TRAPX SECURITY

DECEPTIONGRID™

v. 7.2

Security Deployment Guide

Digitally signed by Cojocari Andrei
Date: 2022.01.14 15:04:52 EET
Reason: MoldSign Signature
Location: Moldova

TrapX® Security, June 2021

trapx.com

Contents

Preface

This guide is about deploying DeceptionGrid deception and emulation in your organizational network, subsequent to initial setup.

For initial installation, see the *DeceptionGrid Installation Guide*.

For further setup, and for ongoing system (non-security) administration, see the *DeceptionGrid Administration Guide*.

For event management and security analysis, see the *DeceptionGrid Security Handling & Analysis Guide*.

Overview: Trap and Token Deployment

TrapX Security® DeceptionGrid includes a multi-tiered set of mechanisms for deception, emulation, and interception, for deployment throughout an organization. These several tiers, when combined in the context of an organizationally-relevant deception and interception strategy, provide a powerful security solution for deterrence and interception of human attackers and active malware.

Recorded events from all tiers are analyzed and collected in the single organizational TrapX Security Operations Console (TSOC), where they are displayed in a central web interface. The entire system is centrally managed from TSOC.

DeceptionGrid's deception and interception tiers are briefly described in the following sections; full deployment instructions are in subsequent chapters.

For system architecture, see the *DeceptionGrid Administration Guide*.

In This Section

Emulation Traps

Configured on DeceptionGrid Appliance interfaces, emulation traps respond to attackers as though they were real devices typical of the organization. Multiple emulation traps can be connected to each existing organizational network VLAN, each configured as an organizationally-relevant device type. You can also open relevant traps to the internet, to cover remote devices such as employees' working from home.

For trap realism, you can configure the traps in several ways. Various services as appropriate to emulation type respond realistically to attackers (medium interaction); you can upload fake but realistic data to the traps for exposure to attackers over these services. Additionally, you can specify basic monitoring (low interaction) of any custom port.

To deploy and configure emulation traps, see Emulation Trap Deployment on page 17.

Full OS traps

Full OS traps provide a high level of realism and full attack monitoring, by installing the TrapX Full OS Trap agent on a full (virtual) computer. The host computer can be configured with any software, data, and settings, and the agent will monitor and record not only inbound connections but also outbound activity (for example, if an attacker attempts to connect from the full OS trap to another endpoint or to the internet).

In addition, you can use a full OS trap to transparently provide a real service to respond to attackers of emulation traps, and full monitoring of those attacks. This is achieved by proxying emulation traps' services to a full OS trap.

Optionally, Full OS includes CryptoTrap, a network share which Full OS automatically and continuously populates with document and media files to slow down and deceive detected ransomware attacks. A deception token for this share is managed in TSOC. With CryptoTrap, Full OS also intelligently detects ransomware behavior anywhere in the file system and accordingly records an enriched event specifically identified as Ransomware.

To deploy and configure full OS traps, see Full OS Trap Deployment on page 41.

Deception Tokens

Deception tokens are lures that are deployed across actual organizational endpoints, inside and outside the organizational network - Windows and Linux, servers and workstations. The tokens are various data items and configuration entries that point to DeceptionGrid emulation trap and full OS trap services, causing attackers that encounter the tokens to then attempt to connect to those services, triggering an alert. Deception tokens significantly add to the deception power of the traps, reducing the time it takes to detect attacks and defend against them.

Tokens can be designed and distributed so that multiple tokens across disparate endpoints create the illusion of the target traps being real and significant organizational assets. Agentless and light, deception tokens are easily deployable on servers and workstations.

To configure and deploy deception tokens, see Deception Token Deployment on page 45.

Planning Your DeceptionGrid Deployment

DeceptionGrid provides a powerful toolset for implementing a deception strategy in an organizational network. Every organization is different, so to maximize trap realism you'll need to plan your DeceptionGrid deployment's details as appropriate for your organization. It is recommended to have a full deployment plan in place before beginning to implement that deployment.

The following are recommended high-level best practices for planning and designing the deployment of DeceptionGrid deception and emulation throughout your organization.

Trap deployment should be planned specifically for each of your existing network assets: server segments (see Planning Emulation in a Server Network Segment below), and workstation segments (see Planning Emulation in a Workstation Network Segment on page 7). In addition, you should plan to deploy deception tokens (see Planning Deception Token Deployment on page 15) throughout your organization, to direct attackers to deployed traps.

If you have a deployed Network Access Control (NAC) system, also consider creating a dedicated network segment to include only traps (Full OS traps and Emulation traps). Upon identifying an attack on other network segments, you can divert the attacker to this dedicated trap segment, allowing you to continue gathering intelligence and occupying the attacker.

In This Section

Planning Emulation in a Server Network Segment

The following are some general high-level emulation and full OS trap recommendations appropriate for a typical organizational server network:

- Deploy 5-7 emulation traps (see Emulation Trap Deployment on page 17) per network segment, with a variety of emulation types and services typical of the network's existing servers. These should include emulated servers, network switches, and IoT devices as relevant. You can check existing subnet trap coverage (see Detection Coverage on page 8).
- Deploy a Full OS trap (see Full OS Trap Deployment on page 41) on a Windows Server host, with CryptoTrap for enhanced anti-ransomware.
- In general, traps' names should follow the network's hostname convention. Some should have names or name parts representing server types that are attractive to attackers, such as those that are typically not well protected, and so would be easy to penetrate. For example:

    - **host023_dev**
    - **test_server456**
    - **server_ERP**
- Make sure to enable NBNS on all Windows emulations, to create realistic network traffic.
- To enhance realism, configure any of the emulation traps' Web, MSSQL, RDP and Active Directory services as proxies to real services with fake or obfuscated data. Depending on the service, these real services can be on a Full OS trap - in which case configure the emulation trap in Proxy Mode (see Configuring Emulation Trap Proxy to Full OS Trap on page 43), which enables fuller event information; or on actual organizational servers, in which case configure the emulation trap service with the server's address (see Emulation Service Configuration on page 30).
- For any in-house application, configure a Custom emulation with the relevant port number.
- Deploy a Cisco network switch emulation, and via its SSH or Telnet service configure its Running Configuration with a fake SNMP community, fake network monitor server IP address, and fake credentials. Also configure these values in your organizational monitoring system watchlist to intercept their use in other systems.
- To emulate a file server, configure a server emulation with SMB service, open a spin data session to it, and create a directory structure reflecting your organization's file server(s). Upload public, non-sensitive but real or realistic data. Some examples of data attractive to attackers are:
    - Saved email messages containing credentials
    - Excel files
    - Network diagrams

  It is recommended to password-protect files, to occupy attackers.
- For some emulated services (such as SMB, FTP, and Linux SSH), configure authentication to slow down attackers; leave others with no authentication.
- For FTP emulations, make sure to configure the FTP banner as relevant for your organizational FTP deployment.

Planning Emulation in a Workstation Network Segment

The following are some general high-level emulation trap recommendations appropriate for a typical organizational workstation network:

- Deploy 3-5 emulation traps (see Emulation Trap Deployment on page 17) per workstation network segment, with a variety of emulation types typical of the network. These should include emulated Windows (various versions), Mac workstations, and IoT devices as relevant. You can check existing subnet trap coverage (see Detection Coverage on page 8).

- In general, traps' names should follow the network's hostname convention. Some should have names or name parts that are attractive to attackers, such as those that would be likely to contain credentials to other assets. For example:
  - **IT Admin**
  - **HelpDesk**
  - **DBA**
- Make sure to enable NBNS on all Windows emulations, to create realistic network traffic.
- Deploy a Full OS trap (see Full OS Trap Deployment on page 41) on a Windows 7 host, with CryptoTrap for enhanced anti-ransomware.
- In emulated workstations' SMB services, configure the C$ drive to reflect typical user directory structure, and upload fake personal files such as credentials to IT systems and family pictures.
- On a workstation emulation, create a shared network drive with unrestricted access, emulating an employee who bypassed organizational policy and exposed their disk drive.

Detection Coverage

To facilitate educated trap deployment decisions, TSOC displays information on trap coverage of organizational network subnets and of remote devices.

Coverage information can be based on analysis, or on actual attack simulations.

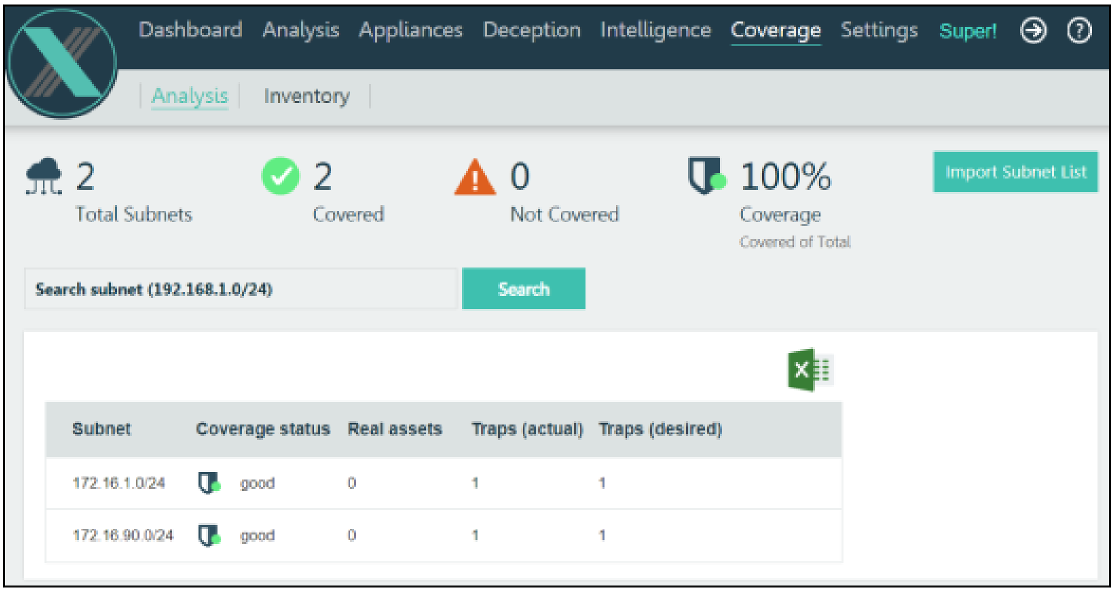
In This Section

Coverage Analysis of Organizational Networks

To facilitate educated trap deployment decisions, TSOC displays information on trap coverage of organizational network subnets.

When enabled, TSOC analyzes coverage for all known subnets, from interface configuration (see Configuring Network Interfaces for Traps on page 17), asset inventory, and uploaded subnet list as below. You can configure the criteria for considering subnets adequately covered by traps.

To view coverage analysis, in TSOC go to **Coverage**:



2 Total Subnets

2 Covered

0 Not Covered

100% Coverage
Covered of Total

Search subnet (192.168.1.0/24)

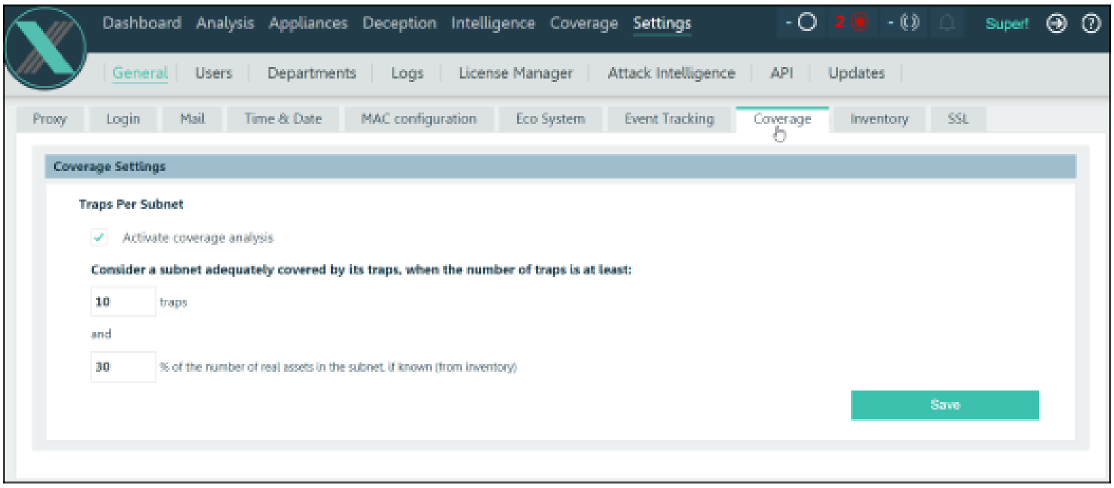
Subnet	Coverage status	Real assets	Traps (actual)	Traps (desired)
172.16.1.0/24	good	0	1	1
172.16.90.0/24	good	0	1	1

Analysis results include:

- Summaries (top row): The **Total** number of know subnets; the numbers of adequately **Covered** and **Not** adequately covered subnets, and overall **Coverage** - the adequately Covered subnets as a percentage of the Total.
- Details (table, downloadable): For each subnet, its numbers of real assets, traps, and desired number of traps to be adequately covered.

To upload a subnet list specifically for coverage analysis, prepare a CSV file, either in format **address**, **mask** (example row: **192.168.1.10, 255.255.255.0**) or in CIDR format. Then click **Import Subnet List** and upload.

To activate coverage analysis and to configure the criteria for considering subnets adequately covered by traps, in TSOC go to **Settings** > **General** > **Coverage**:



Coverage Settings

Traps Per Subnet

☑ Activate coverage analysis

Consider a subnet adequately covered by its traps, when the number of traps is at least:

10 traps

and

30 % of the number of real assets in the subnet, if known (from inventory)

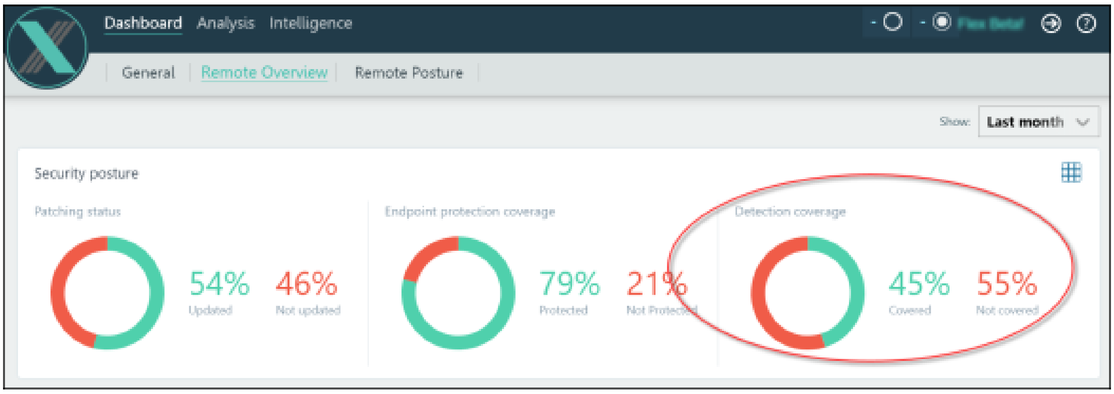
A subnet is considered adequately covered if the number of traps in the subnet is at least the defined number, AND the defined percentage of real assets in the subnet, if known from asset inventory. For example, with settings **10**, **30**, to be adequately covered a subnet with 35 real assets requires 11 traps (rounded up from 0.3 x 35 = 10.5); a subnet with 30 real assets requires 10 traps (by the first setting). For subnets for which there is no asset inventory, only the first setting is considered.

Coverage Analysis of Remote Devices

To facilitate educated remote-trap deployment decisions, TSOC displays information on remote-trap detection coverage of remote devices (outside the organizational network), providing realistic estimates of attack detection.

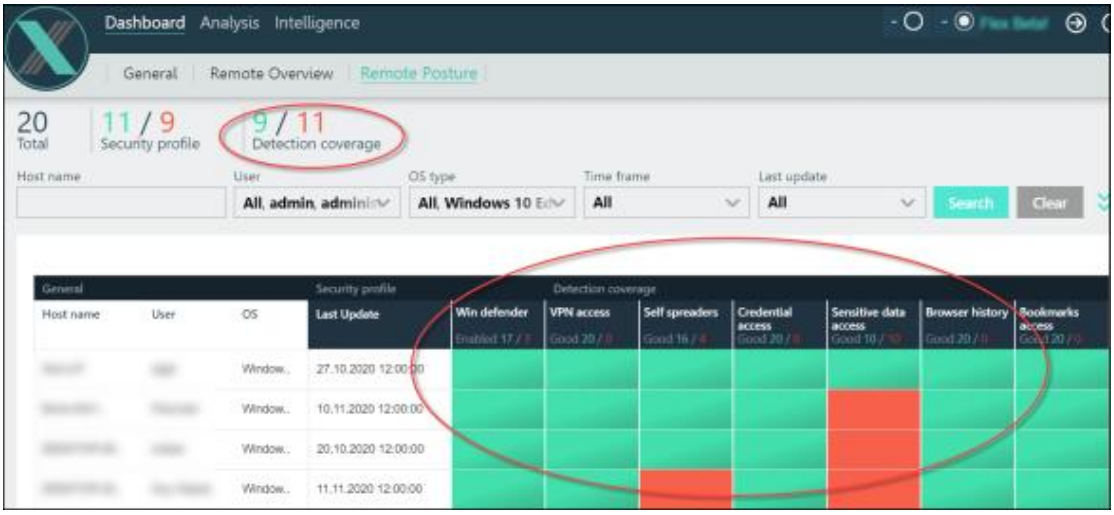
Deployment testing on remote devices is performed upon deception token distribution, by the token installer, and results are sent to TSOC.

The general state of detection coverage in remote devices is displayed in **Dashboard** > **Remote Overview**, under **Detection coverage**:



Displayed information is from latest token distribution, for devices to which tokens were recently distributed (within the time frame selected by **Show**). Of those devices, those that are covered for all tested attack types are here considered Covered. **Not covered** devices require attention to provide coverage for at least one attack type.

Per-device remote-device detection coverage details are displayed in **Dashboard** > **Remote Posture**:



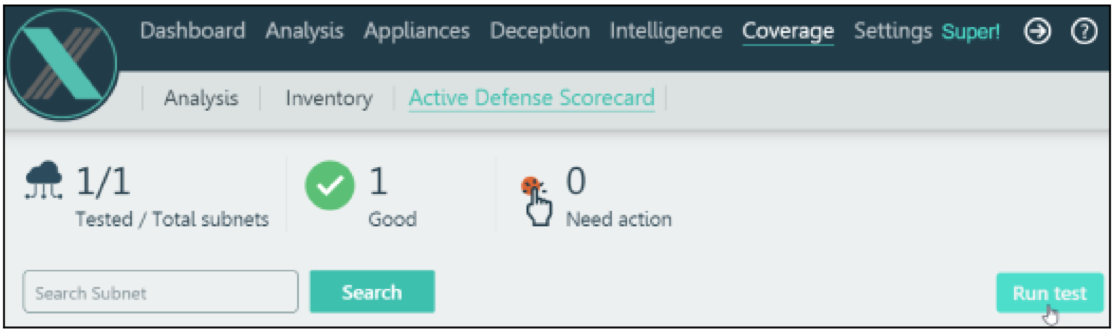
General			Security profile	Detection coverage						
Host name	User	OS	Last Update	Win defender Enabled 17 / 3	VPN access Good 20 / 0	Self spreaders Good 16 / 4	Credential access Good 20 / 0	Sensitive data access Good 10 / 10	Browser history Good 20 / 0	Bookmarks access Good 20 / 0
[redacted]	[redacted]	Window..	27.10.2020 12:00:00	🟩	🟩	🟩	🟩	🟩	🟩	🟩
[redacted]	[redacted]	Window..	10.11.2020 12:00:00	🟩	🟩	🟩	🟩	🟥	🟩	🟩
[redacted]	[redacted]	Window..	20.10.2020 12:00:00	🟩	🟩	🟩	🟩	🟥	🟩	🟩
[redacted]	[redacted]	Window..	11.11.2020 12:00:00	🟩	🟩	🟥	🟩	🟥	🟩	🟩

Active Defense Scorecard

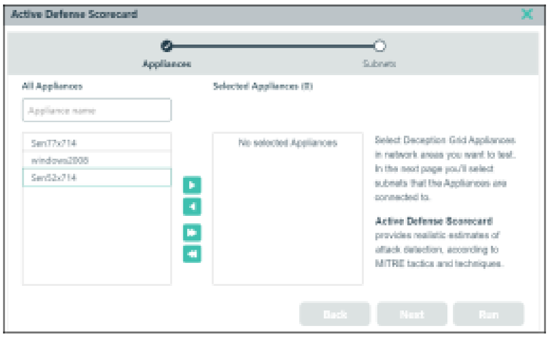
To facilitate educated trap deployment decisions, DeceptionGrid can perform realistic estimates of attack detection, according to MITRE Enterprise tactics and techniques, for specified per-Appliance connected organizational subnets. The specific tests are listed below.

Results are listed per-subnet, and also appear in the Event Analyzer as simulated detailed events.

To run an Active Defense test, in TSOC go to **Coverage** > **Active Defense Scorecard** and click **Run test**:

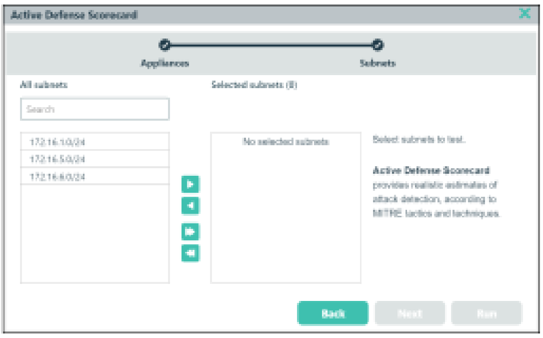


Specify **Appliances**, **Subnets** connected to them, and click **Run**:



Select Deception Grid Appliances in network areas you want to test. In the next page you'll select subnets that the Appliances are connected to.

Active Defense Scorecard provides realistic estimates of attack detection, according to MITRE tactics and techniques.

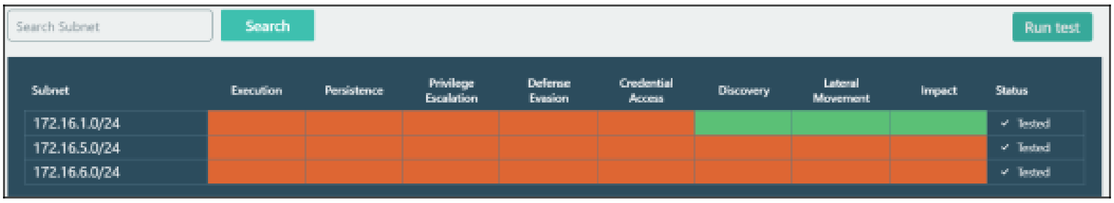


Select subnets to test.

Active Defense Scorecard provides realistic estimates of attack detection, according to MITRE tactics and techniques.

In some cases, the test may take more than 30 minutes.

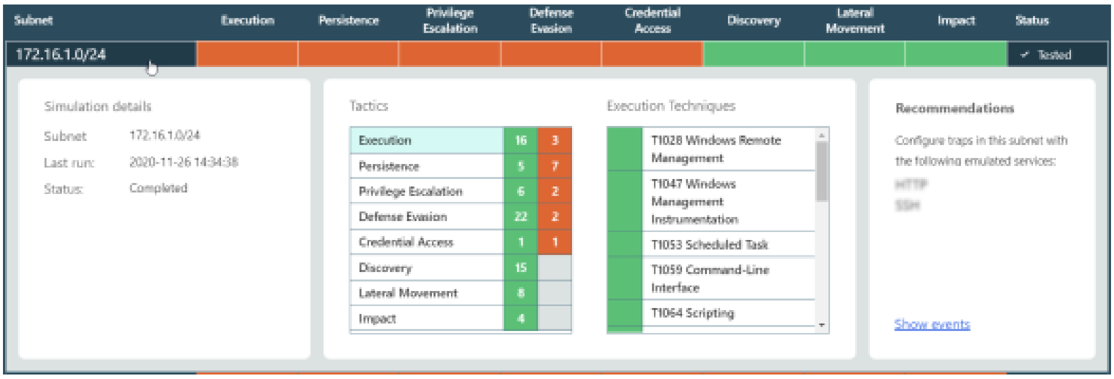
When available, the latest result set appears on the same page, by subnet and MITRE Enterprise tactic:



Subnet	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Impact	Status
172.16.1.0/24	🟧	🟧	🟧	🟧	🟧	🟩	🟩	🟩	✓ Tested
172.16.5.0/24	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	✓ Tested
172.16.6.0/24	🟧	🟧	🟧	🟧	🟧	🟧	🟧	🟧	✓ Tested

If a trap interface does not have connectivity, the row is marked gray.

