is possible to adapt this attack to target other types of accounts that are not domain controllers, such as any workstation or server (that is tier-0 servers), the SSO accounts from AADConnect, Managed Service Accounts, etc.

How the IoA works

Based on the attack process described above, the **SAM Name Impersonation** IoA analyzes all Kerberos TGT requests to check if they are legitimate. Whenever there is a TGT request for an

account, the IoA looks in its data to see if there is another account in the AD with the same name but with a `$` at the end of its `sAMAccountName` attribute. If the IoA finds such an account, then it is an indication of an attack.

Example: The event log shows a TGT request for the account `PRIV-SRV`. If the IoA finds an account in the AD referenced by `PRIV-SRV$` in the `sAMAccountName` attribute (a user, computer, MSA, etc.), it triggers an attack alert.

Caution: In rare cases, a user and a computer may have the same `sAMAccountName` attribute. In this case, you can use the dedicated IoA option in Tenable Identity Exposure to remove future alerts. If your environment uses this practice, the IoA may not function as intended. We recommend either adopting an alternative naming convention or disabling this IoA entirely.

Specific modifications to the environment

None. Tenable Identity Exposure adapts the audit policy to meet the needs of the required Windows event logs.

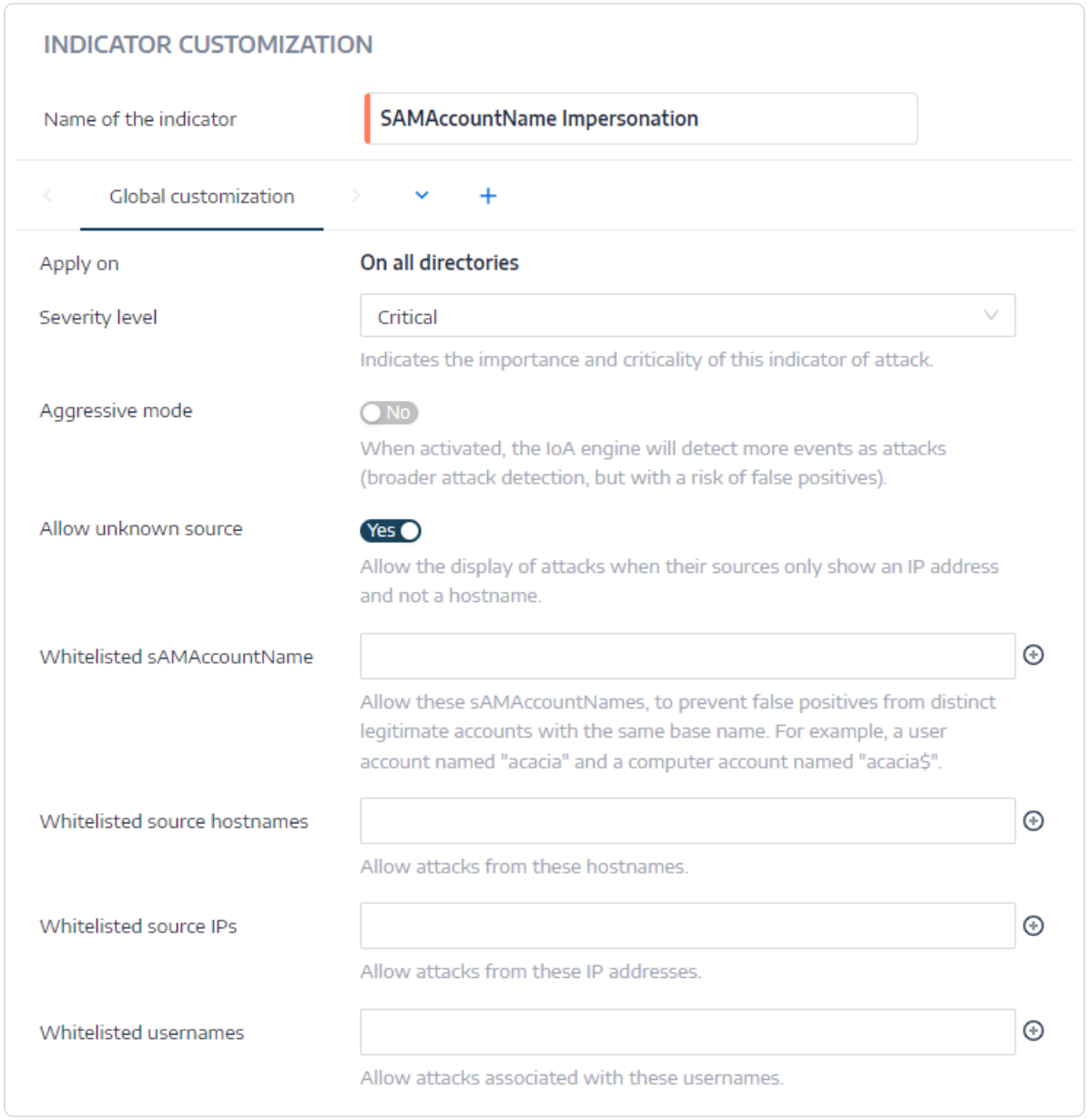
Events Auditing Policy

Provider Name	Channel	Event IDs	Audit Policies	Value
Microsoft-Windows-Security-Auditing	Security	4624	├ Category: Logon/Logoff └ – Sub-category: Logon	Success
Microsoft-Windows-Security-Auditing	Security	4768	├ Category: Account Logon └ – Sub-category: Audit Kerberos Authentication Service	Success
Other requirements				
Sysmon extension	No			
Honey Account	No			

Options

To select options for this Indicator of Attack, see Customize an Indicator of Attack.

Option Name (Type, Default Value)	
Aggressive mode (Boolean, False)	
Description	Detects more attacks but carries a risk of false positives.
Recommendation	Do not change this value unless you have a specific need.
Allow unknown source (Boolean, True)	
Description	Allow the display of attacks when their sources only show an IP address and not a hostname.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted sAMAccountName (List of strings, Empty)	
Description	Allow these sAMAccountNames, to prevent false positives from distinct accounts with the same base name. For example, a user account named `acacia` and a computer account named `acacia$`.
Recommendation	If you identify two legitimate accounts with the same base name, such as a user account named `acacia` and a computer account named `acacia$`, add them to this option.
Whitelisted source hostnames (List of strings, Empty)	
Description	Allow attacks from these hostnames.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted source IPs (List of strings, Empty)	
Description	Allow attacks from these IP addresses.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted usernames (List of strings, Empty)	
Description	Allow attacks associated with these usernames.
Recommendation	Do not change this value unless you have a specific need.



Yara Detection Rules

YARA rules are malware detection patterns that you can customize to identify targeted attacks and security threats specific to your environment.

SamAccountName Impersonation Yara Detection Rules

```
rule samnameimpersonation
{
    meta:
        description = "Detects an attacker which is trying to exploit the vulnerability related to
sAMAccountName impersonation: CVE-2021-42287/CVE-2021-42278 (Yara format for ELAT)"
        author      = "Tenable.AD"
        reference   = "https://msrc.microsoft.com/update-guide/vulnerability/CVE-2021-42287"
        date        = "2022-01-27"

    strings:
        $provider         = "Microsoft-Windows-Security-Auditing"
        $eventid          = "4768"
        $targetusername   != "*$"

    condition:
        ($provider and $eventid and $target)
}
```

See also

- OS Credential Dumping: LSASS Memory
- Suspicious DC Password Change
- DCShadow
- DCSync
- SAM Name Impersonation
- Domain Backup Key Extraction
- Enumeration of Local Administrators
- Golden Ticket
- Kerberoasting
- Massive Computers Reconnaissance
- NTDS Extraction
- Password Guessing
- Password Spraying
- PetitPotam

- Unauthenticated Kerberoasting
- Zerologon Exploitation

Unauthenticated Kerberoasting

Kerberoasting is an attack that requires authentication and targets Active Directory service account credentials for offline password cracking. This attack seeks to gain access to service accounts by requesting service tickets and then cracking the service account's credentials offline. The `Kerberoasting` Indicator of Attack (IoA) focuses on this classic Kerberoasting attack method. Another type of attack, called Unauthenticated Kerberoasting, provides a stealthier approach to execute a Kerberoasting attack by bypassing numerous detections. Advanced attackers may favor this method to remain invisible to most detection heuristics.