For per-technique results, and recommendations, click a row:



Subnet	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Impact	Status
172.16.1.0/24	🟥	🟥	🟥	🟥	🟥	🟩	🟩	🟩	✓ Tested

Simulation details

Subnet: 172.16.1.0/24
Last run: 2020-11-26 14:34:38
Status: Completed

Tactics

Execution	16	3
Persistence	5	7
Privilege Escalation	6	2
Defense Evasion	22	2
Credential Access	1	1
Discovery	15	
Lateral Movement	8	
Impact	4	

Execution Techniques

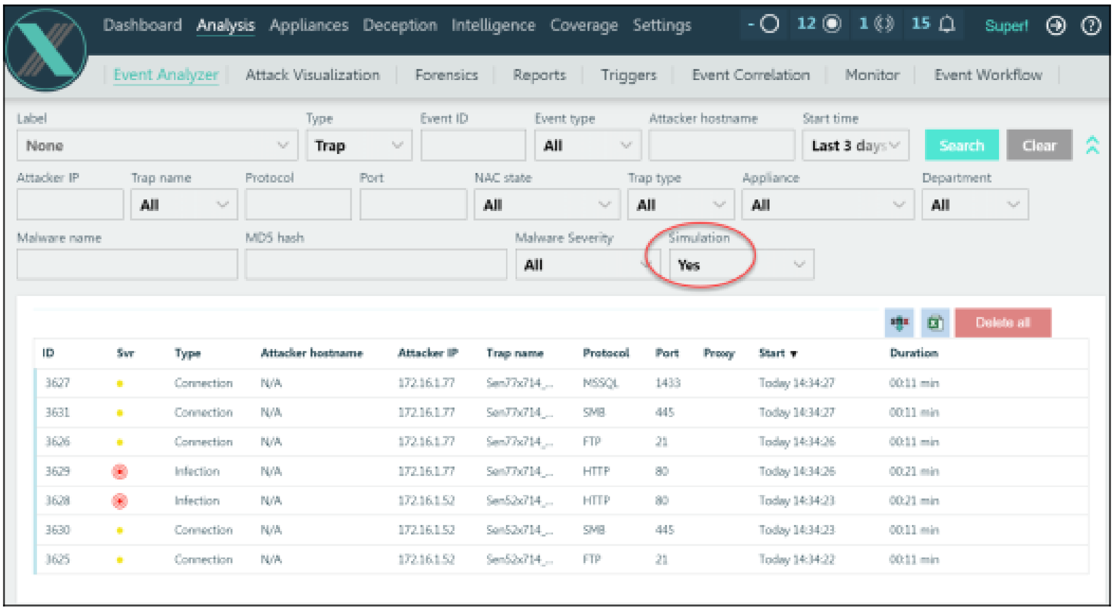
- T1028 Windows Remote Management
- T1047 Windows Management Instrumentation
- T1053 Scheduled Task
- T1059 Command-Line Interface
- T1064 Scripting

Recommendations

Configure traps in this subnet with the following emulated services:
HTTP
SSH

Show events

Individual simulations appear as events in the Event Analyzer, when the **Simulation** filter is set to **Yes**:



Label: None
Type: Trap
Event ID:
Event type: All
Attacker hostname:
Start time: Last 3 days
Attacker IP:
Trap name: All
Protocol:
Port:
NAC state: All
Trap type: All
Appliance: All
Department: All
Malware name:
MD5 hash:
Malware Severity: All
Simulation: Yes

ID	Svr	Type	Attacker hostname	Attacker IP	Trap name	Protocol	Port	Proxy	Start ▾	Duration
3627	●	Connection	N/A	172.16.1.77	Sen77x714_...	MSSQL	1433		Today 14:34:27	00:11 min
3631	●	Connection	N/A	172.16.1.77	Sen77x714_...	SMB	445		Today 14:34:27	00:11 min
3626	●	Connection	N/A	172.16.1.77	Sen77x714_...	FTP	21		Today 14:34:26	00:11 min
3629	●	Infection	N/A	172.16.1.77	Sen77x714_...	HTTP	80		Today 14:34:26	00:21 min
3628	●	Infection	N/A	172.16.1.52	Sen52x714_...	HTTP	80		Today 14:34:23	00:21 min
3630	●	Connection	N/A	172.16.1.52	Sen52x714_...	SMB	445		Today 14:34:23	00:11 min
3625	●	Connection	N/A	172.16.1.52	Sen52x714_...	FTP	21		Today 14:34:22	00:11 min

You can also reach this view, filtered for a single subnet, from the scorecard details, by expanding the subnet's row and clicking **Show events**.

Active Defense Scorecard is not available for Full OS traps.

Attack detection tests

For all emulated services listed in test results, attack detection testing is performed by establishing a connection to the service, and the result of this test is listed for technique # **T1021: Remote Services**. For emulated SMB, the result is listed for several techniques.

Additionally, for HTTP and SSH emulated services, the following specific per-technique tests are performed.

HTTP Tests

For HTTP emulated services, HTTP requests to the following URLs are run on the trap interface.

Technique	***Technique #***	***URL***
Execution through API	T0871	/debug/clip.html

Technique	*Technique #*	*URL*
Exploitation for Privilege Escalation	T1068	/login.cgi /w00tw00t.at.blackhats.romanian.anti-sec:)
Exploitation of Remote Services	T1210	/login/cgi /w00tw00t.at.blackhats.romanian.anti-sec:) /cgi-bin/config.exp

SSH Tests

For SSH emulated services, the following SSH commands are run on the trap interface.

Technique	*Technique #*	*Command*
Service Stop	T0881	systemctl stop interfaces service interfaces stop
System Network Configuration Discover	T1016	ifconfig ping 1.1.1.1
Remote System Discovery	T1018	cat /etc/hosts
Obfuscated Files or Information: Compile After Delivery	T1027.004	gcc Linux.Backdoor.c g++ Linux.Backdoor.c
System Owner/User Discovery	T1033	w who
Boot or Logon Initialization Scripts: Rc.common	T1037.004	cat /etc/rc.common
System Network Connections Discovery	T1049	netstat lsof
Scheduled Task/Job	T1053	cat /etc/crontab cat /etc/cron.d cron at 09:00 -f /home/linuxize/script.sh
Process Discovery	T1057	ps
Command and Scripting Interpreter	T1059	python apt-get install powershell cat updater.vbs java --list-modules
Permission Groups Discovery	T1069	groups ldapsearch
Indicator Removal on Host	T1070	rm -rf /var/log

Technique	***Technique #***	***Command***
Indicator Removal on Host: Clear Command History	T1070.003	unset HISTFILE export HISTFILE=0 history -c rm ~/.bash_history
Indicator Removal on Host: File Deletion	T1070.004	rm backupfile.txt
System Information Discovery	T1082	uname
File and Directory Discovery	T1083	ls find locate LuaBot
Account Discovery	T1087	whoami w cat /etc/passwd groups id
System Time Discovery	T1124	date
Network Share Discovery	T1135	df mount
Create Account	T1136	useradd -d
File and Directory Permissions Modification	T1222	chown root /u chmod 754 Tsunami
Service Stop	T1489	systemctl stop service networking stop
System Shutdown/Reboot	T1529	poweroff shutdown -h
Create or Modify System Process: Systemd Service	T1543.002	cat /etc/systemd/system cat /user/lib/systemd/system/ cat /home//.config/system/user systemctl
Event Triggered Execution: .bash_profile and .bashrc	T1546.004	cat ~/.bash_profile cat ~/.bashrc cat /etc/rc.local
Event Triggered Execution: Trap	T1546.005	trap
Abuse Elevation Control Mechanism: Setuid and Setgid	T1548.001	chmod 777 Linux.Encoder.1
Abuse Elevation Control Mechanism: Sudo and Sudo Caching	T1548.003	sudo -h
Unsecured Credentials: Bash History	T1552.003	cat .bash_history

Technique	*Technique #*	*Command*
Unsecured Credentials: Private Keys	T1552.004	cat saved_session.key cat saved_session.pgp cat saved_session.gpg cat saved_session.ppk cat saved_session.p12 cat saved_session.pem cat saved_session.pfx cat saved_session.cer cat saved_session.p7b cat saved_session.asc cat saved_session.ssh
Impair Defenses: Impair Command History Logging	T1562.003	echo \"HISTCONTROL=ignoreboth\" >>~/.bashrc
Hide Artifacts: Hidden Files and Directories	T1564.001	cat .GafGyt

Planning Deception Token Deployment

Deception tokens are lures that are deployed across actual organizational endpoints, inside and outside the organizational network - Windows and Linux, servers and workstations. The tokens are various data items and configuration entries that point to DeceptionGrid emulation trap and full OS trap services, causing attackers that encounter the tokens to then attempt to connect to those services, triggering an alert. Deception tokens significantly add to the deception power of the traps, reducing the time it takes to detect attacks and defend against them.

Tokens can be designed and distributed so that multiple tokens across disparate endpoints create the illusion of the target traps being real and significant organizational assets. Agentless and light, deception tokens are easily deployable on servers and workstations.

The following are some general high-level recommendations for deploying deception tokens (see Deception Token Deployment on page 45) throughout a typical organization.

- Deception tokens point to specific server emulations. So, tokens should be configured and deployed separately for each organizational location or unit that interacts with a specific set of servers.

- Deploy the following token types to all relevant endpoints, for each relevant server trap:
  - **SMB Network Share**: To traps with SMB service, and to CryptoTrap.
  - **ODBC**: To traps with MS SQL Server service.
  - **RDP**: To traps with RDP service.
  - **Cached Credentials** (with tracking by SIEM)
  - **Deceptive Files**: To traps with web service or DeceptiveFileListener.
- To IT department endpoints, deploy the following token types:
  - **SSH / PuTTY**: To traps with SSH service
  - **WinSCP Session**: To traps with SSH service.
- To users' endpoints that access critical web applications, deploy **Browser History / Credentials / Bookmark** tokens. It is recommended to educate users regarding these tokens, which are not hidden from users, to prevent false positives.
- On organizational domain controllers, deploy Active Directory tokens for traps (except for traps on DHCP interfaces).
- On remote branch users' endpoints, which access the domain controller via a WAN (and therefore may not have a local DNS server), deploy **Hostname** tokens.

Emulation Trap Deployment

Configured on DeceptionGrid Appliance interfaces, emulation traps respond to attackers as though they were real devices typical of the organization. Multiple emulation traps can be connected to each existing organizational network VLAN, each configured as an organizationally-relevant device type. You can also open relevant traps to the internet, to cover remote devices such as employees' working from home.

For trap realism, you can configure the traps in several ways. Various services as appropriate to emulation type respond realistically to attackers (medium interaction); you can upload fake but realistic data to the traps for exposure to attackers over these services. Additionally, you can specify basic monitoring (low interaction) of any custom port.

Before deploying traps, plan your deployment (see Planning Your DeceptionGrid Deployment on page 6).

For deployment of emulation traps, you need to first configure network interfaces (see Configuring Network Interfaces for Traps below).

For scalable, automatable interface and trap provisioning, including customization of emulation types, you can implement client scripts (see relevant guides).

In This Section

Configuring Network Interfaces for Traps

For deployment of emulation traps (see Emulation Trap Deployment above), you need to first configure appropriate (virtual) network interfaces.

In This Section

Network Interfaces for Traps: Overview

In This Section

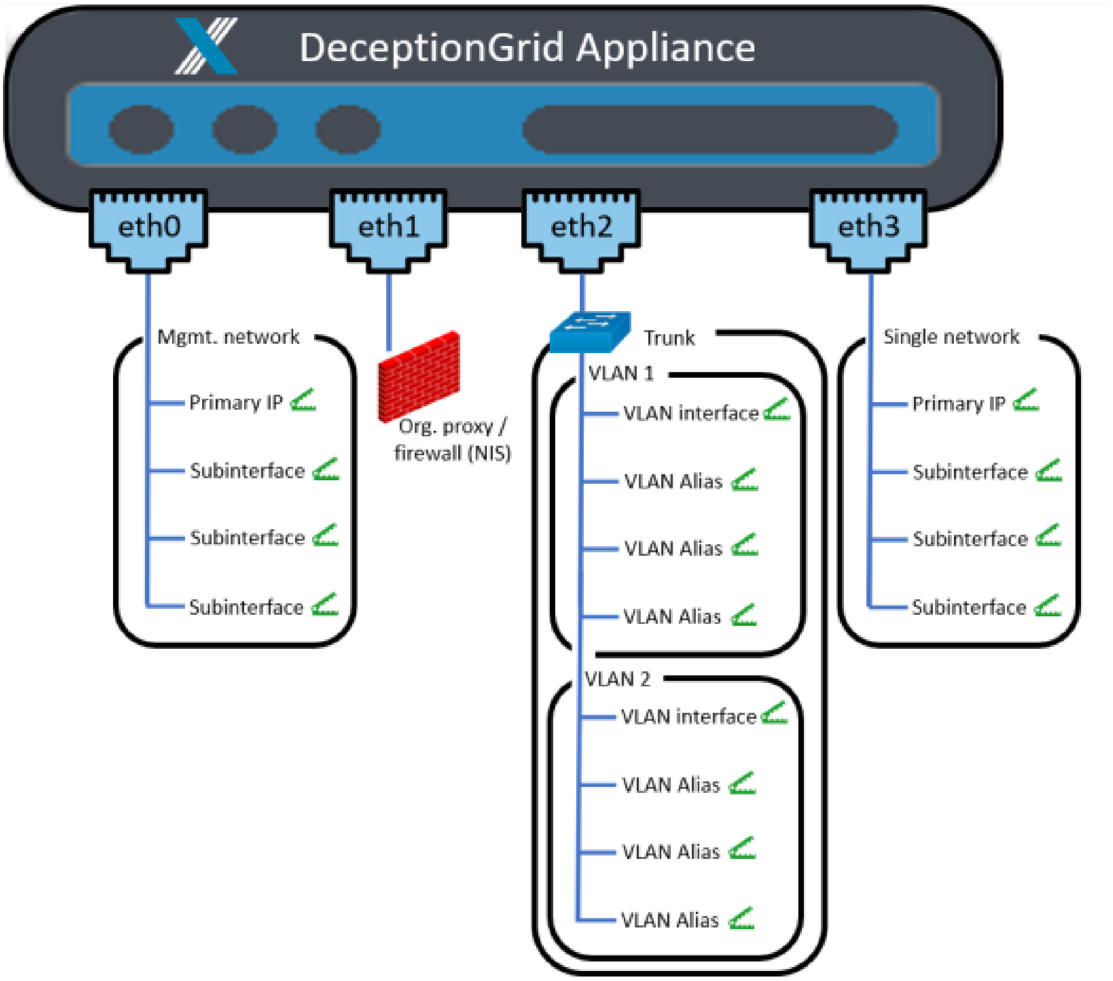
Introduction to Trap Interfaces

Emulation traps, with their emulation details, are configured on DeceptionGrid Appliance interfaces (physical or virtual), each with its own IP address. Attackers that connect to these interfaces receive emulated responses according to trap emulation configuration, and the connection is recorded as an event. Before you can configure traps, you need to configure the interfaces on which the traps will be deployed.

Upon initial setup, a DeceptionGrid Appliance has three or four interfaces named **eth0**, **eth1** etc. To these parent interfaces, you can add virtual child interfaces to enable multiple traps in connected networks.

You can bulk-configure interfaces and traps (see Bulk-Configuring Interfaces and Traps on page 36).

The way in which interfaces and child interfaces are managed depends on how the parent interface was connected during setup, as in the following sections. Here's a setup example:



If you'll be deploying Remote traps, they'll need (virtual) interfaces that are open to the internet.

Trunk connection

In environments where network switches (virtual or physical) are organizationally managed (as opposed to cloud environments such as AWS and Azure), **eth2** was likely connected to a network trunk, to enable deploying traps to multiple VLANs; **eth3** may also have been connected to another network trunk. In the example diagram, eth2 is connected to a network trunk.

In this case, the parent interface (eth2 / eth3) does not have its own IP address, and cannot be a trap. Instead, for each VLAN in your network you add to the parent interface a virtual **VLAN** child interface, with an IP address (static or DHCP) in the VLAN. This VLAN interface can be configured with a trap.

For additional traps in the same VLAN, add to the VLAN interface grandchild **VLAN Alias** interfaces, with static IP addresses in the VLAN.

Configure interfaces in a trunk connection (see Configuring Interfaces in a Trunk Connection on page 20)

Single-network connection

The **eth0** interface is always connected to a single management network, through which the Appliance communicates with TSOC. In addition, some other interfaces may have been connected to single networks rather than to network trunks (in cloud environments like AWS or Azure, this is the only option for additional networks). In the example diagram, eth0 and eth3 are connected to single networks.

An interface connected to a single network has its own IP address, which can be configured with an emulation trap. You can configure virtual **Subinterfaces** for additional traps, with static IP addresses in the same network; in non-cloud environments, subinterfaces are available only if the parent interface's address is **Static** (rather than assigned by DHCP).

Configure interfaces in a single-network connection (see Configuring Interfaces in a Single-Network Connection on page 22)

NIS

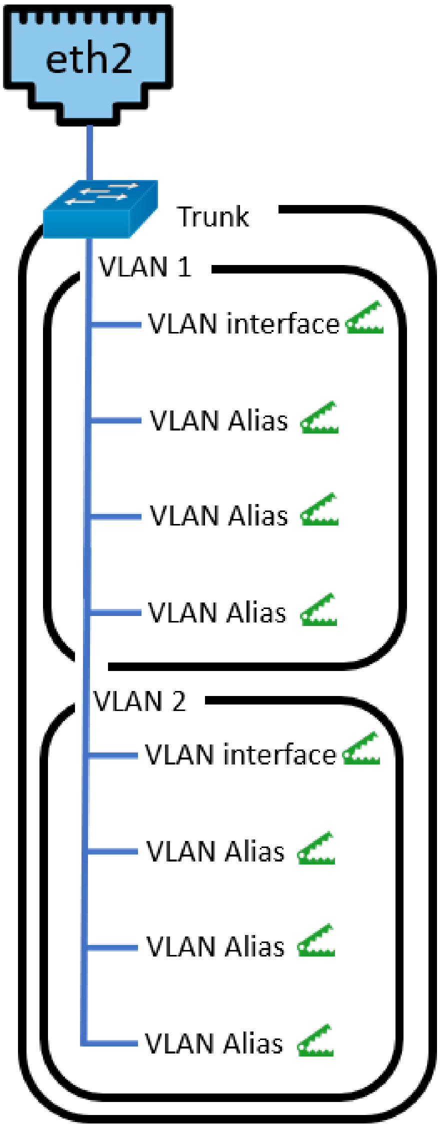
By default, in regular, non-cloud environments, the **eth1** interface is dedicated for NIS. In these environment types, even when NIS is not actually deployed, traps cannot be configured on eth1.

Configuring Interfaces in a Trunk Connection

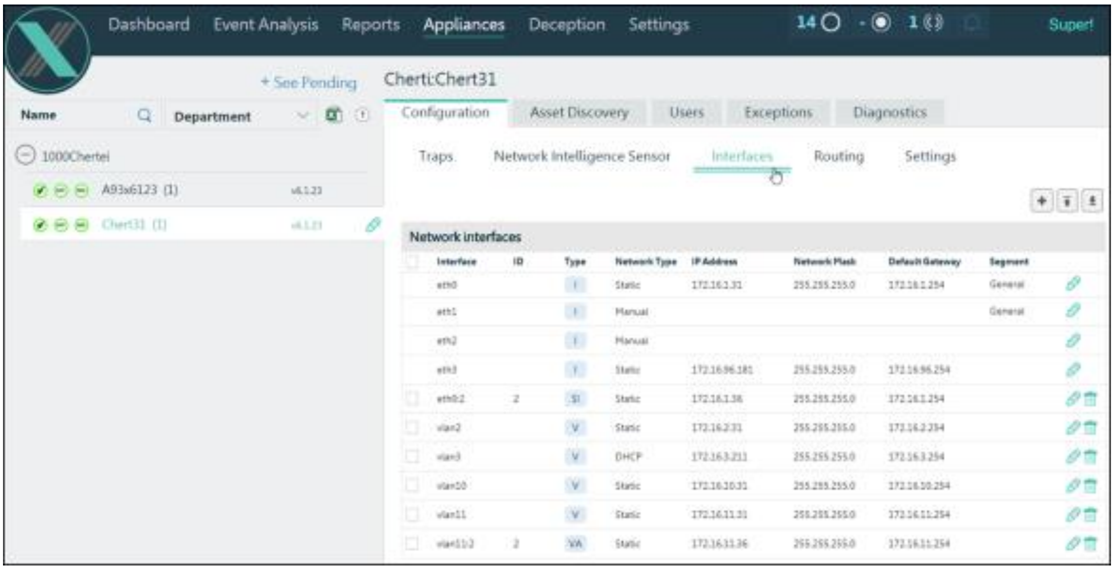
In environments where network switches (virtual or physical) are organizationally managed (as opposed to cloud environments such as AWS and Azure), **eth2** was likely connected to a network trunk, to enable deploying traps to multiple VLANs; **eth3** may also have been connected to another network trunk. In the example diagram, eth2 is connected to a network trunk.

In this case, the parent interface (eth2 / eth3) does not have its own IP address, and cannot be a trap. Instead, for each VLAN in your network you add to the parent interface a virtual **VLAN** child interface, with an IP address (static or DHCP) in the VLAN. This VLAN interface can be configured with a trap.

For additional traps in the same VLAN, add to the VLAN interface grandchild **VLAN Alias** interfaces, with static IP addresses in the VLAN.



Manage interfaces in TSOC > **Appliances** > Appliance > **Configuration** > **Interfaces**:

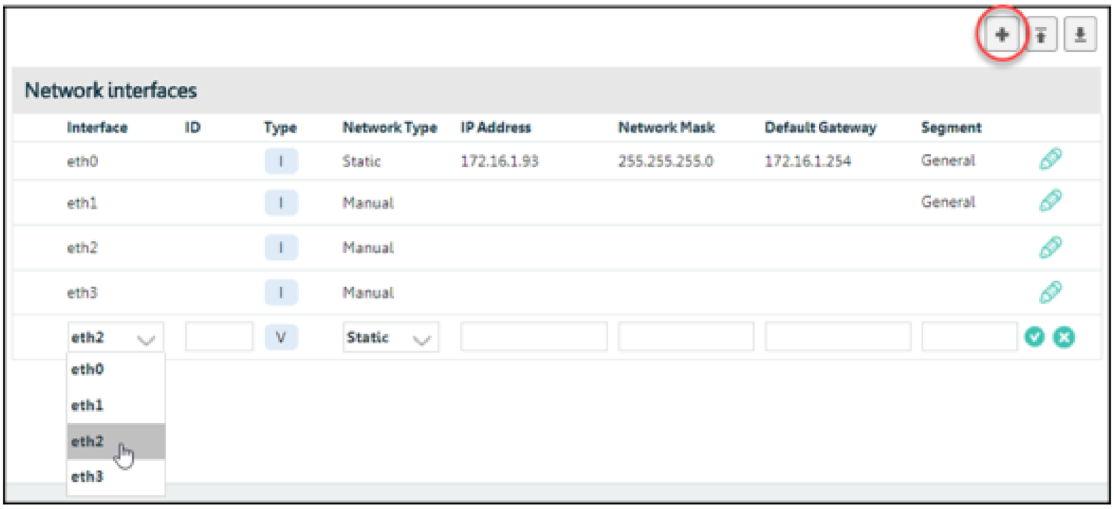


Network interfaces

Interface	ID	Type	Network Type	IP Address	Network Mask	Default Gateway	Segment
eth0		I	Static	172.16.1.31	255.255.255.0	172.16.1.254	General
eth1		I	Manual				General
eth2		I	Manual				
eth3		I	Static	172.16.96.181	255.255.255.0	172.16.96.254	
eth0:2	2	SI	Static	172.16.1.36	255.255.255.0	172.16.1.254	
vlan2		V	Static	172.16.2.31	255.255.255.0	172.16.2.254	
vlan3		V	DHCP	172.16.3.211	255.255.255.0	172.16.3.254	
vlan10		V	Static	172.16.10.31	255.255.255.0	172.16.10.254	
vlan11		V	Static	172.16.11.31	255.255.255.0	172.16.11.254	
vlan11:2	2	VA	Static	172.16.11.36	255.255.255.0	172.16.11.254	

The **Network Type** of a parent interface connected to a network trunk should be **Manual** (meaning its networking properties are managed directly by DeceptionGrid).