Contrary to the classic **Kerberoasting** IoA, the Unauthenticated Kerberoasting IoA does not require the activation of Tenable Identity Exposure's Honey Account feature.

Detection Type	Related to a Common Vulnerabilities and Exposures (CVE)	Available from Tenable Identity Exposure version
Generic IOC	No	3.43

How the attack works

A Kerberoasting attack allows an attacker to retrieve hashes of accounts with a servicePrincipalName (SPN). Typically, this attack targets a privileged account for offline password cracking.

However, using the Kerberos preauthentication, an unauthenticated attacker can perform a variant of the Kerberoasting attack that does not trigger the same event logs in the Domain Controller to allow them to bypass most classic detection methods. This attack is stealthy because the attacker would request a service ticket for a service other than the KDC (krbtgt account) using the KDC authentication service (AS), instead of its dedicated TGS service per the Kerberos protocol.

There are two main scenarios:

- An unauthenticated attacker seeks to gain initial access to any account, and not necessarily a privileged one. The attacker would begin by targeting a classic user account name (such as

"svc_sqlserver" or "helpdesk") or generating a user list via a null session.

- An authenticated attacker attempts a stealthy Kerberoasting attack to bypass most detection methods.

How the IoA works

The Unauthenticated Kerberoasting Indicator of Attack (IoA) can detect this stealthy attack by analyzing all Kerberos TGT requests to check if they are legitimate. Whenever there is a TGT request for an account, the IoA looks in its data to verify if this ticket is for a service other than the KDC (krbtgt account) or SID null (S-1-0-0).

Specific modifications to the environment

None. Tenable Identity Exposure adapts the audit policy to meet the needs of the required Windows event logs.