To add a child **VLAN** interface, click + and select the parent **Interface**:

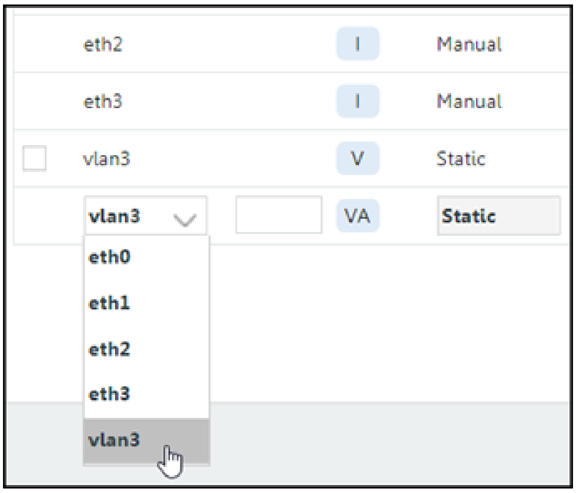


Network interfaces

Interface	ID	Type	Network Type	IP Address	Network Mask	Default Gateway	Segment
eth0		I	Static	172.16.1.93	255.255.255.0	172.16.1.254	General
eth1		I	Manual				General
eth2		I	Manual				
eth3		I	Manual				
eth2		V	Static				

Provide the VLAN **ID**, and configure the virtual VLAN interface's networking properties according to organizational networking requirements. **Segment** is a name for the organizational network area, enabling Attack Visualization by these network segments.

To add a grandchild **VLAN Alias** interface, click + and select the parent **VLAN** interface:



	eth2		I	Manual
	eth3		I	Manual
□	vlan3		V	Static
	vlan3		VA	Static

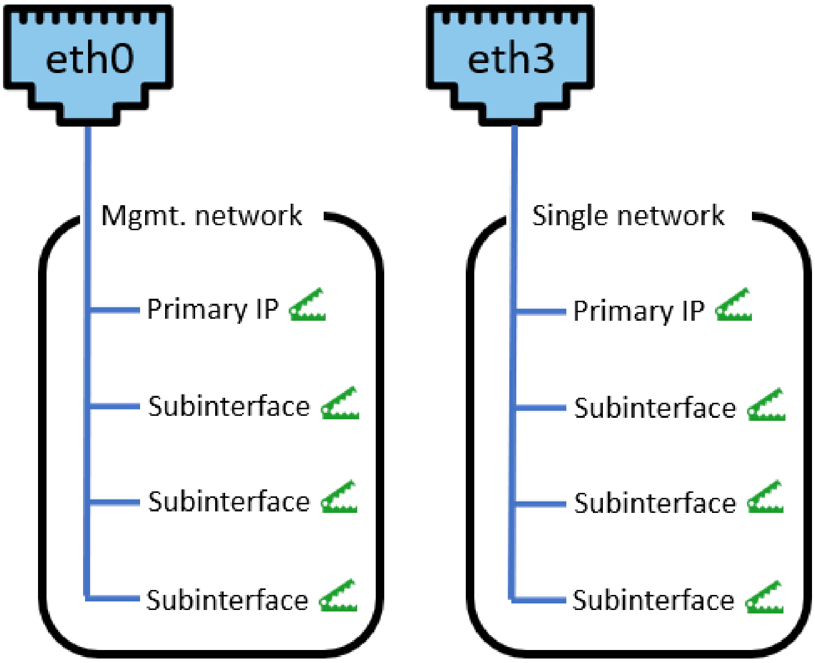
Provide any new **ID** number. **Network Type** is **Static**; configure interface networking accordingly.

Configuring Interfaces in a Single-Network Connection

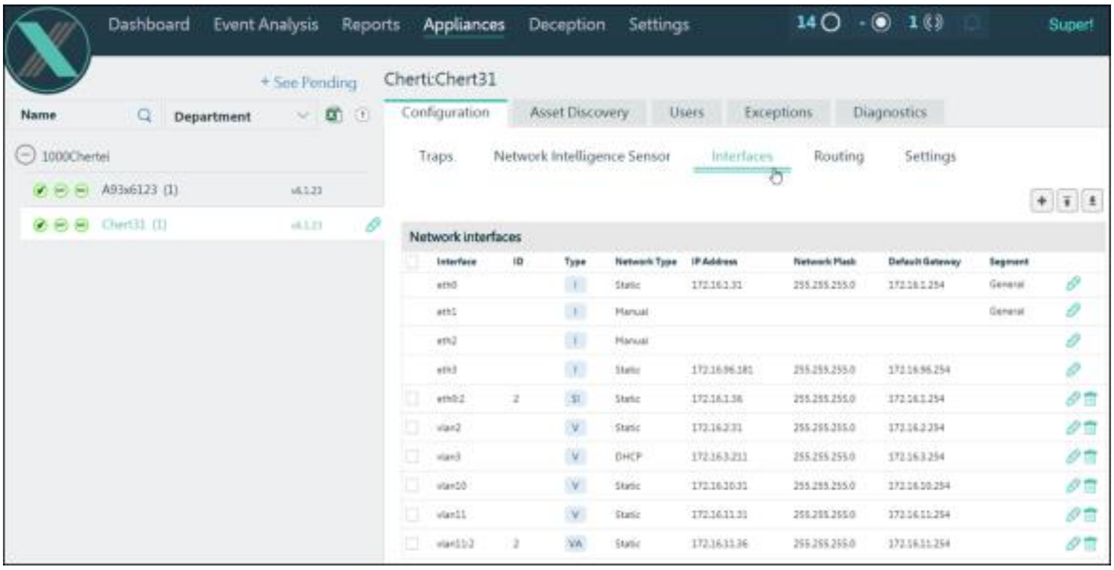
The **eth0** interface is always connected to a single management network, through which the Appliance communicates with TSOC. In addition, some other interfaces may have been connected to single networks rather than to network trunks (in cloud environments like AWS or Azure, this is the only option for additional networks). In the example diagram, eth0 and eth3 are connected to single networks.

An interface connected to a single network has its own IP address, which can be configured with an emulation trap. You can configure virtual **Subinterfaces** for additional traps, with static IP addresses in the same network; in non-cloud environments, subinterfaces are available only if the parent interface's address is **Static** (rather than assigned by DHCP).

An exception is eth1 in cloud environments, which itself cannot be a trap. Only its subinterfaces can be traps.



Manage interfaces in TSOC > **Appliances** > Appliance > **Configuration** > **Interfaces**:

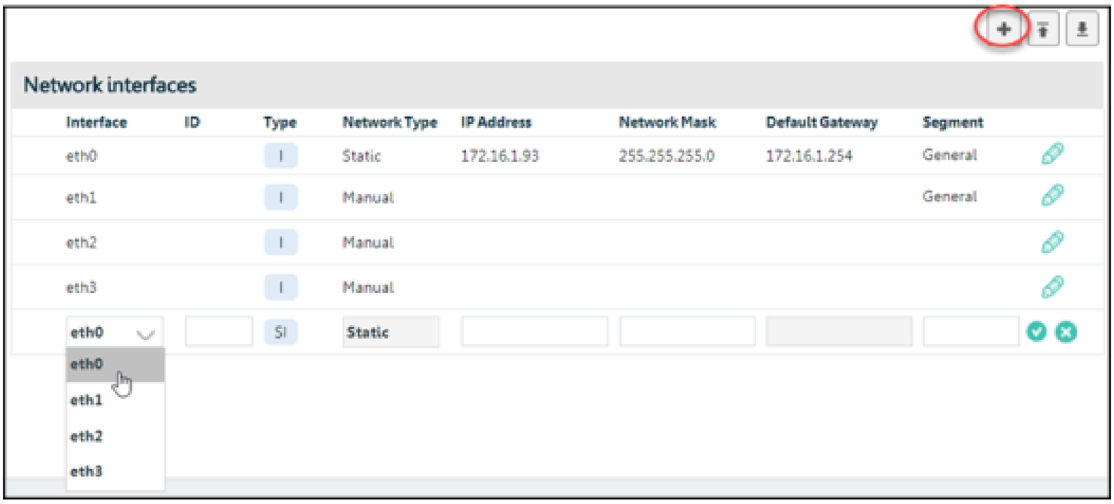


Network interfaces

Interface	ID	Type	Network Type	IP Address	Network Mask	Default Gateway	Segment
eth0		I	Static	172.16.1.31	255.255.255.0	172.16.1.254	General
eth1		I	Manual				General
eth2		I	Manual				
eth3		I	Static	172.16.96.181	255.255.255.0	172.16.96.254	
eth0.2	2	SI	Static	172.16.1.36	255.255.255.0	172.16.1.254	
vlan2		V	Static	172.16.2.31	255.255.255.0	172.16.2.254	
vlan3		V	DHCP	172.16.3.211	255.255.255.0	172.16.3.254	
vlan10		V	Static	172.16.10.31	255.255.255.0	172.16.10.254	
vlan11		V	Static	172.16.11.31	255.255.255.0	172.16.11.254	
vlan11.2	2	VA	Static	172.16.11.36	255.255.255.0	172.16.11.254	

eth0's **Network Type** was determined at DeceptionGrid setup. Any other interface connected to a single network should have a **Network Type** of **Static** to enable subinterfaces; otherwise it can be **DHCP** (but not in cloud environments). **Segment** is a name for the organizational network area, enabling Attack Visualization by these network segments.

To add a subinterface to a single-network static interface, click + and select the parent **Interface**:



Network interfaces

Interface	ID	Type	Network Type	IP Address	Network Mask	Default Gateway	Segment
eth0		I	Static	172.16.1.93	255.255.255.0	172.16.1.254	General
eth1		I	Manual				General
eth2		I	Manual				
eth3		I	Manual				
eth0		SI	Static				

Provide any new **ID** number. **Network Type** is **Static**; configure networking accordingly. **Segment** is a name for the organizational network area, enabling Attack Visualization by these network segments.

Configuring Emulation Traps

Emulation traps (see Emulation Trap Deployment on page 17) are configured on DeceptionGrid Appliance interfaces. Before you can configure the traps, you need to configure appropriate interfaces (see Configuring Network Interfaces for Traps on page 17).

Various emulation types are available (see Available Emulations and Emulated Services below). You can configure them individually (see Configuring Emulation on page 26), or in bulk (see Bulk-Configuring Interfaces and Traps on page 36).

In This Section

Available Emulations and Emulated Services

A wide range of device types is available for emulation. For each such emulation type, relevant services can be emulated. These service ports, when enabled, will be open at the configured trap's IP address, and in most cases the services also respond in an authentic manner as relevant for the device type. Some services (NBNS, CDP) are active, meaning the emulation periodically creates relevant network traffic for trap credibility.

Some of the emulation services have specific, configurable details (see Emulation Service Configuration on page 30).

On Windows emulations, you can also enable **Responder Detector**.

Explain Responder Detector

Responder Detector is a DeceptionGrid detection tool for actively searching for and identifying Responder attacks (commonly found in Windows network environments).

When enabled for a Windows emulation (in the emulation's list of emulated services), Responder Detector periodically sends LLMNR and NBT-NS requests, emulating a situation where a (fake) hostname was not resolvable by DNS and the requestor needs information from its neighboring endpoints. Any response purporting to have knowledge of the fake hostname is identified by DeceptionGrid as a Responder attack, and an appropriate event is recorded.

The interval between requests is random, with a maximum of 15 minutes. For testing purposes, to trigger a request sooner, you can, in the trap list of emulated services, disable and re-enable Responder Detector (make sure to Apply after each action). A first request will be sent within 20 seconds.

You can also create your own emulation type (see Creating a Custom Emulation Type (BYOT) on page 35).

The following device types and associated services are provided for emulation:

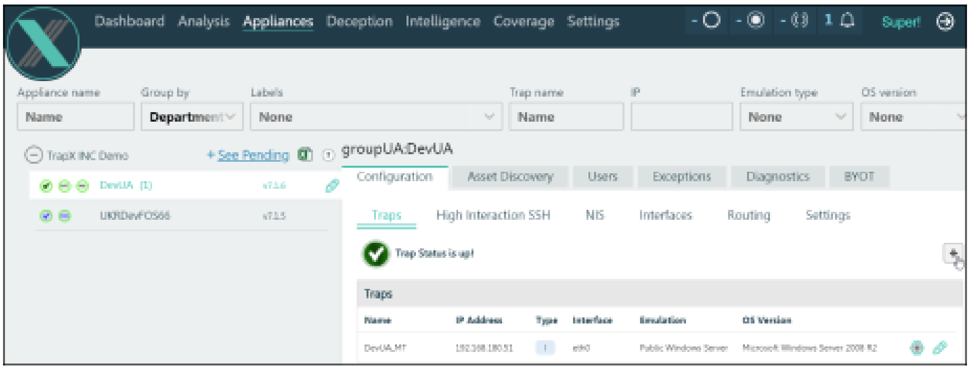
Device category	***Device type and OS version***	***Available services***
Workstations	Windows XP/7/8/10	SMB, NBNS, FTP, RDP, WMI, WinRM, Custom
	Mac OS	SSH, IPP, Bonjour, Custom
Servers	Windows Server 03/SP1 / 08R2 / 12/R2 / 16	SMB, Web, FTP, AD, MS SQL, Oracle, RDP, DNS, NBNS, WMI, WinRM, Custom
	Linux Server CentOS / Red Hat 7 / SUSE 12	SSH, Web, MySQL, Oracle, Custom
	Public Trap, CentOS	Web (generates events only on connections from Deceptive Files token).
Networking	Cisco Catalyst switch 2960/3600, IOS 12.3	Telnet, SSH, Web Managed, SNMP, TFTP, CDP, Custom
	VOIP device, Cisco PBX	SIP, Managed, Custom
	Juniper EX2200, Junos 12.3R9.4	SSH, J-Web, Custom
Industrial	Siemens PLC, S7 300/1200	Web Monitor, Active LLDP, S7, SNMP, Custom
	SCADA device	Modbus, DNP3, Telnet, Web Monitor, FTP Monitor, Custom
	SAP, GUI / Netweaver	SSH, Web Monitor, Custom
	Rockwell PLC, various versions	Web Monitor, CIP_Rockwell, SNMP, Wdbrpc, Custom
Medical	PACS Server, Windows Server 03/SP1/08R2/12/R2	SMB, NBNS, DICOM, WMI, Custom
	CT, Linux 3.7	SSH, DICOM, Custom

Device category	***Device type and OS version***	***Available services***
	MRI, Windows 2012/R2	SMB, NBNS, DICOM, Web, NMF, HTTP API, WMI, Custom
	PACS Viewer, Windows 08R2/12/R2	SMB, NBNS, Web, HTTPAPI, ICE, WMI, Custom
IoT	Point of sale, Win7 Embedded Standard/POSReady	SMB, RDP, NBNS, WMI, Custom
	Philips Smart Light, Busybox 1.19.4 Linux 3.14	Web, 8080, Philips Active Connection, UPNP, Custom
	Lexmark printer, Lexmark embedded	FTP, Finger, HTTP/S, IPP, Printer, TFTP, Telnet, Custom
	Axis network camera, Linux 2.6.19 / 2.6.36	49152, Bonjour, FTP, HTTP/S, RTSP, SNMP, UPNP, Custom
Financial	SWIFT Alliance: Web Platform / Access (SAA) / Gateway (SAG), Windows Server 03/SP1/08R2/2012/R2	Web (on Web Platform), NBNS, SMB, RDP, Custom
	SWIFT Alliance Lite2, Windows XP/7/8/10	WMI, NBNS, SMB, Custom
	ATM, Windows 7	
Remote	VPN Server, Fortinet Fortigate various versions	Web (port 443 only), DeceptiveFileListener (port 80)
	Public Windows Server, Windows Server 08R2 / 12/R2 / 16	SMB, Web (port 443 only), DeceptiveFileListener (port 80)

Configuring Emulation

To configure an emulation trap:

1. For MAC address credibility of emulations in general, enable deceptive MAC addresses for the traps of each virtual VLAN interface (not applicable to traps in single-network connections; VLAN Aliases will share MAC addresses with the parent VLAN):
   a. In TSOC, make sure to have selected **Settings** > **General** > **MAC configuration** > **Dynamically assign MAC addresses**.
   b. Make sure that Appliance infrastructure and network environment allow MAC spoofing. For example:
      - On **VMware**: In the vSwitch **Properties**, make sure that both **Promiscuous Mode** and **Forged Transmits** are set to **Accept**.
      - On **HyperV**: In the Network Adapter **Advanced Features**, make sure that **Enable MAC address spoofing** is selected.
2. In TSOC, go to **Appliances** > Appliance > **Traps**, and do one of the following:
   - To configure a new trap (on an already-configured interface (see Configuring Network Interfaces for Traps on page 17)), click ⊞:



Appliance name: Name
Group by: Department
Labels: None
Trap name: Name
IP:
Emulation type: None
OS version: None

groupUA:DevUA

Trap Status is up!

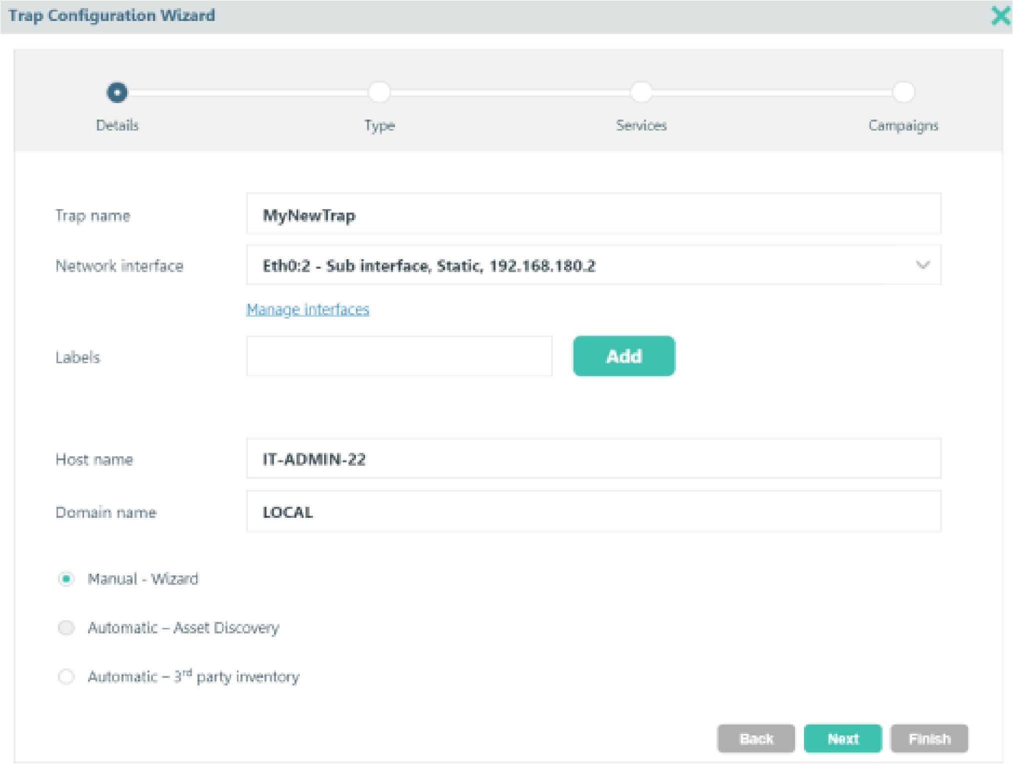
Traps

Name	IP Address	Type	Interface	Emulation	OS Version
DevUA_MT	192.168.180.51	I	eth0	Public Windows Server	Microsoft Windows Server 2008 R2

- To edit an existing trap, in the trap's row click 🖉. To find a specific trap, you can filter the list by **Appliance name** and/or trap attributes (**Trap name**, **Labels**, **IP**, **Emulation type**, and **OS version**).

Note: Some of the trap settings below are not editable for existing traps.

3. **Name** the trap, and select the relevant interface on which to configure the trap:



Trap name: MyNewTrap
Network interface: Eth0:2 - Sub interface, Static, 192.168.180.2
Labels:
Host name: IT-ADMIN-22
Domain name: LOCAL

● Manual - Wizard
○ Automatic – Asset Discovery
○ Automatic – 3rd party inventory

Optionally, type one or more labels and **Add** them. This will enable searching the Appliance page and its traps by traps with specified labels, and filtering the Event Analyzer by events' originating traps with specified labels.

Note: If you leave **Host name** as is, it will subsequently change upon every change to emulation type as appropriate; to prevent automatic changes, manually set the host name. To later revert to automatic host hame, set it to: **default** .

4. Select how the emulation profile should be determined:

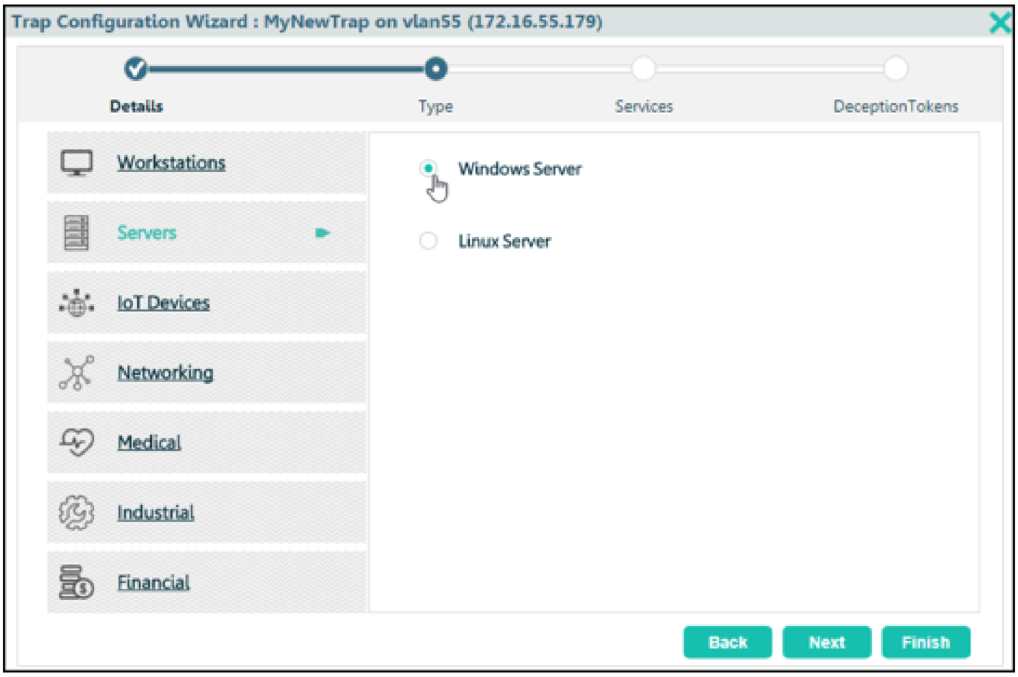
- **Manual**: You'll set emulation details in the following pages of the trap configuration wizard.

- **Automatic - Asset Discovery**: Available if there are Asset Discovery results (see Asset Discovery on page 39). The trap will be automatically set, upon future changes to discovery results, to emulate the most common device type in its network.
- **Automatic - 3rd party inventory** (available if Asset Discovery is disabled): Emulation details will be automatically set, upon future asset inventory updates, to emulate the most common device type according to inventory.