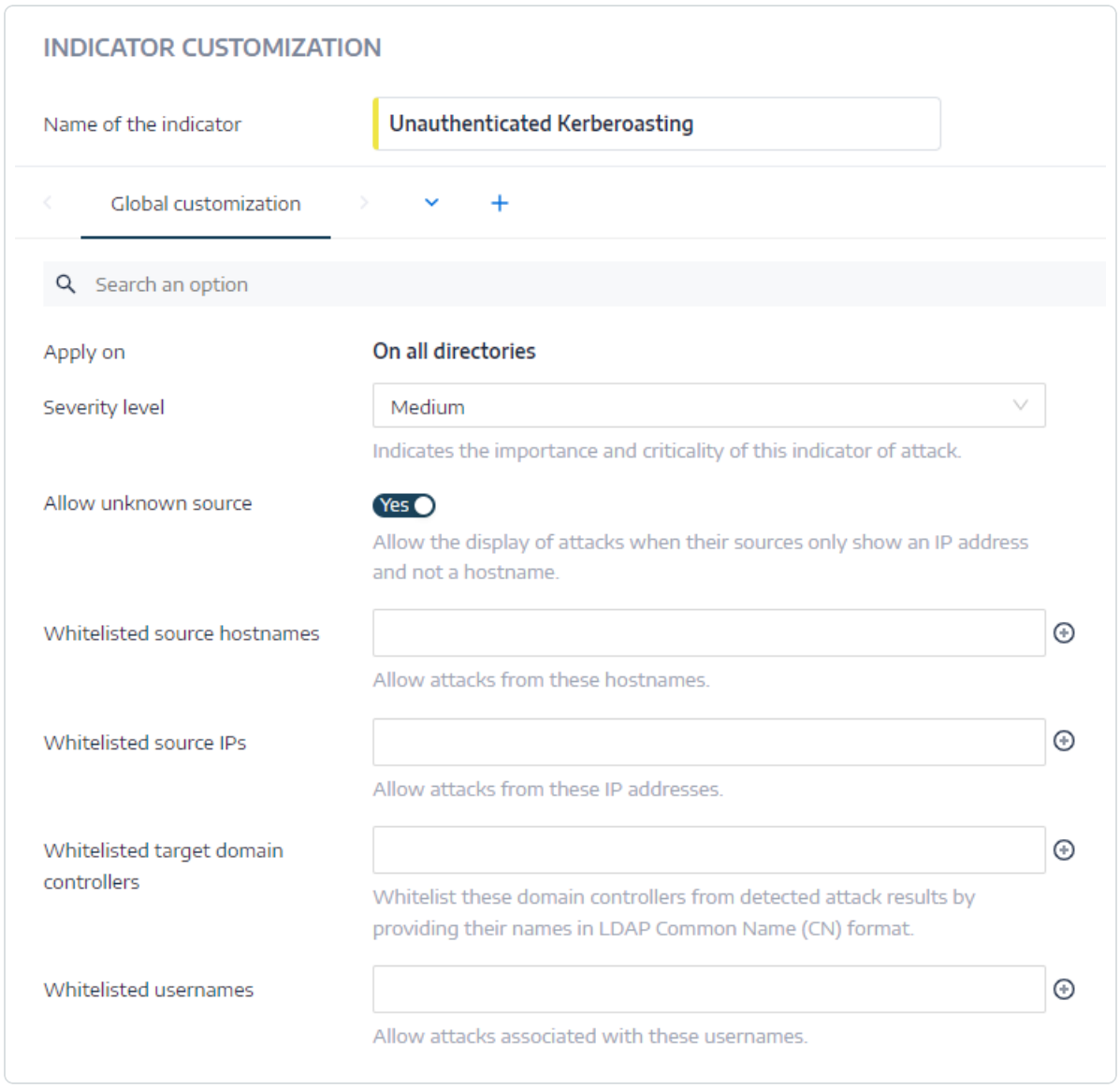
Events Auditing Policy

Provider Name	Channel	Event IDs	Audit Policies	Value
Microsoft-Windows-Security-Auditing	Security	4768	├ Category: Account Logon └ – Sub-category: Kerberos Authentication Service	Success
Microsoft-Windows-Security-Auditing	Security	4769	├ Category: Account Logon └ – Sub-category: Kerberos Service Ticket Operations	Success and Failure
Other requirements				
Sysmon extension	No			
Honey Account	No			

Options

To select options for this Indicator of Attack, see Customize an Indicator of Attack.

Option Name (Type, Default Value)	
Allow unknown source (Boolean, True)	
Description	Allow the display of attacks when their sources only show an IP address and not a hostname.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted source hostnames (List of strings, Empty)	
Description	Allow attacks from these hostnames.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted source IPs (List of strings, Empty)	
Description	Allow attacks from these IP addresses.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted target domain controllers (List of strings, Empty)	
Description	Whitelist domain controllers from detected attack results by providing their names in LDAP Common Name (CN) format.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted usernames (List of strings, Empty)	
Description	Allow attacks associated with these usernames.
Recommendation	Do not change this value unless you have a specific need.



Yara Detection Rules

YARA rules are malware detection patterns that you can customize to identify targeted attacks and security threats specific to your environment.

Unauthenticated Kerberoasting

```
rule hacktool_msil_rubeus_1
{
    meta:
        description = "The TypeLibGUID present in a .NET binary maps directly to the ProjectGuid
found in the '.csproj' file of a .NET project. This rule looks for .NET PE files that contain the
ProjectGuid found in the public Rubeus project."
```

```
        md5 = "66e0681a500c726ed52e5ea9423d2654"
        rev = 4
        author = "FireEye"
        reference1 = "https://github.com/GhostPack/Rubeus"
    strings:
        $typelibguid = "658C8B7F-3664-4A95-9572-A3E5871DFC06" ascii nocase wide
    condition:
        uint16(0) == 0x5A4D and $typelibguid
}

rule kekeo {
   meta:
      description = "Detects Kekeo"
      author = "Tenable.AD"
      reference1 = "https://github.com/gentilkiwi/kekeo/releases"
      date = "2023-02-10"
      hash1 = "1DEBB8471C513466ADAAB22978C9EEC9D3702344CCD1A294E6C7F7E2EE95CAF0"
      hash2 = "C1BDEA8A7AF27AF7634D89FD7FB8EC32701B48CCFE40C1483967D5D17B59AED3"
   strings:
      $x1 = "ERROR kuhl_m_misc_changepw ; A TGT is needed ( /tgt:filename.kirbi )" fullword wide
      $x2 = "kull_m_kerberos_asn1_ApReq_build" fullword wide
   condition:
      (uint16(0) == 0x5a4d) and $x1 and $x2
}
```

See also

- OS Credential Dumping: LSASS Memory
- Suspicious DC Password Change
- DCShadow
- DCSync
- Unauthenticated Kerberoasting
- Domain Backup Key Extraction
- Enumeration of Local Administrators
- Golden Ticket
- Kerberoasting
- Massive Computers Reconnaissance
- NTDS Extraction
- Password Guessing

- Password Spraying
- PetitPotam
- SAM Name Impersonation
- Zerologon Exploitation

Zerologon Exploitation

Tip: The **Zerologon Exploitation** Indicator of Attack (IoA) dates from 2020. If all of your domain controllers (DCs) received updates within the past three years, they are protected from this vulnerability. To determine the required patches for securing your DCs against this vulnerability, consult the information in Netlogon Elevation of Privilege Vulnerability from Microsoft. Once you've confirmed your DCs' security, you can safely deactivate this IoA to avoid unnecessary alerts.

The Netlogon Remote Protocol is a remote procedure call (RPC) interface used for user and machine authentication on domain-based networks. It also maintains and manages relationships between Domain Controllers (DCs) and their domain members as well as between DCs across domains.

The Zerologon vulnerability (CVE-2020-1472) consists of an elevation of privilege when an attacker establishes a vulnerable Netlogon secure channel connection to a domain controller using the Netlogon Remote Protocol. The vulnerability resides in a cryptographic flaw that allows an attacker to fake an authentication against a domain controller. Among possible exploitation scenarios, one can consist of updating the domain controller password to allow the attacker to compromise fully the domain. However, the Zerologon Indicator of Attack (IoA) focuses on the Netlogon authentication bypass, which is the mandatory first step before any possible exploitation scenario.

The only prerequisite for an attacker is to access the network through a vulnerable DC without any domain credentials requirement.

Detection Type	Related to a Common Vulnerabilities and Exposures (CVE)	Available from Tenable Identity Exposure version
Generic IOC	Yes (CVE-2020-1472)	3.46

How the attack works

The attack exploits a cryptographic flaw in the function used to initiate the Netlogon secure channel.

By initializing several fields to 0 (null bytes), the attacker can make several requests to a server and bypass the authentication without having valid identifiers beforehand. This operation can take place within a few seconds. Once authenticated, attackers can modify the password of any computer account or domain controller and thereby gain high privilege and persistence on the domain.

How the IoA works

The Zerologon IoA detects a failure in the Netlogon authentication process which indicates that attackers are trying to exploit the Zerologon vulnerability to gain privileges on the domain. Due to the nature of this attack, the system generates only a few distinctive events that may misinterpret the source of the attack (user account and computer.) When this IoA triggers an alert, you must first identify if the attacked domain controller has the latest security features, and then investigate whether this is an exploitation attempt.