If you selected one of the **Automatic** options, click **Finish** and you're done.

Otherwise, click **Next** and continue:

5. In the **Type** page, select emulation category and emulation type:

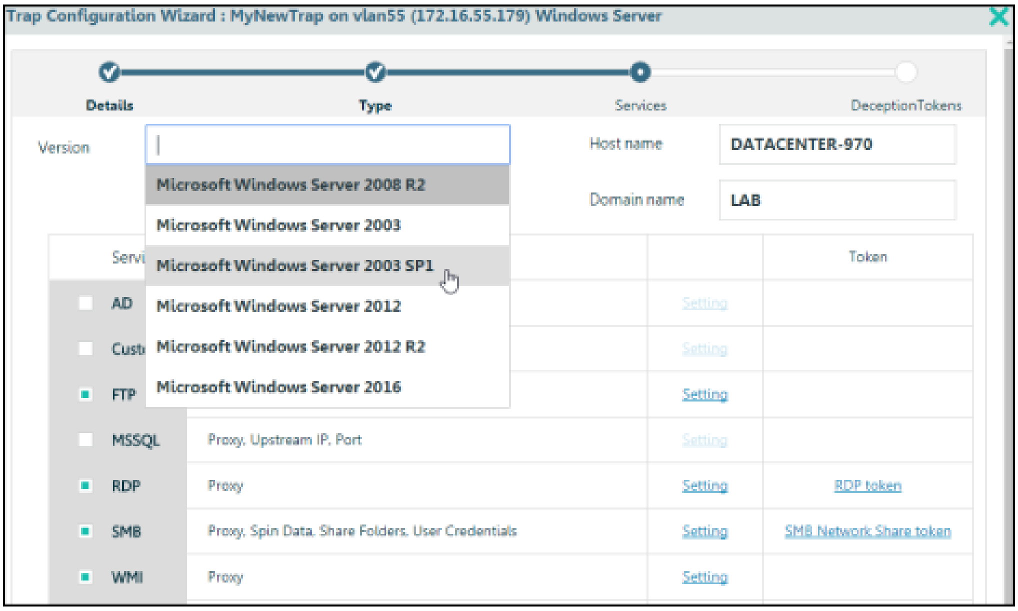


Trap Configuration Wizard : MyNewTrap on vlan55 (172.16.55.179)

Servers: ● Windows Server

Click **Next**.

6. In the **Services** page, select an emulated OS **Version**, and optionally change the emulation's **Host name** and/or (for Windows emulations) **Domain name**:



Trap Configuration Wizard : MyNewTrap on vlan55 (172.16.55.179) Windows Server

Version: |

- Microsoft Windows Server 2008 R2
- Microsoft Windows Server 2003
- Microsoft Windows Server 2003 SP1
- Microsoft Windows Server 2012
- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2016

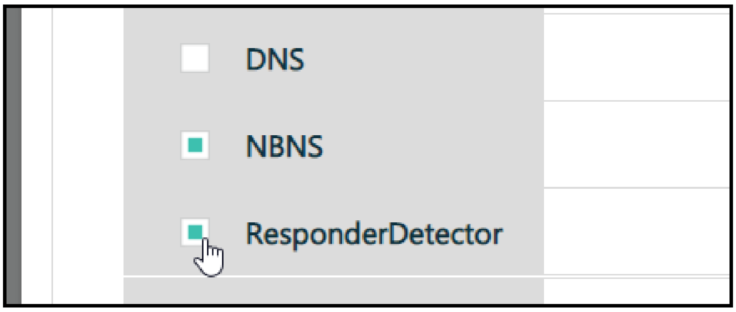
Host name: DATACENTER-970

Domain name: LAB

	Servi			Token
☐	AD		Setting	
☐	Cust		Setting	
☑	FTP		Setting	
☐	MSSQL	Proxy, Upstream IP, Port	Setting	
☑	RDP	Proxy	Setting	RDP token
☑	SMB	Proxy, Spin Data, Share Folders, User Credentials	Setting	SMB Network Share token
☑	WMI	Proxy	Setting	

Note: The domain name should be the real domain used in the relevant network, not only for realism and credibility to attackers as with all other emulation settings but also because if an SMB service is here enabled and the Appliance is enabled for SMB Signing (see the *DeceptionGrid Administration Guide*), this domain name will be used for responses to SMB signing. If the domain name is incorrect, signed SMB connections to the trap may fail.

7. From the OS-appropriate list, select **Services** to be emulated. Optionally, clicking **Setting** to configure any service's details (see Emulation Service Configuration on page 30).
8. In addition to emulated services, for Windows emulations you can here enable Responder Detector (see Available Emulations and Emulated Services on page 24):



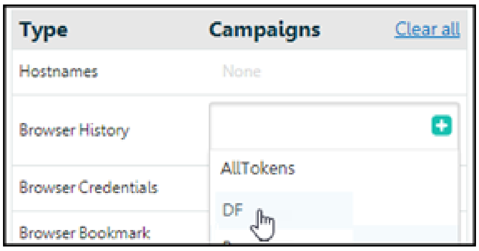
☐ DNS

☑ NBNS

☑ ResponderDetector

9. Optionally, click to change any **Token** details (see Configuring Deception Tokens on page 50).

   Click **Next**.
10. In the **DeceptionTokens** page, if campaigns have already been configured (see Configuring Token Campaigns on page 48), to assign the trap's tokens to campaigns, hover over each token's row, click ✎ , ⊞ , and select campaigns:



Type	Campaigns
Hostnames	None
Browser History	
Browser Credentials	
Browser Bookmark	

Optionally, you can **Assign all tokens** to a campaign.

11. Click **Finish**.

Emulation Service Configuration

In the trap configuration wizard (see Configuring Emulation on page 26), each available service can be enabled or disabled for the individual trap. Additionally, some of the emulation services have specific, configurable details for trap customization, as below.

Note: Some configuration options may appear only for relevant emulation types.

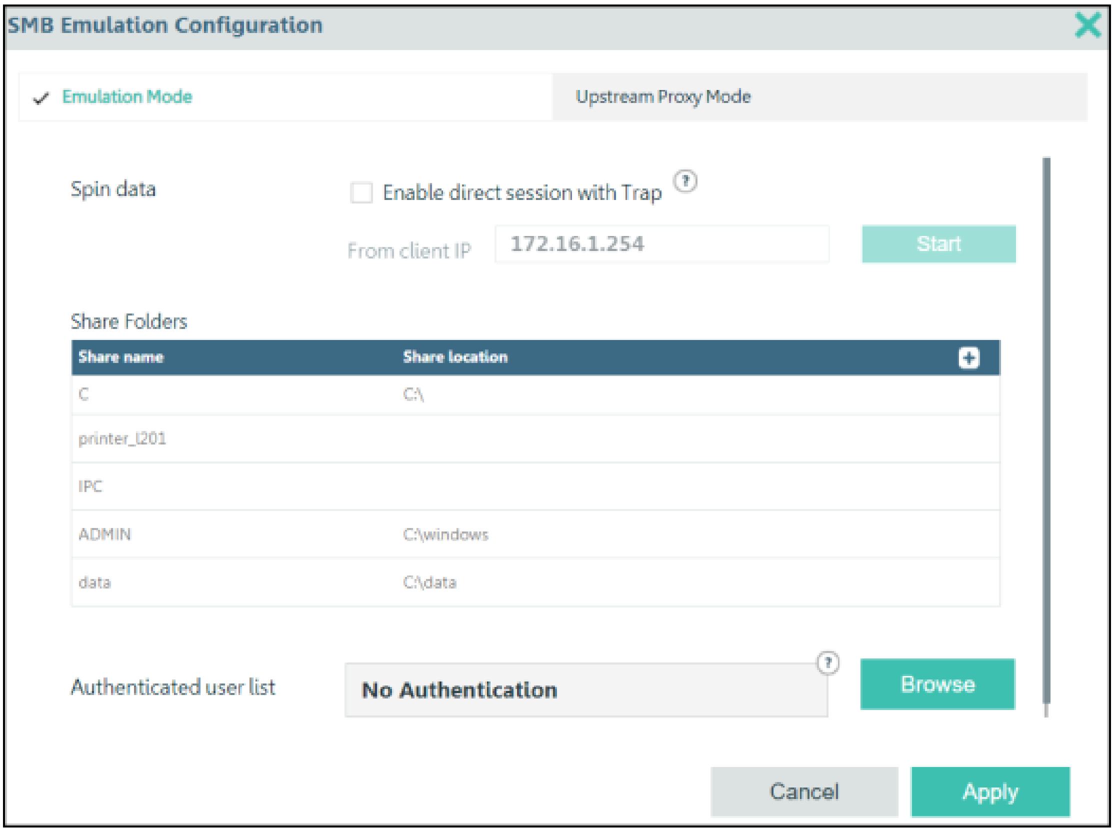
In addition, some services may be directly configurable.

In addition to service configuration from TSOC, you can connect to a Cisco network switch emulation via its SSH or Telnet service and configure its Running Configuration. For example, you could configure a fake SNMP community, fake network monitor server IP address, and fake credentials, and then configure these values in your organizational monitoring system watchlist to intercept their use in other systems.

The following emulation service configuration options are available in TSOC. For some services, for increased service response realism (and, in some cases, fuller event details) you can set the service to **Proxy Mode** for the emulated service to connect to a real service on a deployed Full OS trap and convey its real responses. In this case, you'll select the specific Full OS trap and its service (in case the Full OS trap has multiple services of the same type).

SMB

Either configure **Proxy Mode** to a full OS trap (see above), or, in regular **Emulation Mode**, configure the following for the emulation:



Spin data

□ Enable direct session with Trap

From client IP: 172.16.1.254

Share Folders

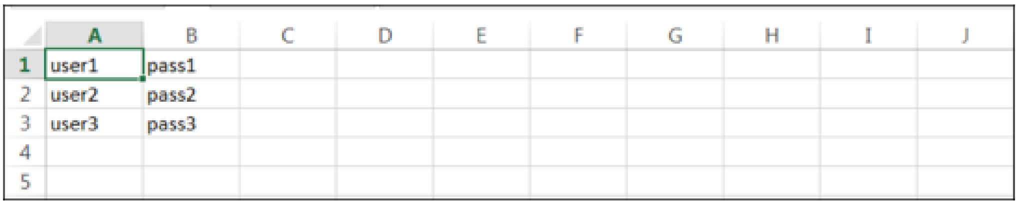
Share name	Share location
C	C:\
printer_l201	
IPC	
ADMIN	C:\windows
data	C:\data

Authenticated user list: No Authentication

- **Spin data**: You can open a direct FTP session (requires port 9445 to be open from your workstation to the Appliance) and upload realistic files that attackers will see upon connecting to the trap via SMB. Select **Enable direct session with trap** and click **Start**.
- **Share Folders**: Some shared folder emulations are provided by default, and you can add (⊞) additional ones. To edit names and locations of added folders, double-click them; to remove predefined or added folders, hover and click 🗑. The share name that points to **C:\data** will be distributed with the SMB Network Share Deception Token.
- **Authenticated user list**: In most cases, response to signed SMB should have been enabled for the Appliance (see the *DeceptionGrid Administration Guide*), in which case do not provide a user list here. Otherwise, upload a list of credential sets to authorize for connections to the emulation, making the trap appear more realistic to attackers and causing them to divulge the credentials they have. Best practice is to configure a few traps with easy-to-guess username - password combinations, to facilitate attacks to the traps.

Note: If response to signed SMB is enabled, providing an authenticated user list here will effectively disable response to signed SMB. The authenticated user list should be provided only if response to signed SMB is not enabled.

The uploaded list should be a two-column CSV file (can be created in Excel) of a maximum of 100 username-password sets. For example:



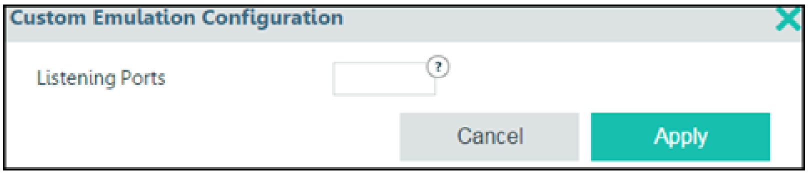
	A	B	C	D	E	F	G	H	I	J
1	user1	pass1								
2	user2	pass2								
3	user3	pass3								
4										
5										

You can use * to accept any username; * cannot be used for password.

Browse to upload the CSV file. If you leave the setting as **No Authentication**, all credentials will be accepted.

Custom

Specify ports on which the trap should listen. Attempting to connect to this port will trigger an event.



Custom Emulation Configuration

Listening Ports

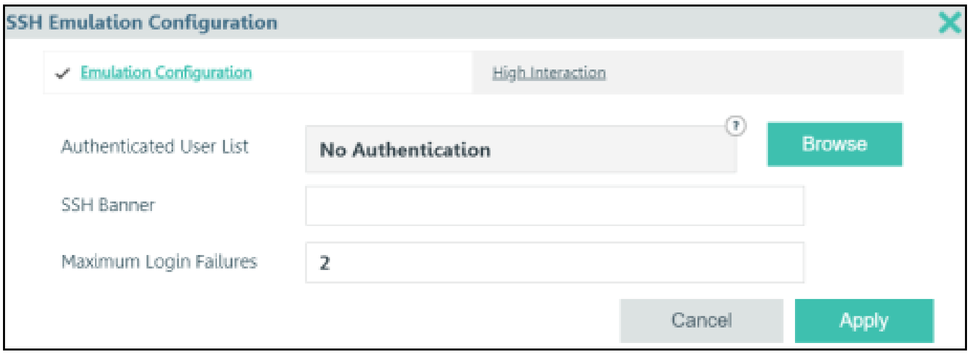
SSH

Do one of the following:

- Connect the SSH service to the **High Interaction** full Linux OS (see High-Interaction Linux Services on page 37): Provides optimal realism and deception.
- Set **Emulation** Mode: Enables open, unauthenticated access.

  For emulation mode, you can:

  - Customize the login **Banner** sent during login
  - Upload an **Authenticated user list** as for SMB (above), and set the number of failed login attempts after which an attacker will be authorized even without valid credentials:



SSH Emulation Configuration

✓ Emulation Configuration | High Interaction

Authenticated User List: No Authentication

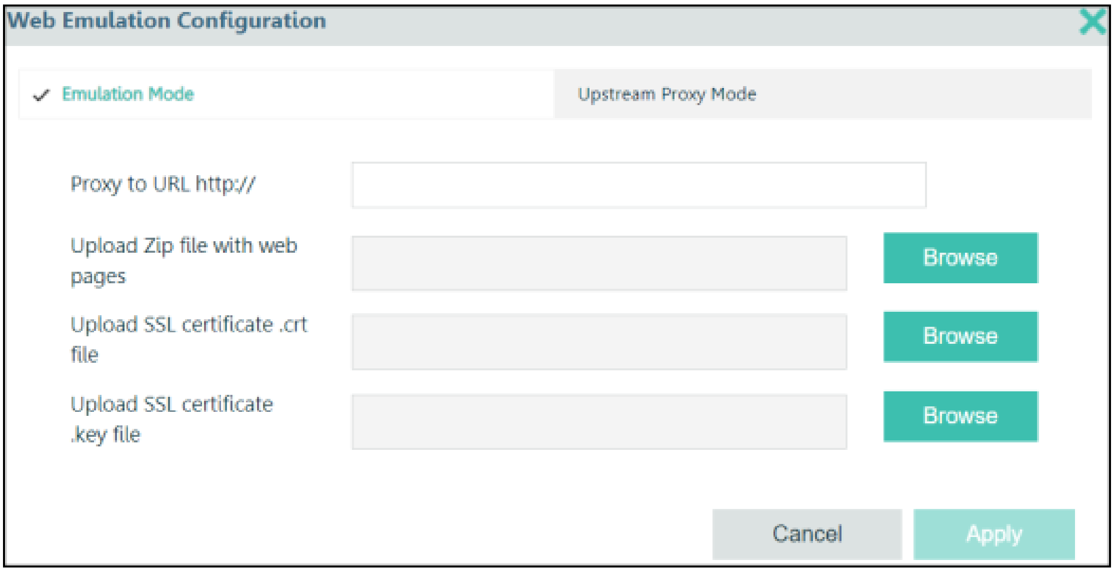
SSH Banner:

Maximum Login Failures: 2

Note: If the attacker is using OpenSSH, the client software will not allow more than three failed attempts. It is therefore not recommended in most cases to set the number of maximum login failures to more than 2.

Web (except for Linux Server web)

Either configure **Proxy Mode** to a full OS trap (see above), or, in regular **Emulation Mode**, configure the following for the emulation:



Web Emulation Configuration

✓ Emulation Mode | Upstream Proxy Mode

Proxy to URL http://

Upload Zip file with web pages

Upload SSL certificate .crt file

Upload SSL certificate .key file

Note: Proxy and web page upload are available only in relevant emulation types.

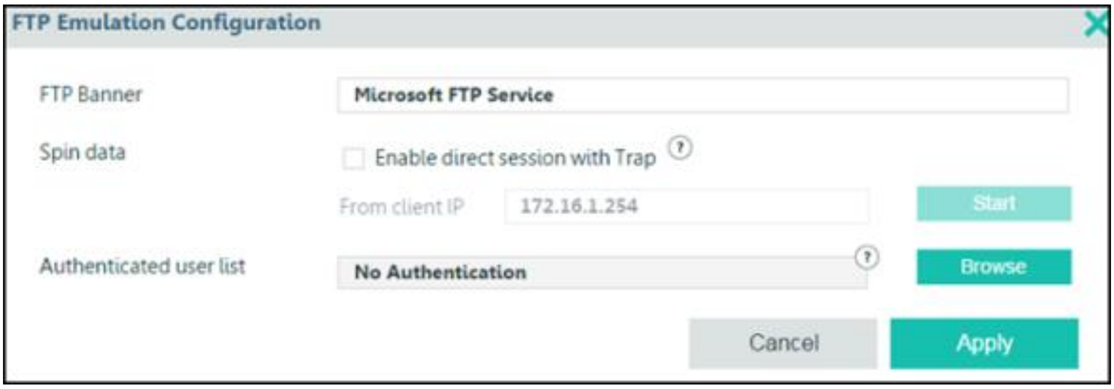
- For actual web pages to serve to web connections, provide one of:
  - **Proxy to URL**: The URL of an actual organizational website. Upon an attacker's connecting to the trap over HTTP/S, that site will be served.
  - **Upload Zip file with web pages**: Actual web pages to be served.
- For HTTPS connections, provide **SSL certificate** files (**.crt** and **.key**).

Web (Linux Server)

In Linux Server emulations, the Web service has, rather than Proxy mode as above, the option to connect to a **High-Interaction** real web service serving typical web content (emulating a Jenkins build server). The web service is hosted by full Linux OS, which must be enabled (see High-Interaction Linux Services on page 37).

FTP / FTP Monitor

The FTP emulation has several configurable options:



FTP Emulation Configuration

FTP Banner: Microsoft FTP Service

Spin data: □ Enable direct session with Trap

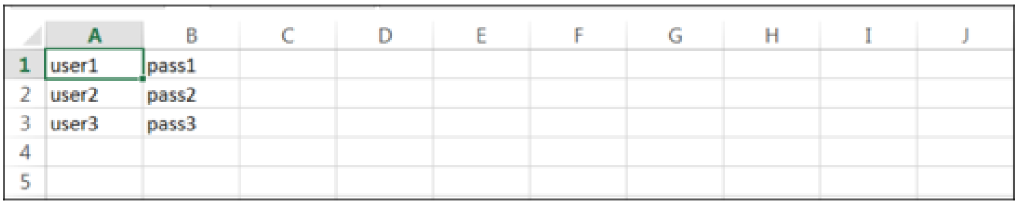
From client IP: 172.16.1.254

Authenticated user list: No Authentication

- **FTP Banner**: Customize the login **Banner** sent during login

- **Spin data**: You can open a direct FTP session (requires port 9021 to be open from your workstation to the Appliance) and upload realistic files that attackers will see upon connecting to the trap. Select **Enable direct session with trap** and click **Start**.
- **Authenticated user list**: Upload a list of credential sets to authorize for connections to the emulation, making the trap appear more realistic to attackers and causing them to divulge the credentials they have. Best practice is to configure a few traps with easy-to-guess username - password combinations, to facilitate attacks to the traps.

  The uploaded list should be a two-column CSV file (can be created in Excel) of a maximum of 100 username-password sets. For example:

	A	B	C	D	E	F	G	H	I	J
1	user1	pass1								
2	user2	pass2								
3	user3	pass3								
4										
5										

  You can use ***** to accept any username; * cannot be used for password.

  Browse to upload the CSV file. If you leave the setting as **No Authentication**, all credentials will be accepted.

AD

Provide the IP address of an actual Active Directory domain controller (the real organizational Active Directory, or one with fake content). Upon an attacker's connecting to the trap and sending LDAP queries, the data from the domain controller will be served, lending authenticity to the emulation.

MSSQL

Either configure **Proxy Mode** to a full OS trap (see above), or, in regular **Emulation Mode**, provide the IP address and port of an actual SQL Server. Upon an attacker's connecting to the trap and sending SQL queries, data from the SQL Server will be served, lending authenticity to the emulation.

RDP

Either configure **Proxy Mode** to a full OS trap (see above), or, in regular **Emulation Mode**, ,leave as is.

WMI

Either configure **Proxy Mode** to a full OS trap (see above), or, in regular **Emulation Mode**, leave as is.

WinRM

Either configure **Proxy Mode** to a full OS trap (see above), or, in regular **Emulation Mode**, leave as is.

Web Managed / Managed / Web Monitor

Provide the URL of an actual organizational device. Upon an attacker's connecting to the trap over HTTP, the device UI will be served, lending authenticity to the emulation.

Telnet / J-Web / WebManaged / Axis HTTP/S

Set the number of failed login attemps after which an attacker will be authorized even without valid credentials.

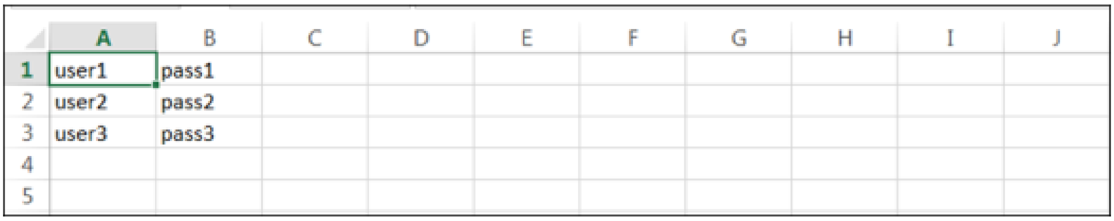
Oracle

Either configure **Proxy Mode** to a full OS trap (see above), or, in regular **Emulation Mode**, provide the IP address and port of an actual Oracle service. Upon an attacker's connecting to the trap and sending Oracle queries, data from the real Oracle will be served, lending authenticity to the emulation.

MySQL

Authenticated user list: Upload a list of credential sets to authorize for connections to the emulation, making the trap appear more realistic to attackers and causing them to divulge the credentials they have. Best practice is to configure a few traps with easy-to-guess username - password combinations, to facilitate attacks to the traps.

The uploaded list should be a two-column CSV file (can be created in Excel) of a maximum of 100 username-password sets. For example:



	A	B	C	D	E	F	G	H	I	J
1	user1	pass1								
2	user2	pass2								
3	user3	pass3								
4										
5										

You can use ***** to accept any username; * cannot be used for password.

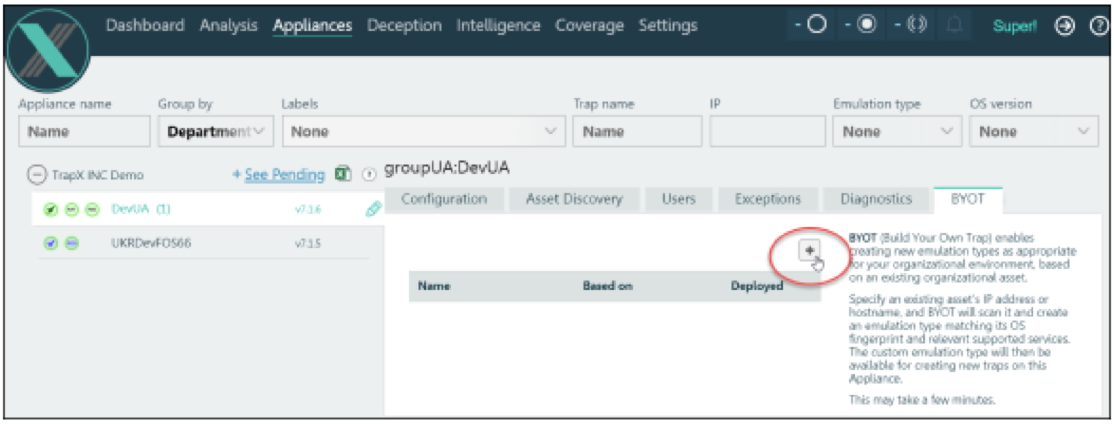
Browse to upload the CSV file. If you leave the setting as **No Authentication**, all credentials will be accepted.

Creating a Custom Emulation Type (BYOT)

You can create new emulation types as appropriate for your organizational environment, based on existing organizational assets.

Specify an existing asset's IP address or hostname, and BYOT (Bring Your Own Trap) scans it and creates an emulation type matching its OS fingerprint and relevant supported services. The custom emulation type will then be available for creating new traps, on a specified Appliance, from the trap creation wizard (Custom category).

To create an emulation type, in TSOC go to **Appliances** > Appliance > BYOT and click +:



BYOT (Build Your Own Trap) enables creating new emulation types as appropriate for your organizational environment, based on an existing organizational asset.

Specify an existing asset's IP address or hostname, and BYOT will scan it and create an emulation type matching its OS fingerprint and relevant supported services. The custom emulation type will then be available for creating new traps on this Appliance.

This may take a few minutes.

When complete, the new emulation type is listed here, and available for creating new traps on this Appliance.

Bulk-Configuring Interfaces and Traps

Instead of adding virtual interfaces to an Appliance and configuring their traps one at a time, you can provide TSOC with a list of interfaces and their trap details, and TSOC will add the virtual interfaces and configure them accordingly.

In the list, you can specify for any of the traps to be created that its emulation details be copied from an existing trap. These details include emulation type, OS version, service configuration including spin data, and network segment.

If Asset Discovery results are available (see Asset Discovery on page 39), you can have traps' emulation set automatically according to those results (now, and automatically upon future changes to discovery results). To achieve this, in the import CSV leave emulation type and emulation version blank.

You can download a list of all of an Appliance’s configured virtual interfaces. You can then use this list as a template for creating new interfaces, or to migrate interfaces between Appliances.

Importing interfaces cannot be used to change the details of existing interfaces.

The interface list is a CSV file with a heading row (ignored by TSOC) and a row for each interface to be created (maximum 100 interfaces). It includes the following columns:

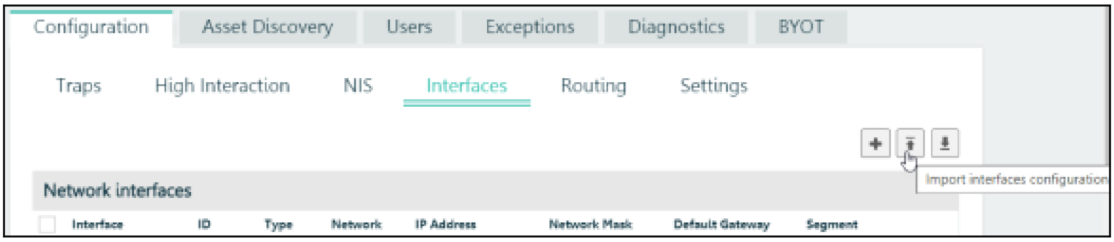
- **Parent interface**: The physical Appliance interface on which a sub interface or VLAN interface is to be configured, or the VLAN interface on which a VLAN Alias is to be configured.
- **Interface**, **Type**, **Network type**, **IP address**, **Network mask**, and **Default gateway**: As in the columns of these names in the TSOC Appliance **Configuration** > **Interfaces** table.
- **Trap**: Whether this interface is a Trap: **yes** / **no** .
- **Trap name**: As in the column of the same name in the TSOC Appliance **Configuration** > **Interfaces** table.
- **Emulation type**, **Emulation version**, **Host name**, and **Domain name**: As in the Appliance **Configuration** > **Traps** > Trap configuration wizard.

For the emulation to be set automatically according to Asset Discovery results, leave the emulation type and emulation version blank.

- **Segment**: As in the column of the same name in the TSOC Appliance **Configuration** > **Interfaces** table.
- **Copy from**: An existing trap whose emulation details and network segment should be copied to this trap.

Interfaces with traps, if not specified to be copied from existing traps, are configured with the default services for the emulation type; you can subsequently manually change service configuration. If copying is specified but the **Emulation type** is also specified in the CSV list – the list is used, and no emulation details are copied. If copying is specified and either of **Emulation version** and/or **Segment** is specified in the CSV list – the list is used.

To provide or download an interface list, go to **Appliances** > Appliance > **Configuration** > **Interfaces** and click ⤒ (upload) or ⤓ (download) respectively:



High-Interaction Linux Services

For ultimate realism and deception, you can connect the SSH service on relevant emulation traps and the Web service on Linux Server emulations to an Appliance-based sandboxed full Linux OS. With SSH, attackers receive real, high-interaction responses to all relevant commands, and the environment includes all real OS components such as file system and network connections. With Linux Server Web, attackers receive typical web content (emulating a Jenkins build server).

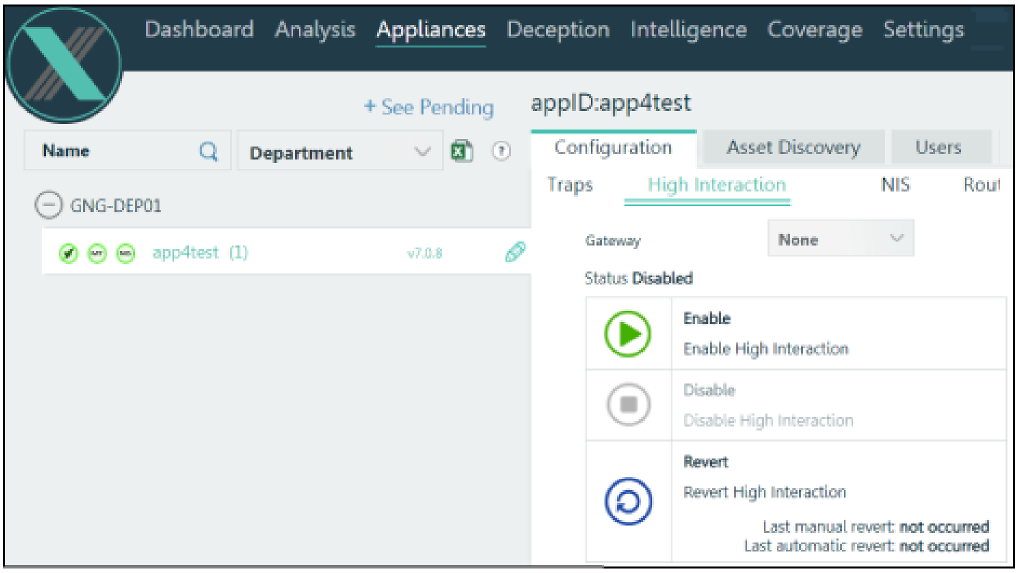
SSH connections to traps connected to the high-interaction full Linux OS are authorized according to a configurable credential list. Username **root** is always authorized, with password **root** or as overridden by the list. If, on the other hand, you want to provide open, non-authenticated access to some traps, they should use the **emulation** mode SSH service rather than the high-interaction full Linux OS.

You can configure the full OS's outbound network access by setting its gateway. The gateway can be **None** (no network access; default), or an Appliance interface (real or virtual) to limit outbound access to that interface's network. The gateway can be changed only while the full OS is disabled. Outbound network access from the Full Linux OS should conform to your organization’s security policy; please contact TrapX support to enable outbound network access.

If the full Linux OS or its own SSH service goes down, the full Linux OS is automatically reverted to its original state. You can also manually revert, to remove all changes made by attackers.

To enable and configure high-interaction SSH on an Appliance:

1. In TSOC go to **Appliances** > Appliance > **Configuration** > **High Interaction**:



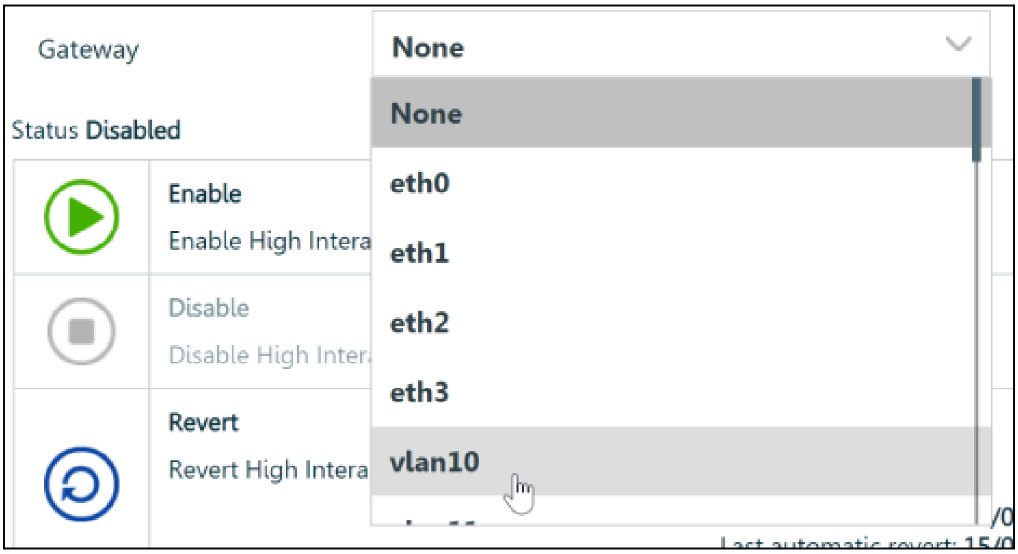
applD:app4test

Gateway None

Status Disabled

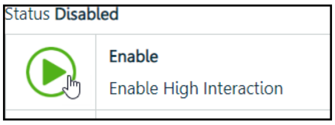
Last manual revert: not occurred
Last automatic revert: not occurred

2. By **gateway**, set the full OS's outbound network access:

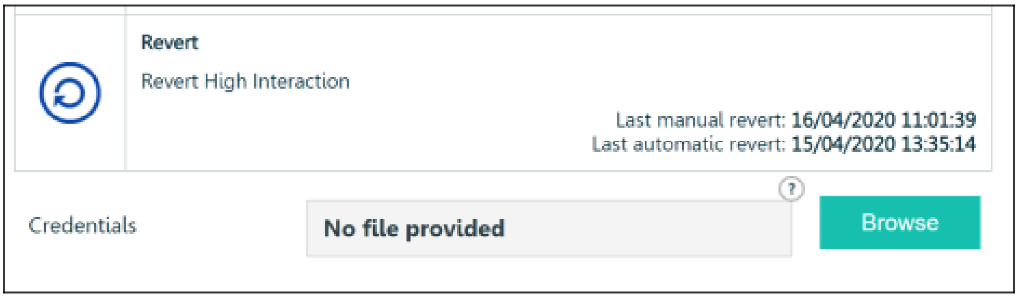


The gateway can be **None** (no network access; default), or an Appliance interface (real or virtual) to limit outbound access to that interface's network. The gateway can be changed only while the full OS is disabled. Outbound network access from the Full Linux OS should conform to your organization’s security policy; please contact TrapX support to enable outbound network access.

3. Click by **Enable**:



4. Once enabled, you can provide a **Credentials** list to be authorized for SSH connections to connected traps:



Revert

Revert High Interaction

Last manual revert: 16/04/2020 11:01:39
Last automatic revert: 15/04/2020 13:35:14

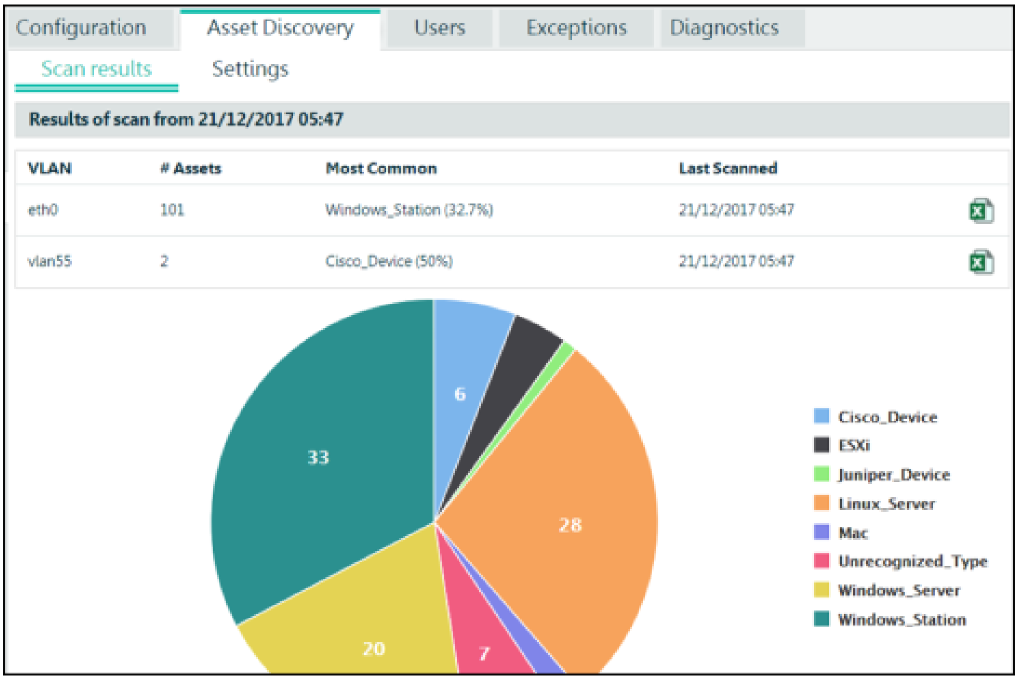
Credentials: No file provided

Create and upload a CSV file of authorized credentials, with each row in format: **user**, **password** (note that username **root** is always authorized, with password **root** or as overridden by the list).

Asset Discovery

With Asset Discovery, DeceptionGrid Appliances can scan their networks and identify organizational devices. The results can be used in several ways:

- **Visibility**: TSOC displays per-VLAN device type distribution:



Results of scan from 21/12/2017 05:47

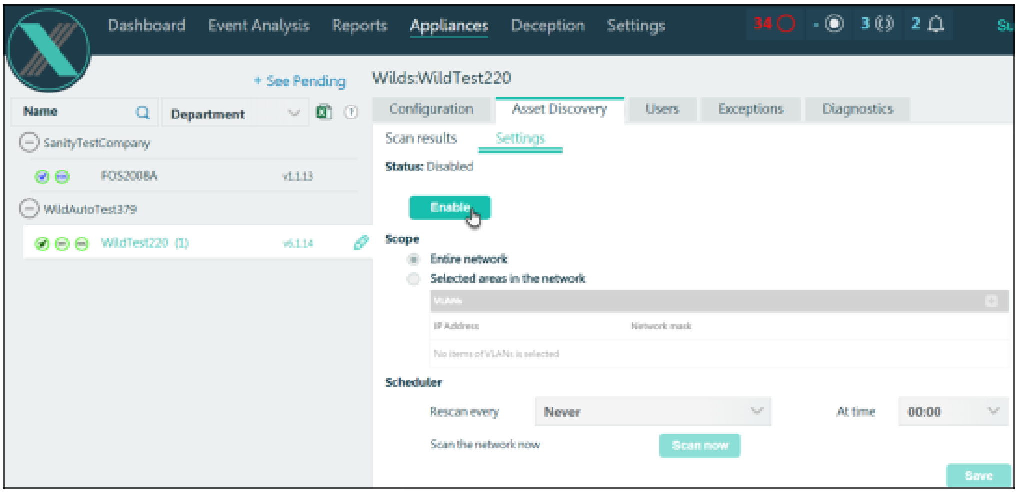
VLAN	# Assets	Most Common	Last Scanned
eth0	101	Windows_Station (32.7%)	21/12/2017 05:47
vlan55	2	Cisco_Device (50%)	21/12/2017 05:47

You can also download the data in CSV format, by clicking .

- **Automatic emulation**: You can configure any emulation trap (see Configuring Emulation on page 26) to automatically emulate the most common device type in its network. As discovery results change, trap configuration is automatically updated.
- **Deception token target selection**: When configuring token distribution from TSOC to organizational endpoints (see TSOC Distribution on page 65), you can select targets from among discovered endpoints.

To activate Asset Discovery:

1. In TSOC go to **Appliances** > Appliance > **Asset Discovery** > **Settings**. **Enable** the feature and set the scan's **Scope**:



Status: Disabled

Scope

● Entire network

○ Selected areas in the network

No items of VLANs is selected

Scheduler

Rescan every: Never   At time: 00:00

Scan the network now

2. Click **Scan now**, and/or to keep results up-to-date schedule **Rescan**s.

   Make sure to **Save**.

Once a scan is complete, to view latest results go to the **Scan results** tab.

Full OS Trap Deployment

In This Section

Full OS Trap Overview

In This Section

Introduction to Full OS Trap

Full OS traps provide a high level of realism and full attack monitoring, by installing the TrapX Full OS Agent on a full (virtual) computer. The host computer can be configured with any software, data, and settings, and the agent will monitor and record not only inbound connections but also outbound activity (for example, if an attacker attempts to connect from the Full OS trap to another endpoint or to the internet).

In addition, you can use a Full OS trap to transparently provide a real service to respond to attackers of emulation traps, and full monitoring of those attacks. This is achieved by proxying emulation traps' services to a Full OS trap. For example, you can have multiple emulation traps proxy Remote Desktop (RDP) sessions to a full Windows Server trap. Attackers who connect to any of those targets will be provided with full real desktop experiences, and their activity will be fully recorded by the Full OS trap.

Optionally, Full OS includes CryptoTrap, a network share which Full OS automatically and continuously populates with document and media files to slow down and deceive detected ransomware attacks. A deception token for this share is managed in TSOC. With CryptoTrap, Full OS also intelligently detects ransomware behavior anywhere in the file system and accordingly records an enriched event specifically identified as Ransomware.

Before deploying traps, plan your deployment (see Planning Your DeceptionGrid Deployment on page 6).

How it Works

Full OS Trap is provided as a software agent to be installed on a full virtual computer with a full operating system. You set up and configure the computer as typical for a relevant organizational network, including any installed software, data, and configuration as appropriate. You then install the Full OS Trap agent on the computer to monitor and record remote user activity.

Recorded activity is collected, analyzed, and displayed, along with activity from all other trap types, in the TrapX Security Operation Console (TSOC).

In case of a Full OS trap host being infected, you can revert it to a pre-attack snapshot. An initial baseline snapshot is created automatically upon initializing the trap, and from TSOC you can set subsequent baseline snapshots and revert as needed.

Just as for emulation traps, you can configure deception tokens (see Deception Token Deployment on page 45) to lure attackers to Full OS trap services.

To remain hidden from attackers, the Full OS trap agent is obfuscated. Its service name, descriptions, and location are according to a profile selected at installation. Defense mechanisms prevent attackers from changing agent files and configuration.

Several defense mechanisms protect the agent itself from attack. In some cases, upon an attack the trap computer may be automatically reverted to the snapshot.

Supported Services

A Full OS trap can monitor, identify and report on remote user activity via any of the following:

- WMI
- Remote Desktop (RDP)
- SMB
- SQL Server (MSSQL) 2014 or above
- Web (HTTP to IIS; HTTPS not currently supported)
- Windows Remote Management (WinRM; partial activity identification)
- Active Directory

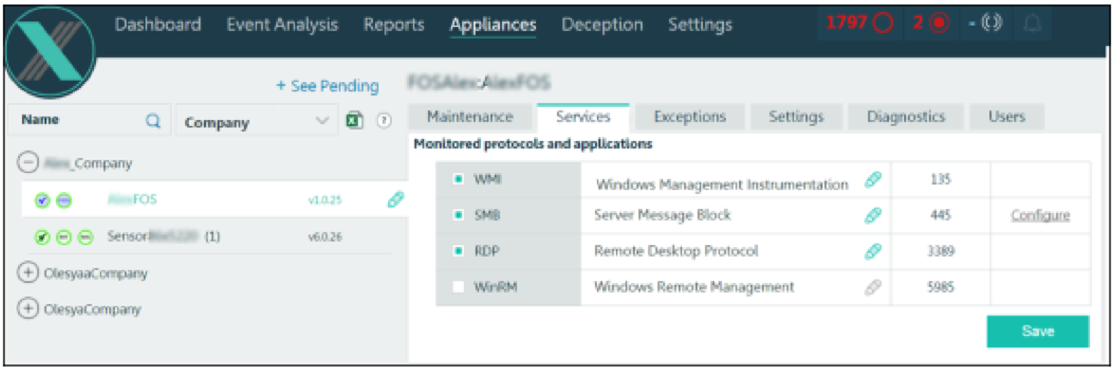
To enable monitoring Windows services listed above, the service must be active on the Full OS trap host computer, and selected in TSOC for the trap (see Configuring Full OS Trap Monitoring below).

Deploying Full OS Trap

In This Section

Configuring Full OS Trap Monitoring

Once a Full OS trap is set up as in the *DeceptionGrid Administration Guide*, you can select which of supported services (see Full OS Trap Overview on page 41) will be monitored for inbound events. In the TSOC **Appliances** page select the trap, and in its **Services** tab select relevant services:



[redacted]

Maintenance | Services | Exceptions | Settings | Diagnostics | Users

[redacted]_Company

- [redacted]FOS v1.0.25
- Sensor[redacted] (1) v6.0.26

OlesyaaCompany

OlesyaCompany

Monitored protocols and applications

■ WMI	Windows Management Instrumentation	135	
■ SMB	Server Message Block	445	Configure
■ RDP	Remote Desktop Protocol	3389	
□ WinRM	Windows Remote Management	5985	

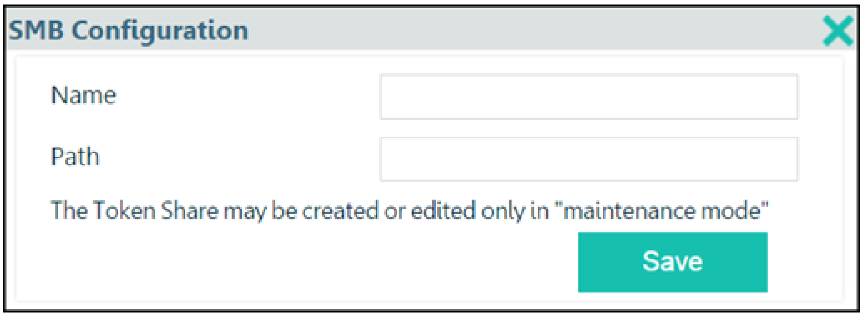
Save

Optionally, you can change the service’s name (click ✎) as it will appear anywhere in TSOC (in recorded events, and for emulation trap proxy configuration). It does not affect actual service configuration on the trap host computer.

Click **Save**.

If you don’t see a particular service, make sure the service is active on the trap host computer.

For the SMB service, click **Configure** to provide TSOC with details of a share folder existing on the full OS trap. If you selected CryptoTrap at Full OS installation, make sure to here provide the CryptoTrap share's location (by default - **C:\CT**):



SMB Configuration

Name

Path

The Token Share may be created or edited only in "maintenance mode"

Save

These details will be used in the SMB network share deception token (see Deception Token Types and Availability on page 45). For deception realism, save relevant spin data in the shared folder (not necessary with CryptoTrap, in which case the folder will be automatically populated).