Specific modifications to the environment

None. Tenable Identity Exposure adapts the audit policy to meet the needs of the required Windows event logs.

Events Auditing Policy

Provider Name	Channel	Event IDs	Audit Policies	Value
Microsoft-Windows-Security-Auditing	Security	4624	├ Category: Logon/Logoff └ – Sub-category: Audit Logon	Success
NETLOGON	System	5805	N/A	N/A
NETLOGON	System	5722	N/A	N/A
NETLOGON	System	5723	N/A	N/A
Other requirements				

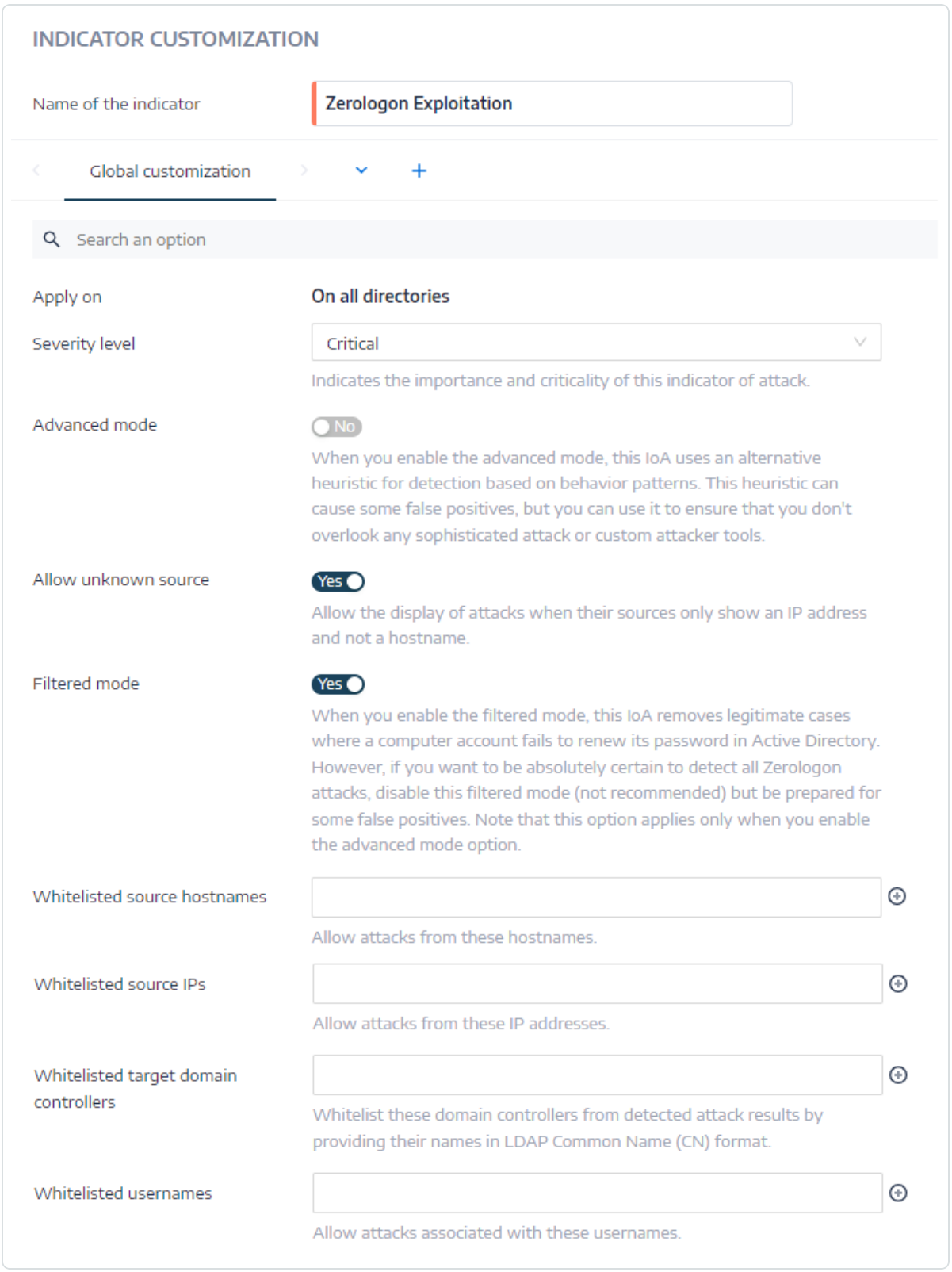
Sysmon extension	No
Honey Account	No

Options

To select options for this Indicator of Attack, see Customize an Indicator of Attack.

Option Name (Type, Default Value)	
Filtered Mode (Boolean, True)	
Description	With the filtered mode enabled, this IoA removes legitimate cases when a computer account fails to renew its password in Active Directory. However, if you want to be certain to detect all Zerologon attacks, disable this filtered mode (not recommended) but be prepared for some false positives.
Recommendation	Do not change this value unless you have a specific need (Example: The attacks are going undetected because they are attempting to bypass the heuristic to avoid detection.)
Allow unknown source (Boolean, True)	
Description	Allow the display of attacks when their sources only show an IP address and not a hostname.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted source hostnames (List of strings, Empty)	
Description	Allow attacks from these hostnames.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted source IPs (List of strings, Empty)	
Description	Allow attacks from these IP addresses.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted target domain controllers (List of strings, Empty)	

Description	Whitelist domain controllers from detected attack results by providing their names in the LDAP Common Name (CN) format.
Recommendation	Do not change this value unless you have a specific need.