Configuring Emulation Trap Proxy to Full OS Trap

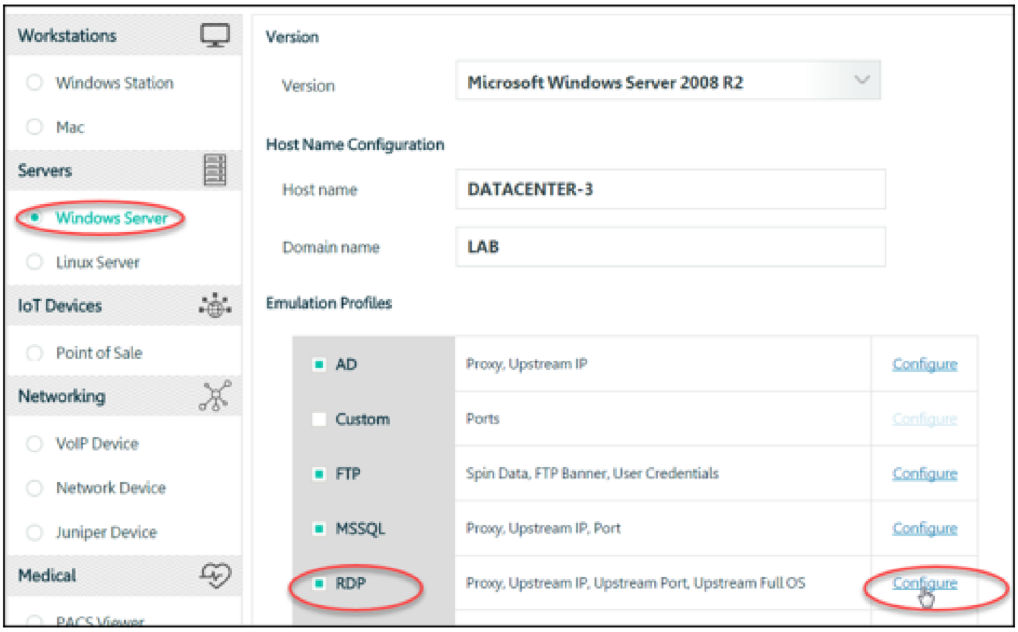
You can use a Full OS trap to transparently provide a real service to respond to attackers of emulation traps, and full monitoring of those attacks. This is achieved by proxying emulation traps' services to a Full OS trap. For example, you can have multiple emulation traps proxy Remote Desktop (RDP) sessions to a full Windows Server trap. Attackers who connect to any of those targets will be provided with full real desktop experiences, and their activity will be fully recorded by the Full OS trap.

To configure an emulation trap to be a proxy (see Full OS Trap Overview on page 41) to a full OS trap:

1. Make sure you have a properly set up full OS trap as in the *DeceptionGrid Administration Guide*, and a properly set up Windows emulation trap (see Emulation

Trap Deployment on page 17). All components must be of the current DeceptionGrid release.

2. To edit the emulation trap configuration, in TSOC go to **Appliances** > Appliance > **Configuration** > **Traps**, and by the trap click 🖉.
3. Make sure the emulation is of a Windows OS, select the service to be proxied, and by it click **Configure**:



Version

Version: Microsoft Windows Server 2008 R2

Host Name Configuration

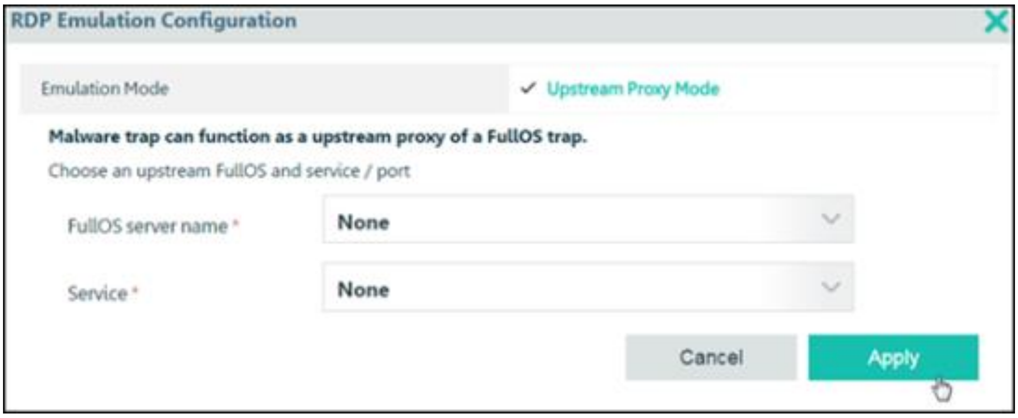
Host name: DATACENTER-3

Domain name: LAB

Emulation Profiles

☑ AD	Proxy, Upstream IP	Configure
□ Custom	Ports	Configure
☑ FTP	Spin Data, FTP Banner, User Credentials	Configure
☑ MSSQL	Proxy, Upstream IP, Port	Configure
☑ RDP	Proxy, Upstream IP, Upstream Port, Upstream Full OS	Configure

4. Select **Proxy Mode**, and the relevant **Full OS** trap. You can then select the exact **Service** (in case the full OS trap host has multiple services of the same type):



RDP Emulation Configuration

Emulation Mode: ✓ Upstream Proxy Mode

Malware trap can function as a upstream proxy of a FullOS trap.

Choose an upstream FullOS and service / port

FullOS server name *: None

Service *: None

5. Click **Apply**.

The proxy is configured.

Deception Token Deployment

In This Section

Deception Token Overview

Deception tokens are lures that are deployed across actual organizational endpoints, inside and outside the organizational network - Windows and Linux, servers and workstations. The tokens are various data items and configuration entries that point to DeceptionGrid emulation trap and full OS trap services, causing attackers that encounter the tokens to then attempt to connect to those services, triggering an alert. Deception tokens significantly add to the deception power of the traps, reducing the time it takes to detect attacks and defend against them.

Tokens can be designed and distributed so that multiple tokens across disparate endpoints create the illusion of the target traps being real and significant organizational assets. Agentless and light, deception tokens are easily deployable on servers and workstations.

The following topics describe the available tokens (see Deception Token Types and Availability below), and provide a high-level explanation of token deployment (see Deception Token Deployment: How it Works on page 47).

Before deploying traps, plan your deployment (see Planning Your DeceptionGrid Deployment on page 6).

In This Section

Deception Token Types and Availability

Token availability and actual deployment are dependent on trap type, its configured services, and endpoint environment. For example, ODBC tokens are available only for traps emulating Windows Server with SQL Server; and deployment of some types of browser tokens is dependent on specific endpoint and browser factors such as some antiviruses, existing browser usage, and trap connection availability.

Tokens can be deployed to the following endpoint types:

- Windows 7 and above

- Windows Server 2008 R2 and above
- VDI-based Windows endpoints of the above versions running desktops (persistent or non-persistent) in single-user mode (multi-user modes not supported)
- Unix-based endpoints with Bash (for tokens specified below for Linux)
- macOS 10.11 and above (for tokens specified below for Mac)

The following types of deception tokens are available for placement on organizational endpoints:

- **SMB Network Share**: Places a mapped drive with SMB connection details and credentials to an emulated file share on a DeceptionGrid trap, representing the trap as a Windows file server with a shared file directory.

  One random credential set from those uploaded to the trap's service (see Emulation Service Configuration on page 30) will be distributed with the token. If no credential set was uploaded, and SMB signing is configured for the appliance as recommended (see the *DeceptionGrid Administration Guide*), to connect to the trap's SMB service the token will use the credentials of the logged-on user.

  Note: On endpoints with non-administrative credentials, and on endpoints on which the Cached Credentials token is installed, the SMB share will be unmapped (even if here configured to have a drive letter), and therefore non-persistent through logoff unless a logon script is configured to install tokens upon every login.

- **ODBC Connection**: Places ODBC connection details to a DeceptionGrid trap, representing the trap as a Windows Server with SQL Server.
- **RDP**: Places Remote Desktop connection details to a DeceptionGrid trap in Remote Desktop configuration file and in registry key for servers, representing the trap as a Windows Server with Remote Desktop server.
- **Oracle**: Places connection details to the Oracle service on a DeceptionGrid trap. The port number is configurable.
- **SSH / PuTTY**: Places saved PuTTY session details (on Windows endpoints) or SSH keys (on Linux or Mac endpoints) representing an SSH connection to an emulation trap with SSH service. Deployable to Windows, Linux, and Mac endpoints.
- **WinSCP Session**: Places saved WinSCP session details of an SFTP connection to an emulation trap with SSH service. Deployable to Windows endpoints.
- **Hostname**: Places a hostname IP resolution record, with an optional inline comment, of a DeceptionGrid trap. Deployable to Windows, Linux, and Mac endpoints.

  Note: The Hostname token can be installed only with administrative privileges (on Linux: as root).

- **Deceptive Files**: Places files that when opened (even after copied away) automatically attempt to connect to a trap, thus triggering a trap event (if the trap is reachable).

You provide custom organizationally-relevant files, and the distribution mechanism embeds the functionality in them. The triggered event is displayed in TSOC specifically as a deceptive file event, with details of the source user.

Currently supported: Word **.docx** and Excel **.xlsx** files, for Windows Server emulations (with Web emulated service), Full OS traps (with web service), Public Trap (with Web emulated service), and Remote traps (with DeceptiveFileListener service). Deployable to Windows endpoints; effective on endpoints with MS Word / Excel 2007 or above.

- **Browser History** / **Credentials** / **Bookmark**: Places various types of browser-stored URL records of a DeceptionGrid trap as a visited site. **History** also places a browser cookie.

  It is recommended to educate users regarding these tokens, which are not hidden from users, to prevent false positives.

- **VPN Credentials**: Places a VPN Server trap's address with credentials in browser and in Windows.

- **Cached Credentials**: Places the trap's IP address with specified credentials in endpoint Windows Credential Manager (Vault). The token can be configured for any of several Windows emulation types (regardless of emulated services), and is deployable to Windows endpoints.

  The credentials should be designed to appear administrative and useful to an attacker. Traps' emulation services will not specifically authorize these credentials or recognize them as deceptive; if an attacker follows the recorded IP address, an event will be recorded as for any connection.

- **History**: Places the record of a ping command to the trap in the endpoint's command history. Deployable to Linux and Mac endpoints.

- **WinRM**: Registers the trap in endpoint TrustedHosts. Deployable to Windows endpoints.

- **Active Directory**: Places computer records of DeceptionGrid traps on organizational domain controllers.

Deception Token Deployment: How it Works

For each configured DeceptionGrid trap, deception tokens are automatically configured according to trap type and details, available services (emulated or real) and their configuration, and with recommended defaults. Changes to trap configuration automatically affect the trap’s tokens (for subsequent distributions to endpoints). Optionally, you can further configure some token settings.

As preparation for token distribution to endpoints, configure lists of trap tokens called Campaigns; at token distribution, you’ll specify one or more campaigns to be distributed to each set of endpoints. Each configured campaign should represent a type of organizational endpoint, to which relevant tokens can be assigned.

You then use one of the provided distribution methods to distribute tokens of specified campaigns to relevant endpoints. All distribution methods distribute a signed (for Windows), non-persistent installer that uses administrative credentials to add specified campaigns' tokens to endpoints, and in most cases hides them from legitimate endpoint UI users, leaving them visible to attackers using tools such as Mimikatz, net use, and ODBC Manager. Browser tokens are not hidden.

At distribution, all existing tokens that are not included in the new distribution are removed.

Follow this high-level procedure:

1. Plan and configure campaigns (see Configuring Token Campaigns below).
2. Configure deception tokens (see Configuring Deception Tokens on page 50).
3. Distribute tokens to organizational endpoints (see Distributing Deception Tokens to Endpoints on page 52).

Configuring Token Campaigns

As preparation for token distribution to endpoints, configure lists of trap tokens called Campaigns; at token distribution, you'll specify one or more campaigns to be distributed to each set of endpoints. Each configured campaign should represent a type of organizational endpoint, to which relevant tokens can be assigned.

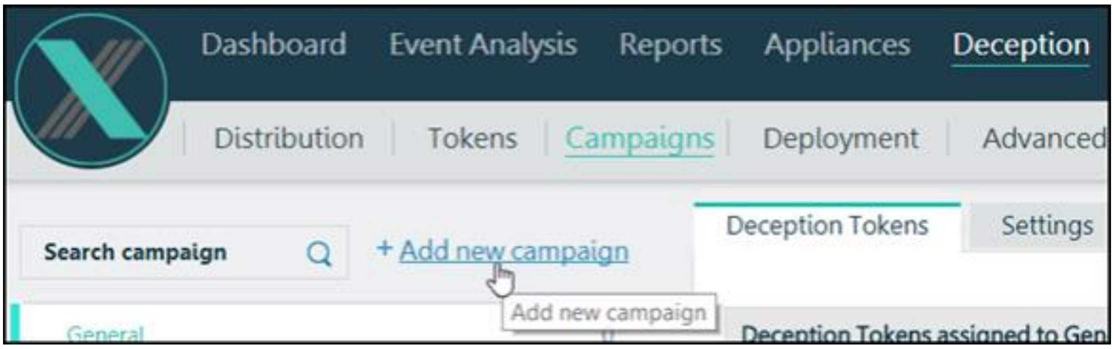
For example, a **Workstations** campaign might include the Hostname tokens of traps in that department's network and SMB Network Share and Browser tokens of traps meant to look like relevant organizational resource servers; A **Servers** campaign would include tokens of the types that would typically exist on servers. Your configured list of campaigns should reflect your organization's different departments and endpoint types; as you configure traps and tokens you'll assign tokens as relevant.

Each campaign belongs to a specific company or department, and only tokens of that company's or department's traps can be assigned to the campaign. So, for example, you would configure a workstations campaign for each of your departments (for example: Workstations-finance, Workstations-RnD, etc.), and assign to each campaign tokens of its department's traps.

Campaigns are unified (OR) at distribution, meaning all tokens of all specified campaigns are installed. Existing tokens from previous distributions are removed if not included in the new distribution.

The **General** campaign is preconfigured and non-deletable; you can define its department and tokens.

To create a new campaign, in TSOC go to **Deception** > **Campaigns** and click **Add new campaign**:

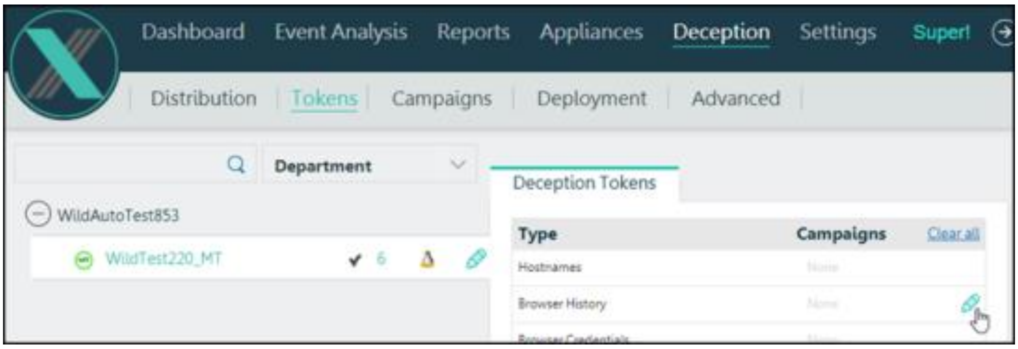


To subsequently edit an existing campaign's description, select the campaign and go to **Settings**.

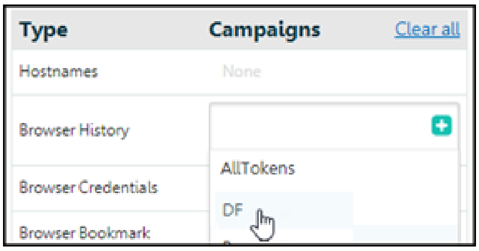
You can manage token assignment to campaigns during trap configuration (see Configuring Emulation Traps on page 24); or, subsequently – by trap and token, or by campaign.

Manage by trap and token

1. Go to **Deception** > **Tokens**, select a trap, hover over a token's row and click ✎:



2. Click ⊞, and select campaigns:

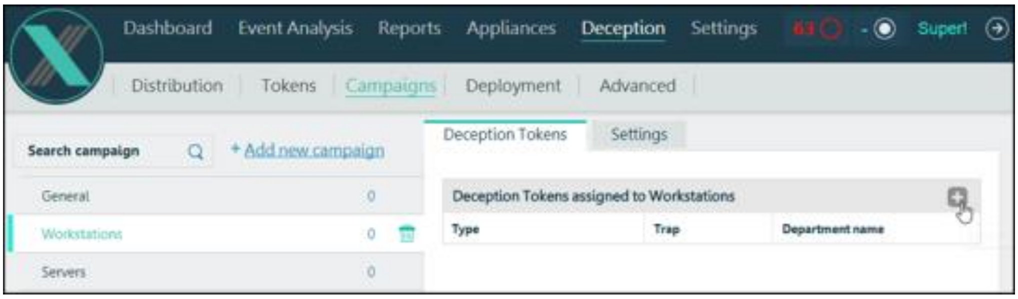


Optionally, you can **Assign all tokens** to a campaign.

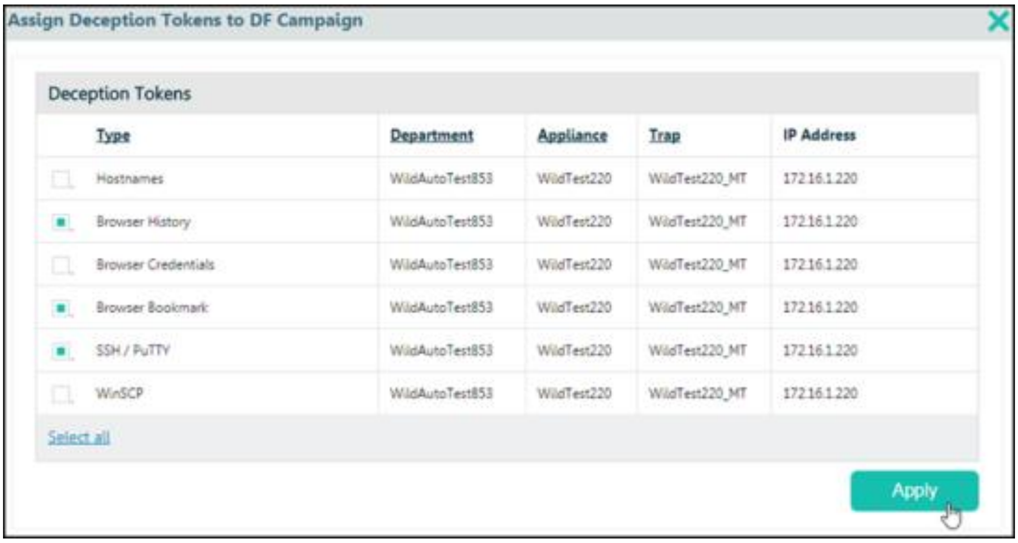
Save.

Manage by campaign

1. Go to **Deception** > **Campaigns**, select a campaign and click ⊞:



2. The company's or department's tokens are listed by token type and trap details. Select one or more tokens (or: **Select all**), and click **Apply**:



Assign Deception Tokens to DF Campaign

Deception Tokens

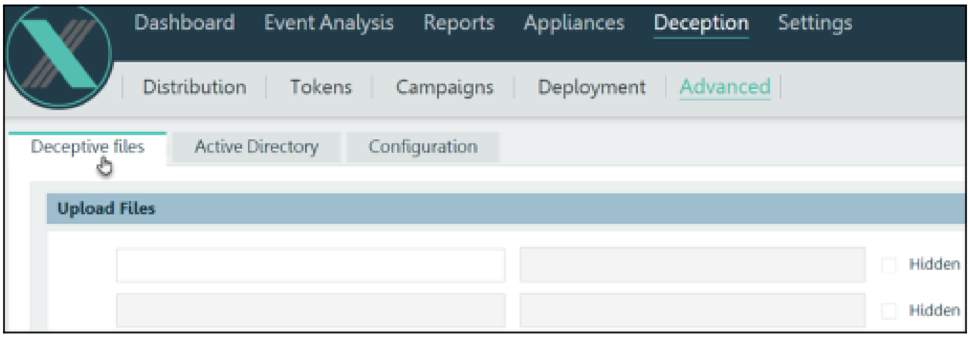
	Type	Department	Appliance	Trap	IP Address
□	Hostnames	WildAutoTest853	WildTest220	WildTest220_MT	172.16.1.220
■	Browser History	WildAutoTest853	WildTest220	WildTest220_MT	172.16.1.220
□	Browser Credentials	WildAutoTest853	WildTest220	WildTest220_MT	172.16.1.220
■	Browser Bookmark	WildAutoTest853	WildTest220	WildTest220_MT	172.16.1.220
■	SSH / PuTTY	WildAutoTest853	WildTest220	WildTest220_MT	172.16.1.220
□	WinSCP	WildAutoTest853	WildTest220	WildTest220_MT	172.16.1.220

Configuring Deception Tokens

For each configured DeceptionGrid trap, deception tokens are automatically configured according to trap type and details, available services (emulated or real) and their configuration, and with recommended defaults. Changes to trap configuration automatically affect the trap's tokens (for subsequent distributions to endpoints). Optionally, you can further configure some token settings. Some configuration is required to be able to deploy the Cached Credentials and Deceptive Files tokens.

To configure deception tokens:

1. If you'll be distributing **Cached Credentials** (see Deception Token Types and Availability on page 45) (not Browser Credentials) tokens for any traps, it is recommended to perform additional supportive configuration (see Supporting Cached Credentials Tokens on page 71).
2. If you intend to distribute the **Deceptive Files** (see Deception Token Types and Availability on page 45) token for any trap, provide files to be configured as decoys for the token (effective for all traps):
   a. Go to **Deception** > **Advanced** > **Deceptive files**:



   b. In the left column, upload .docx (not .doc!) Word decoy files. Maximum allowed file size is 1 MB.

For each uploaded file, in the right-hand column specify the location path in which to place the files on endpoints. Paths can include environment variables such as %USERPROFILE% , and they will resolve according to endpoint local environment. Optionally, select to set the uploaded file's properties attribute to **Hidden**.

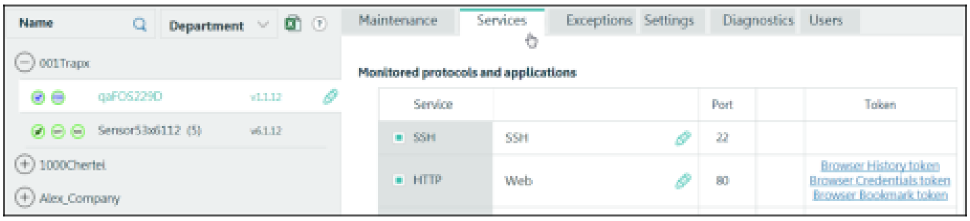
3. Make sure that each relevant trap is fully configured and connected to the relevant network.

Note: Most tokens use traps' IP addresses. If a trap uses DHCP, outdated tokens may be rendered ineffective.

4. From this step onward, these are optional per-trap configuration tasks that can be performed during trap creation (see Configuring Emulation on page 26) or from trap configuration pages as here.

   In TSOC, do one of the following:

   - For a full OS trap, go to **Appliances** > trap > **Services**:

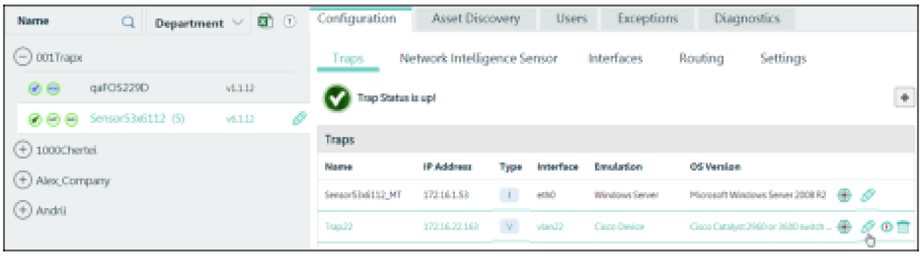


Monitored protocols and applications

Service		Port	Token
SSH	SSH	22	
HTTP	Web	80	Browser History token Browser Credentials token Browser Bookmark token

   - For an emulation trap:

     i. Go to **Appliances** > Appliance> **Configuration** > **Traps**, and in the trap's row click 🖉:

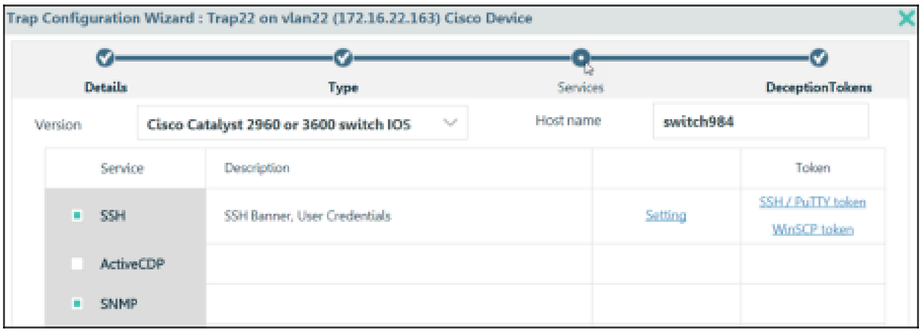


Trap Status is up!

Traps

Name	IP Address	Type	Interface	Emulation	OS Version
Sensor53x6112_MT	172.16.1.53	I	eth0	Windows Server	Microsoft Windows Server 2008 R2
Trap22	172.16.22.163	V	vlan22	Cisco Device	Cisco Catalyst 2960 or 3600 switch ...

     ii. Go to the **Services** page:


Trap Configuration Wizard : Trap22 on vlan22 (172.16.22.163) Cisco Device
Details
Type
Services
DeceptionTokens

Version: Cisco Catalyst 2960 or 3600 switch IOS
Host name: switch984

Service	Description		Token
SSH	SSH Banner, User Credentials	Setting	SSH / PuTTY token WinSCP token
ActiveCDP			
SNMP			

5. In the **Token** column, click the relevant token name and configure available settings.

   In general, the configurable settings are variable details that would typically be saved on endpoints (trap and service details such as the trap's IP address and a service's valid credentials are automatically added as relevant). For example, for

SMB Network Share – you can configure the share's mapped drive letter; for ODBC – database names and usernames (to represent SQL Server authentication, for which usernames but not passwords are saved on client hosts).

Some non-trivial settings are:

- **Browser** tokens (see Deception Token Types and Availability on page 45) tokens:
    - **Profile**: The browser user account under which to create the token. If the user account doesn't exist, it will be created. **Default** to use the browser's default user account.
    - **Maximum Version**: Deploy the token only if the browser version is not greater than the one specified. Should usually be left **0** to deploy to all versions.
- **Hostname** token (see Deception Token Types and Availability on page 45) **Comment**: Inline comment appended to hosts file entry. Use to describe the trap as a significant resource that will be attractive to attackers.
- **SMB Network Share** token **Drive Letter**: Selecting **Unmapped** will emulate the kind of automatic network share short-term mapping that occurs when a user connects to a network share without mapping a drive letter. This may reduce potential false positives, at slight risk of being ignored by ransomware or other attackers.

    Note: On endpoints with non-administrative credentials, and on endpoints on which the Cached Credentials token is installed, the SMB share will be unmapped (even if here configured to have a drive letter), and therefore non-persistent through logoff unless a logon script is configured to install tokens upon every login.

- **Cached Credentials**: Provide credentials to be stored in endpoints' Windows Credential Manager (Vault).

Click **Apply**, and **OK**.

Distributing Deception Tokens to Endpoints

For installing deception tokens on organizational endpoints (Windows and Linux), a signed (for Windows), non-persistent executable installer is provided. After installing tokens, the installer and its dependencies are automatically deleted.

Two distribution methods are available:

- **External distribution** (see External Distribution on page 53): Download from TSOC a ZIP archive including the installer and additional relevant files, and distribute to endpoints. You can do this distribution with configuration management systems such as GPO, SCCM, Chef, and Puppet. It is also possible to distribute tokens to non-domain endpoints (see Distribution to Non-Domain Endpoints on page 68).
- **TSOC distribution** (see TSOC Distribution on page 65) - not available in MSSP mode: Distribute tokens from TSOC on a fixed schedule and/or on demand, to endpoints selected from Asset Discovery or from the organizational Active Directory. An intuitive

wizard guides you through finding and selecting target endpoints and which token campaigns to distribute to each target.

This method is simplest to implement, and enables periodic distribution scheduling to keep endpoint tokens up-to-date with current trap and token configuration. However, it requires direct communication from TSOC to target endpoints and requires you to provide endpoint administrative credentials.

After initial deployment to target endpoints via TSOC distribution, you can update token deployment (see Updating Token Deployment on page 73). When distribution is managed externally, you can make changes only by repeated such distributions; to completely remove tokens from an endpoint, configure an empty campaign (see Configuring Token Campaigns on page 48) and distribute it.

Note: These distribution methods do not apply to the **Active Directory** token, which needs to be installed separately (see Installing Active Directory Tokens on page 69).

In This Section

External Distribution

External distribution is one of the two methods (see Distributing Deception Tokens to Endpoints on page 52) available for distributing deception tokens to endpoints. With external distribution, you download from TSOC a ZIP archive including the installer and additional relevant files, and distribute to endpoints. You can do this distribution with configuration management systems such as GPO, SCCM, Chef, and Puppet. It is also possible to distribute tokens to non-domain endpoints (see Distribution to Non-Domain Endpoints on page 68).

The downloaded archive includes a signed (for Windows), non-persistent executable installer. For Windows endpoints, the installer is: endpnt.exe ; for Linux - endpnt.bin ; for Mac - endpnt . The package also includes relevant encrypted data and configuration files that will be used by the installer. To refresh tokens, you can use a logon script to re-install upon every user login.

The installer can operate in one of two ways:

- **Connected installation** (recommended; requires connectivity from target endpoints to TSOC over port 6443): The installer will communicate with TSOC to obtain up-to-date token details and to report token installation status. This method enables changes that were made to trap details after package download to be reflected in tokens, and TSOC will be able to track and display token deployment status information (see Tracking Token Deployment on page 72); if token installation is implemented by a logon script, these will occur upon every user login.

- **Disconnected installation** (for endpoints that do not have connectivity to TSOC): The installer will take token details from a data file included in the package.

Note: The following distribution instructions do not apply to the **Active Directory** token, which needs to be installed separately (see Installing Active Directory Tokens on page 69).

The distribution procedure applies one or more specified campaigns; so, to distribute different campaigns to different endpoint sets you'll need to perform different distributions. For example, you might specify the **Workstations** campaign when distributing to your organization's relevant workstations, and then distribute the **Servers** campaign to relevant servers.

For devices that are used only connected to a VDI server such as Remote Desktop server or Citrix, distribute tokens to the server, not to the connecting devices.

The following is the standard procedure for distributing a specified set of token campaigns to an endpoint. Additional options appear below, under **Installation Notes**.

To distribute deception tokens:

1. Make sure you've properly configured tokens (see Configuring Deception Tokens on page 50), and you've assigned them to campaigns (see Configuring Token Campaigns on page 48).
2. Confirm the following target endpoint requirements:
   - Some endpoint security products may block deception token distribution. In this case, configure the security product to trust (whitelist) the following processes, by name or by MD5. You can use standard tools to obtain the MD5s, but be aware that it may change upon TrapX product upgrades.
     - Windows endpoints:
       - **endpnt.exe**
       - **ffcred.exe**
       - **ffcred32.exe**
       - **ffurlhash.exe**
       - (For TSOC distribution) **clssrw.exe**
     - Linux endpoints: **endpnt.bin**
     - Mac endpoints: **endpnt**
   - Windows endpoints the logged-on user must be able to run PowerShell scripts, and the PowerShell Language Mode must be: **FullLanguage** .
3. For connected installation:
   a. Make sure organizational firewalls allow port 6443 from target endpoints to TSOC.
   b. Make sure that the correct IP address at which token installations should access TSOC is defined in TSOC, at **Settings** > **General** > **IP**.
4. In TSOC, go to **Deception** > **Distribution** > **External Distribution** and click **Download**.

In MSSP mode, you'll be prompted to select a company. The resulting package will include only that company's API key, tokens, and campaigns.

Select one or more token campaigns to distribute. Complete the download, and extract the archive.

5. For disconnected installation (see above), you need to specify obtaining token details from the data file rather than from TSOC. In the distribution package open **endpnt_conf.json** and change:

   ```
   "use_file": "no",
   ```

   to:

   ```
   "use_file": "yes",
   ```

   Note: By default, the installer will expect the data file to be in its same location on the endpoint, with its original, unchanged name. Otherwise, here add also: `"data_file": "<full path>\<filename>",`

6. Optionally, for connected installation, for the installer to validate TSOC's certificate (if the certificate has been organizationally signed), in **endpnt_conf.json** set:

   ```
   "check_certificate": "yes"
   ```

7. Optionally, for improved obfuscation, change the name of the installer executable to something that could reasonably appear on organizational endpoints. For example, use the name of some legacy non-functioning organizational system. Make sure that endpoint security products still whitelist the installer.

8. The steps from here onward can be done manually, or implemented by automated systems (see Automated Distribution on page 56).

   Copy the following files to the endpoint:

   - Relevant installer executable: **endpnt.exe** (Windows), **endpnt.bin** (Linux), or **endpnt** (Mac)
   - **endpnt_conf.json**
   - For disconnected execution: **endpnt_data.json**
   - Entire **globals** directory with its contents (deceptive document files), if it exists.

   All of the above files must be at the same location on the endpoint (the contents of the **globals** directory should remain in it).

9. Run the installer executable: **endpnt.exe** (Windows), **endpnt.bin** (Linux), or **endpnt** (Mac).

   If the configuration file is not in the same location with the executable, or it has been renamed, append to the command:

   ```
   --config_file <full path>\<config-file>
   ```

   For example:

   ```
   Start /wait c:\endpnt.exe  config_file c:\My_edited_endpnt_c
   onf.json
   ```

If you encounter issues, troubleshooting is available (see Troubleshooting Token Distribution on page 70).

Installation Notes

Automated Distribution

Steps 8-9 above can be implemented by configuration management (CM) systems such as SCCM, Chef, Puppet, or (for Mac devices) Jamf. Or, if you don't have a configuration management system, you can use Group Policy (GPO; example below) for Windows domain endpoints.

The following are full procedure examples for SCCM, GPO, and Jamf:

Distribution Example: SCCM

To use SCCM, you'll place the files to be copied in an accessible network share, and have a batch file copy them from there to endpoints (to locations on which logged-on users will have execution permissions) and run the installer. SCCM will distribute a PowerShell script to Windows endpoints, which will in turn download and install the batch script in users' startup folders, so that upon each user login the token installer and files will be retrieved and activated.

This is part of the above general procedure for External Distribution as above. The following steps implement steps 8-9 above.

To distribute tokens with SCCM:

1. Place the files to be copied (see step 8 above) in an accessible network share. For example, in: \\%UserDNSDomain%\SysVol\MyDistribution .

   Note: If you'll be distributing different campaigns or campaign sets to different user groups, make sure to use separate network folders, so they don't overwrite each other.

2. Create a batch file (.bat) to copy the files from the above network share to endpoints and run the installer. For example:

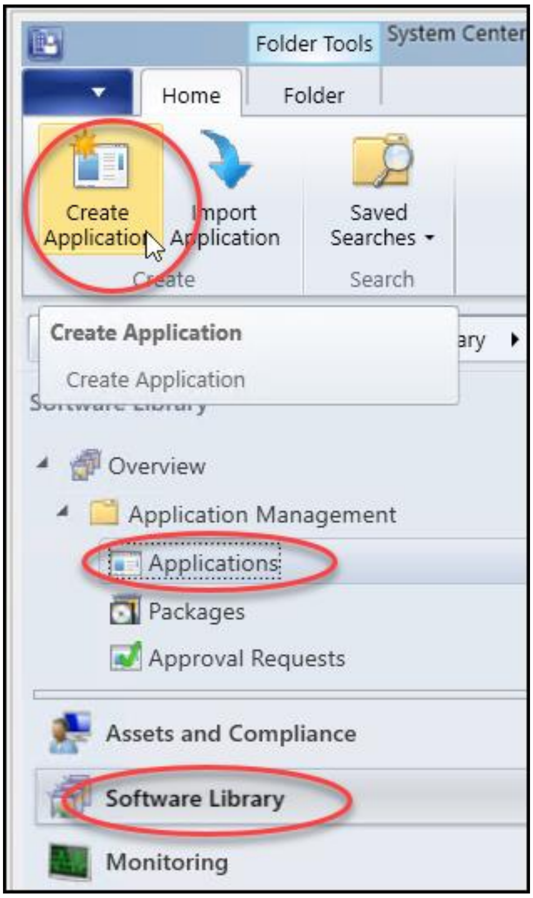
```
SET SourceFolder=\\%UserDNSDomain%\SysVol\MyDistribution
SET DestinationFolder=%UserProfile%
if exist "%DestinationFolder%\endpnt.exe" (
 REM "Do nothing, another tokens deployment is already running, lets avoid having multiple endpnt.exe process running."
) else (
 REM "Copy files over to destination folder and run installer"
xcopy /H /K /F /C /Y /I "%SourceFolder%\endpnt.exe" "%DestinationFolder%\endpnt.exe*"
xcopy /H /K /F /C /Y /I "%SourceFolder%\endpnt_conf.json" "%DestinationFolder%\endpnt_conf.json*"
xcopy /E /S /H /K /F /C /Y /I "%SourceFolder%\globals" "%DestinationFolder%\globals"
xcopy /H /K /F /C /Y /I "%SourceFolder%\endpnt_data.json" "%DestinationFolder%\endpnt_data.json*"
start /B /WAIT /MIN "Token" "%DestinationFolder%\endpnt.exe"
)
exit
```

3. In an accessible network share (can be same as token installer's location), place the batch script and a PowerShell script file (**.ps1**) with the following content, replacing the value of **$Source1** with the correct path to the batch file:

```
$Source1 =
"\\%UserDNSDomain%\SysVol\MyDistribution\endpnt.bat"
$destination1 =
"C:\Users\*\AppData\Roaming\Microsoft\Windows\Start
Menu\Programs\Startup"
$items = Get-ChildItem -Path $destination1
Write-Verbose "Number of items: $($items.Count)" -Verbose
foreach ($item in $items)
{
Write-Verbose "Item: $item" -Verbose
Copy-Item -Path $Source1 -Destination $item -Force -Verbose
}
```

4. Make sure that on endpoints to which tokens will be deployed, the PowerShell ExecutionPolicy allows the PowerShell script to run.

5. In the SCCM **Software Library**, go to **Applications** and click **Create Application**:



6. Go through the first few pages of the **Create Application** wizard with the following page settings:

   General: Select **Manually specify...**.

   General Information: Just give the application a **Name**.

   Application Catalog: Leave as is.

7. In the **Deployment Types** page, click **Add**, and go through the **Create Deployment Type** wizard with the following page settings:

   General: By **Type** select **Script Installer**.

   General Information: Just give the application a **Name**.

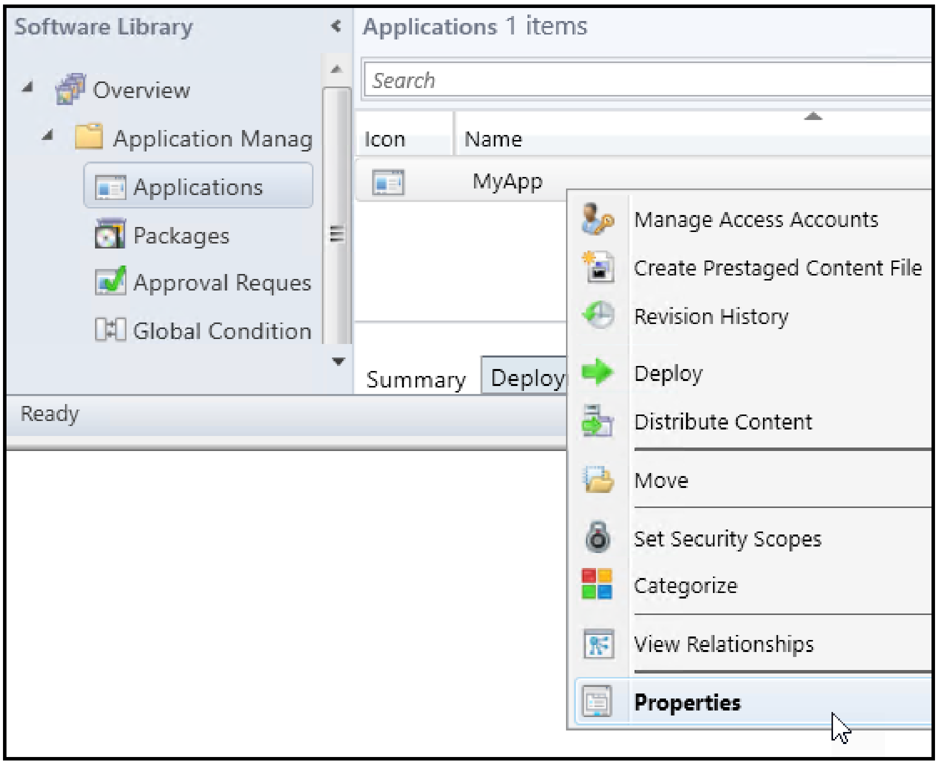
   Content: By **Content location**, **Browse** to the PowerShell file's location; by **Installation program**, **Browse** to (you may need to select to view **All files**) the PowerShell file.

   Detection Method: Click **Add Clause**, and in the **Detection Rule** window, by **Path** enter: **%AppData%\Microsoft\Windows\Start Menu\Programs\Startup** , and by **File or folder name** enter the batch file name. Click **OK**.

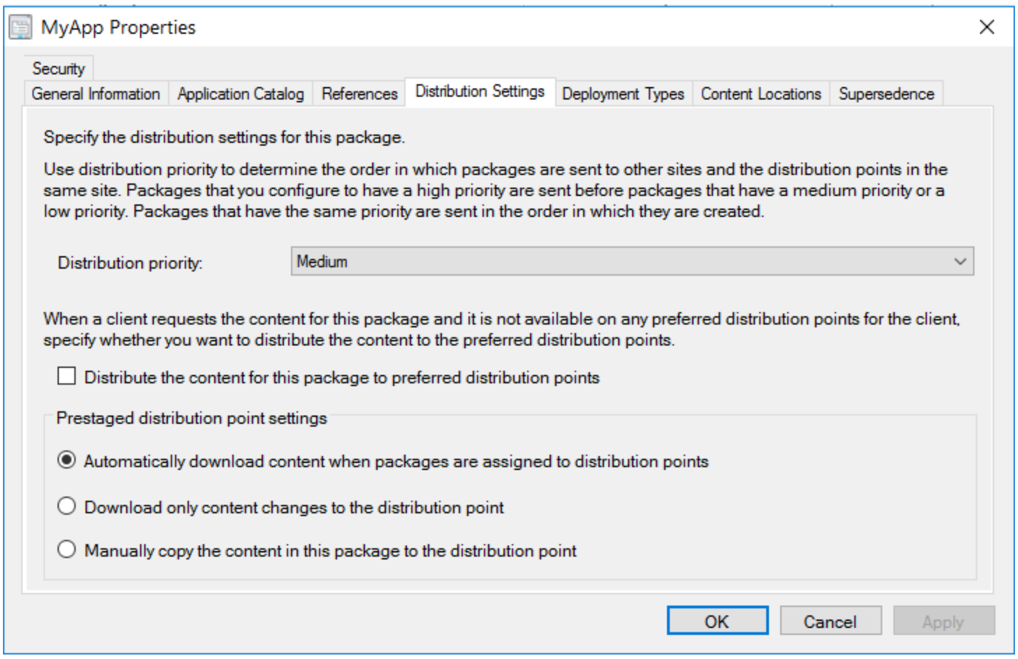
   User Experience: By **Installation program visibility** select **Hidden**.

8. Complete both wizards with no further changes.

9. Right-click the application you just created and select **Properties**:



In the **Distribution Settings** tab, select **Automatically download...** and click **OK**:



MyApp Properties

Specify the distribution settings for this package.

Use distribution priority to determine the order in which packages are sent to other sites and the distribution points in the same site. Packages that you configure to have a high priority are sent before packages that have a medium priority or a low priority. Packages that have the same priority are sent in the order in which they are created.

Distribution priority: Medium

When a client requests the content for this package and it is not available on any preferred distribution points for the client, specify whether you want to distribute the content to the preferred distribution points.

☐ Distribute the content for this package to preferred distribution points

Prestaged distribution point settings

● Automatically download content when packages are assigned to distribution points

○ Download only content changes to the distribution point

○ Manually copy the content in this package to the distribution point

10. Distribute the application to distribution points: Right-click the application and select **Distribute Content**. Go through the **Distribute Content** wizard, and in the **Content Destination** page click **Add** and select the relevant distribution points, groups, and collections. Complete the wizard.
11. Deploy the application to endpoints: Right-click the application and select **Deploy**. Go through the **Deploy Software** wizard with the following page settings:

    General: By **Collection**, **Browse** to relevant target users and groups.

    Content: Optionally add distribution points.

    Deployment Settings: By **Purpose** select **Required**.

    Scheduling: Optionally set a deployment time.

    User Experience: By **User notifications** select **Hide in Software Center and all notifications**.

    Complete the wizard.

Distribution Example: GPO

In an Active Directory environment, you'll place the files to be copied in an accessible network share, and use GPO to distribute a batch script as a logon script. The batch script will retrieve the token installer and files upon each logon, and run the installer.

This is part of the above general procedure for External Distribution as above. The following steps implement steps 8-9 above.

To distribute tokens with GPO:

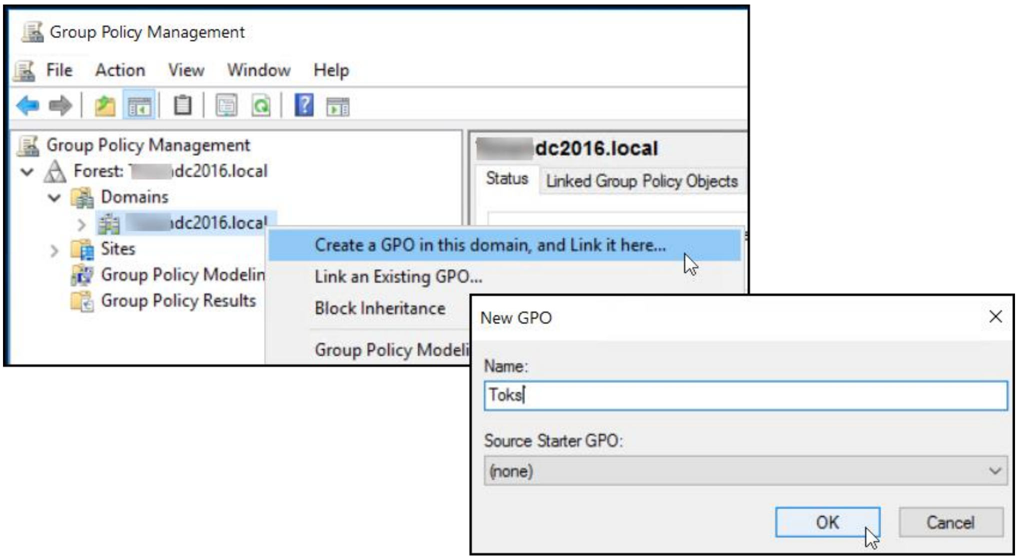
1. Place the files to be copied (see step 8 above) in an accessible network share. For example, in: \\%UserDNSDomain%\SysVol\MyDistribution .

Note: If you'll be distributing different campaigns or campaign sets to different user groups, make sure to use separate network folders, so they don't overwrite each other.