Whitelisted usernames (List of strings, Empty)	
Description	Allow attacks associated with these usernames.
Recommendation	Do not change this value unless you have a specific need.



Yara Detection Rules

YARA rules are malware detection patterns that you can customize to identify targeted attacks and security threats specific to your environment.

```
rule Zerologon_exploit
{
    meta:
        vulnerability = "CVE-2020-1472"
        description = "Memory detection of Zerologon exploits"
        reference = "<https://www.secura.com/blog/zero-logon>"
        reference = "<https://www.cynet.com/zerologon>"

    strings:
        $pattern = {00 ?? ?? (0? | 10 | 11) 00 00 00 00 00 00 00 (0? | 10 | 11) 00 00 00 (4? | 5? |
6? | 7? | 2D) 00 [1-27] 00 (24 | 00) 00 00 00 ( 00 00 | 06 00 | 06 00 00 00) (0? | 10) 00 00 00 00 00
00 00 (0? | 10) 00 00 00  (4? | 5? | 6? | 7? | 2D) 00 [1-27] 00 00 00 [8] FF FF 2F 21}

    condition:
        $pattern
}

rule cve_2020_1472
{
   meta:
      description = "Detects Dirkjan cve-2020-1472.py script"
      author = "Tenable.AD"
      reference1 = "https://github.com/dirkjanm/CVE-2020-1472/blob/master/cve-2020-1472-exploit.py"
      date = "2023-03-22"
      hash1 = "50AF4367EADD55236D085D8221815EA06992D6C0E1AB3ED6848DC3BDACA6F7DD"
   strings:
      $x1 = "0x212fffff" fullword
      $x2 = "NETLOGON_SECURE_CHANNEL_TYPE.ServerSecureChannel" fullword
      $x3 = "NetrServerPasswordSet2" fullword
   condition:
      $x1 and $x2 and $x3
}
```

See also

- OS Credential Dumping: LSASS Memory
- Suspicious DC Password Change
- DCShadow
- DCSync
- Zerologon Exploitation
- Domain Backup Key Extraction
- Enumeration of Local Administrators

- Golden Ticket
- Kerberoasting
- Massive Computers Reconnaissance
- NTDS Extraction
- Password Guessing
- Password Spraying
- PetitPotam
- SAM Name Impersonation
- Unauthenticated Kerberoasting