2. Create a batch file (.bat) to copy the files from the above network share to endpoints and run the installer. For example:

```
SET SourceFolder=\\%UserDNSDomain%\SysVol\MyDistribution
SET DestinationFolder=%UserProfile%
if exist "%DestinationFolder%\endpnt.exe" (
 REM "Do nothing, another tokens deployment is already running, lets avoid having multiple endpnt.exe process running."
) else (
 REM "Copy files over to destination folder and run installer"
xcopy /H /K /F /C /Y /I "%SourceFolder%\endpnt.exe" "%DestinationFolder%\endpnt.exe*"
xcopy /H /K /F /C /Y /I "%SourceFolder%\endpnt_conf.json" "%DestinationFolder%\endpnt_conf.json*"
xcopy /E /S /H /K /F /C /Y /I "%SourceFolder%\globals" "%DestinationFolder%\globals"
xcopy /H /K /F /C /Y /I "%SourceFolder%\endpnt_data.json" "%DestinationFolder%\endpnt_data.json*"
start /B /WAIT /MIN "Token" "%DestinationFolder%\endpnt.exe"
)
exit
```

3. Create a Group Policy Object: On the Active Directory domain controller, in **Group Policy Management** (run: gpmc.msc) right-click the top-level domain (or another appropriate level to apply to relevant users) and select **Create a GPO...** . **Name** the GPO (don't make it too descriptive, in case an attacker sees it) and click **OK**:

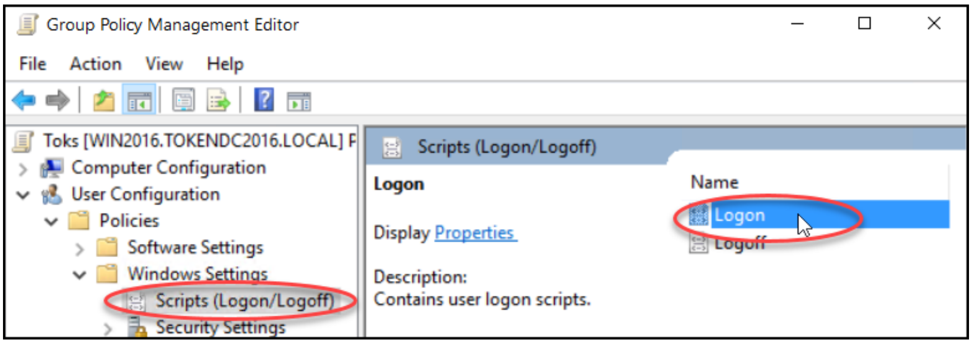


Name: Toks

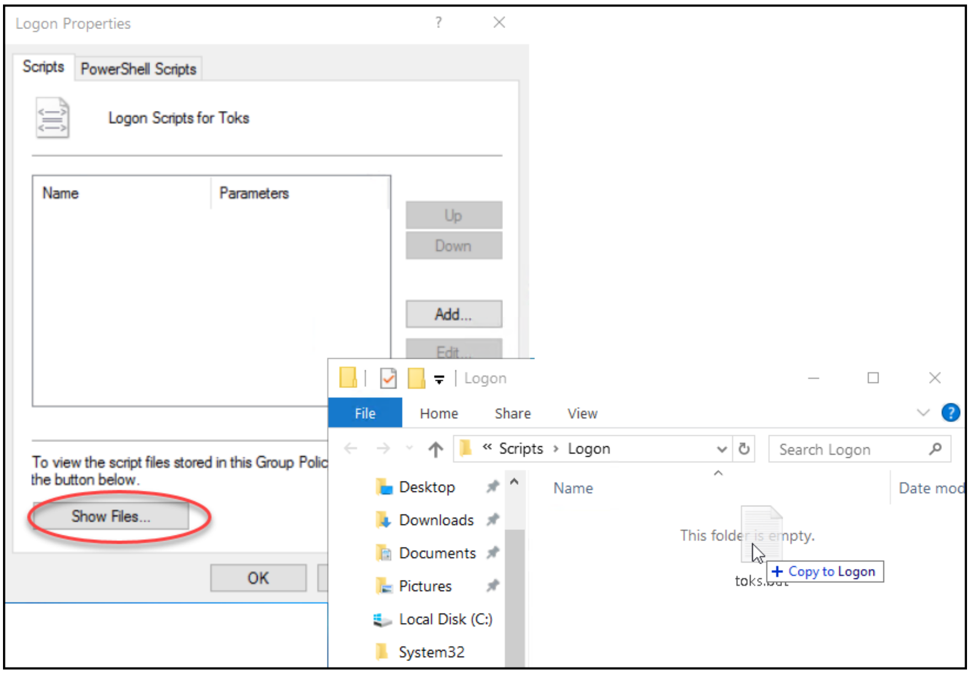
Source Starter GPO: (none)

4. Add the batch script:

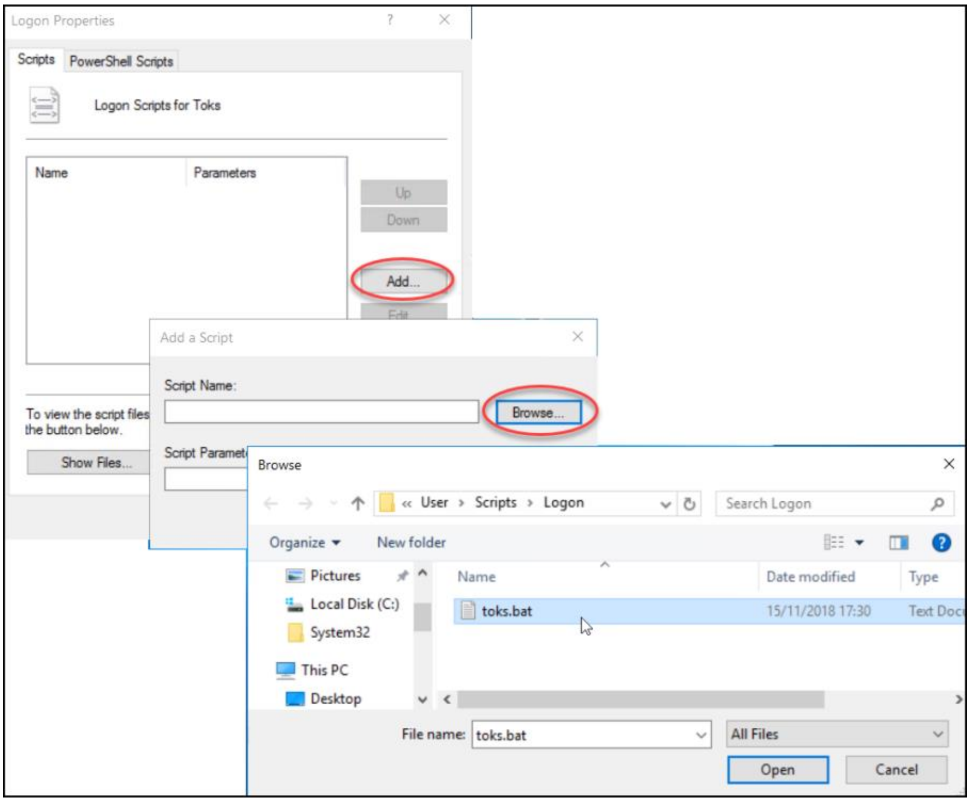
   a. Right-click the new GPO and select **Edit**. In the Editor, go to **User Configuration** > **Policies** > **Windows Settings** > **Scripts**, and double-click **Logon**:



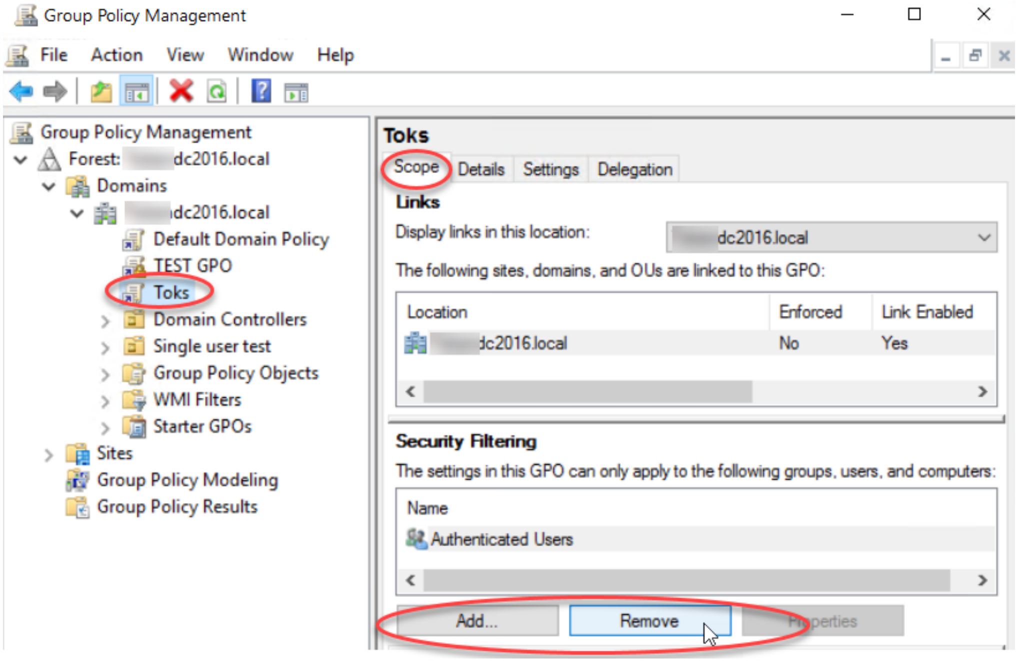
b. In the **Logon Properties** window, click **Show Files**, and copy the batch script to the file explorer window that appears:



c. Back in the **Logon Properties** window, click **Add**, **Browse** to the batch file and click **Open**, **OK**, **OK**:



5. If the GPO should apply only to some user groups, back in **Group Policy Management** (not in the Editor) select the GPO, and in the **Scope** tab, under **Security Filtering**, **Remove** the **Authenticated Users** group and **Add** the group(s) who should receive the batch script:



6. Right-click the GPO and select **Enforced**.

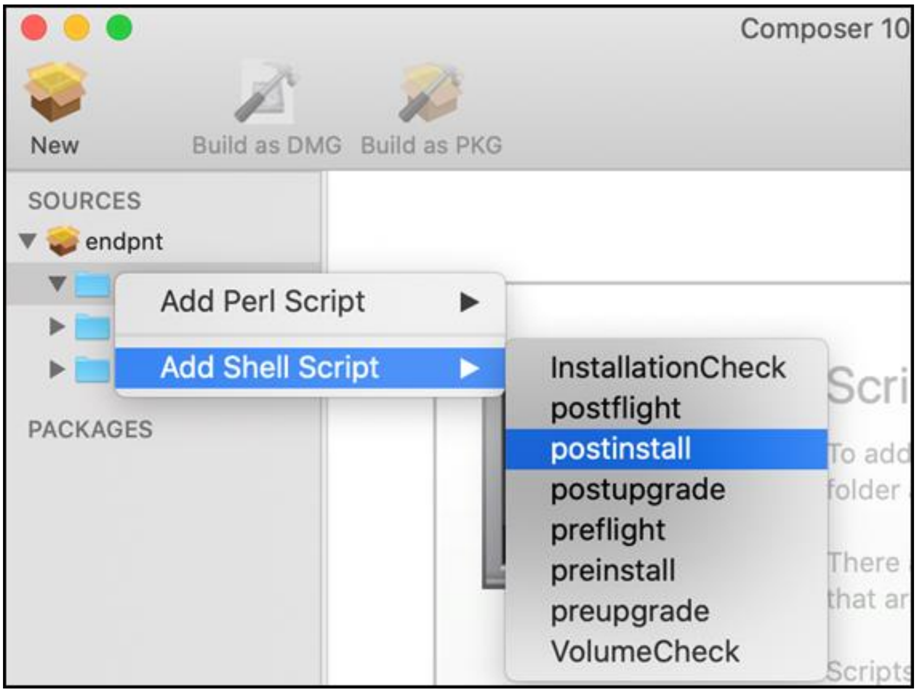
Distribution example: Jamf

To distribute tokens to Mac devices, you can use Jamf Composer to create a package, and then use Jamf Pro or Jamf Now to apply it to enrolled devices.

This is part of the above general procedure for External Distribution as above. The following steps implement steps 8-9 above.

To distribute tokens with Jamf:

1. Make sure you have Jamf Composer, appropriately configured for creating securely signed flat packages (in **Composer Preferences** > **Packaging**).
2. On the Jamf Composer host, place the files to be copied to endpoints (see step 8 above) in the **Applications** folder.
3. Drag the files from the Finder (Applications folder) into the Composer **Sources** list.
4. In the Sources list, Control-click or right-click **Scripts** and select **Add Shell Script** > **postinstall**:



5. From the downloaded token distribution package, copy the contents of **postinstall_jamf** into the Composer postinstall script.
6. Click **Build as PKG** and save the resulting package.
7. In Jamf Now, go to the relevant **Blueprint**, click **Add App** and upload the created package. Or, in Jamf Pro, use a policy to distribute the package.

Command-line arguments

In the above procedure, a configuration file is used to define the connection to TSOC (for connected execution), whether to use a data file (for disconnected execution), and campaigns whose tokens should be installed. As an alternative, you can define some or all of these as arguments in the execution command. Command-line arguments override settings in the configuration file; if all required settings are defined in the command, the configuration file is not needed.

- To specify campaigns, for each campaign whose tokens should be installed include:

```
--campaign <campaign>
```

- For connected execution:

  a. Include the following arguments with values copied from the configuration file:

  ```
  --tsoc_address <ip>
  --tsoc_port <port>
  --api_key <key>
  ```

  b. Optionally, for the installer to validate TSOC's certificate (if the certificate has been organizationally signed), include:

  ```
  --check_certificate yes
  ```

- For disconnected execution, include:

  ```
  --use_file yes
  ```

  or, if the data file is in a different location or has been renamed:

  ```
  --use_file yes --data_file <full path>\<filename>
  ```

For example:

```
C:\endpnt.exe tsoc_address 192.168.1.1 tsoc_port 8443 api_key
2ec3c65e-73c6-c9fC-b282-87ebe5837e2e campaign General –
campaign Department1 –campaign Department2
```

Advanced

When the installation executable runs on an endpoint, it searches for tokens existing on the endpoint, compiles a list of required tokens by specified campaigns, and accordingly creates PowerShell scripts to first remove existing tokens and then install the required ones. The installer runs these scripts and subsequently deletes them.

For advanced testing, troubleshooting, and customization purposes, you can have the token installation executable create and save the scripts without actually running them. The saved files include:

- **rm.ps1** : Removes existing tokens
- **add.ps1** : Installs required tokens
- **executables** (directory): Contains dependencies for the above PowerShell scripts
- **globals** (directory): Exists if deceptive document files have been configured; contains those files

To create and save the component files, run the installation executable (endpnt.exe , endpnt.bin , or endpnt) with the following argument:

```
--output_dir_no_execution <location>\
```

where <location> is the location where the files should be saved. For example:

```
C:\endpnt.exe --output_dir_no_execution C:\
```

To subsequently install tokens from those files, keeping them in their same locations relative to each other run first **rm.ps1** and then **add.ps1** .

Note: rm.ps1 will be hard-coded to remove only the specific tokens found on the endpoint by the installer prior to creating rm.ps1. Therefore do not use rm.ps1 on other endpoints, which may have other existing tokens.

TSOC Distribution

TSOC distribution (not available in MSSP mode) is one of the two methods (see Distributing Deception Tokens to Endpoints on page 52) available for distributing deception tokens to endpoints. With TSOC distribution, you distribute tokens from TSOC on a fixed schedule and/or on demand, to endpoints selected from Asset Discovery or from the organizational Active Directory. An intuitive wizard guides you through finding and selecting target endpoints and which token campaigns to distribute to each target.

Note: The following distribution instructions do not apply to the **Active Directory** token, which needs to be installed separately (see Installing Active Directory Tokens on page 69).

TSOC distribution requires that on target endpoints the logged-on user be connected directly to the endpoint, not via terminal service such as Remote Desktop. For devices that are used only connected to a VDI server such as Remote Desktop server or Citrix, use External distribution (see External Distribution on page 53).

To distribute deception tokens from TSOC:

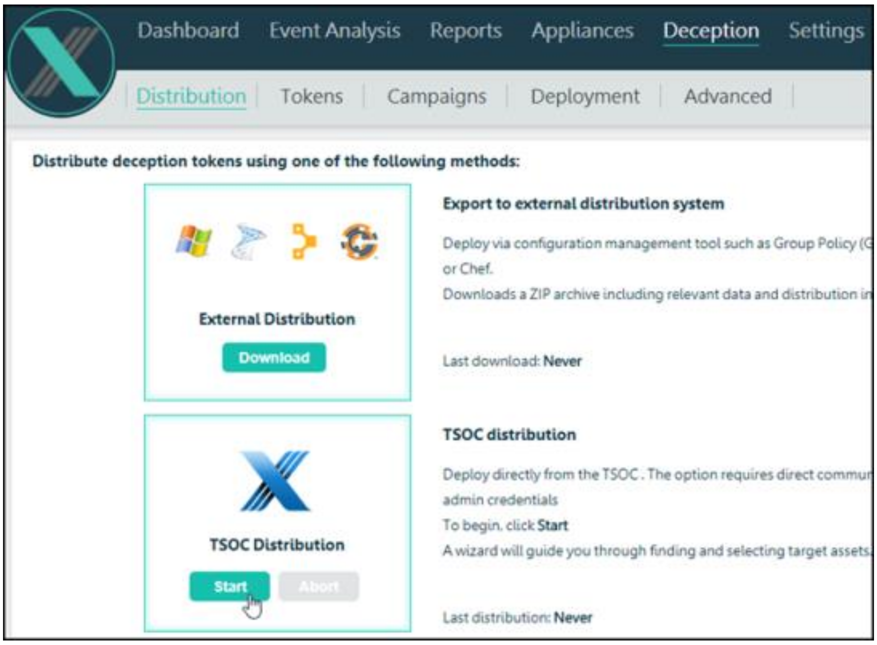
1. Make sure you've properly configured tokens (see Configuring Deception Tokens on page 50), and you've assigned them to campaigns (see Configuring Token Campaigns on page 48).
2. Make sure the following ports are allowed by organizational firewalls:

Source	***Destination***	***Port***	***Purpose***
TSOC	Target Windows endpoints	445	
		139	SMB/PSEXEC
		135	RPC
		Dynamic ports assigned by RPC for WMI	
	Target Linux endpoints	22 or custom	SSH/SFTP
Target endpoints	TSOC	6443	HTTPS

3. Confirm the following target endpoint requirements:
   - Some endpoint security products may block deception token distribution. In this case, configure the security product to trust (whitelist) the following processes, by name or by MD5. You can use standard tools to obtain the MD5s, but be aware that it may change upon TrapX product upgrades.
     - Windows endpoints:
       - **endpnt.exe**

        - **ffcred.exe**
        - **ffcred32.exe**
        - **ffurlhash.exe**
        - (For TSOC distribution) **clssrw.exe**
    - Linux endpoints: **endpnt.bin**
    - Mac endpoints: **endpnt**
- Windows endpoints the logged-on user must be able to run PowerShell scripts, and the PowerShell Language Mode must be: **FullLanguage** .

4. Make sure that target endpoints' antivirus systems do not block PSEXEC.
5. To be able to later manage token deployment, make sure that the correct IP address at which token installations should access TSOC is defined in TSOC, at **Settings** > **General** > **IP**.
6. In TSOC, go to **Deception** > **Distribution**, and in **TSOC Distribution** click **Start**:



Distribute deception tokens using one of the following methods:

Export to external distribution system

Deploy via configuration management tool such as Group Policy (G
or Chef.
Downloads a ZIP archive including relevant data and distribution in

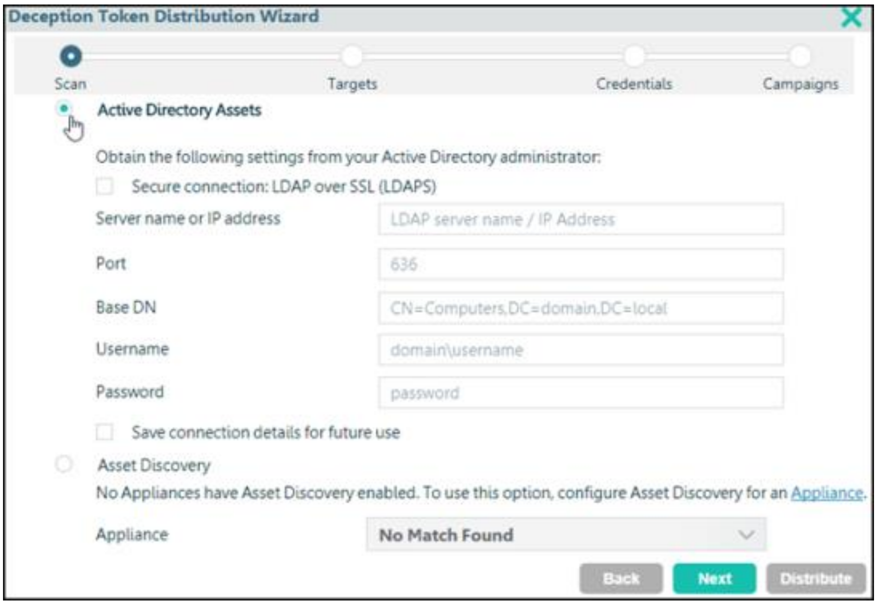
Last download: **Never**

TSOC distribution

Deploy directly from the TSOC . The option requires direct commun
admin credentials
To begin, click **Start**
A wizard will guide you through finding and selecting target assets

Last distribution: **Never**

7. In the distribution wizard's **Scan** page, select from where to obtain potential target endpoints: the organizational **Active Directory**, or **Asset Discovery**, and provide relevant settings:



Deception Token Distribution Wizard

● Active Directory Assets

Obtain the following settings from your Active Directory administrator:

□ Secure connection: LDAP over SSL (LDAPS)

Server name or IP address: LDAP server name / IP Address

Port: 636

Base DN: CN=Computers,DC=domain,DC=local

Username: domain\username

Password: password

□ Save connection details for future use

○ Asset Discovery

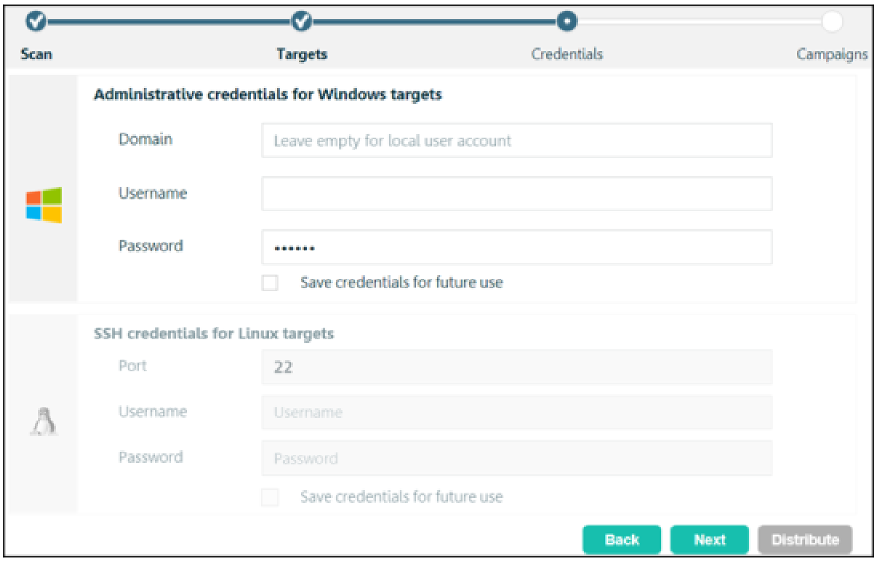
No Appliances have Asset Discovery enabled. To use this option, configure Asset Discovery for an Appliance.

Appliance: No Match Found

Click **Next**.

8. In the **Targets** page, use the search bar to filter the list of endpoints, and select target endpoints. Click **Next**.

9. In the **Credentials** page, provide Domain Administrator credentials:



Administrative credentials for Windows targets

Domain: Leave empty for local user account

Username:

Password: ••••••

□ Save credentials for future use

SSH credentials for Linux targets

Port: 22

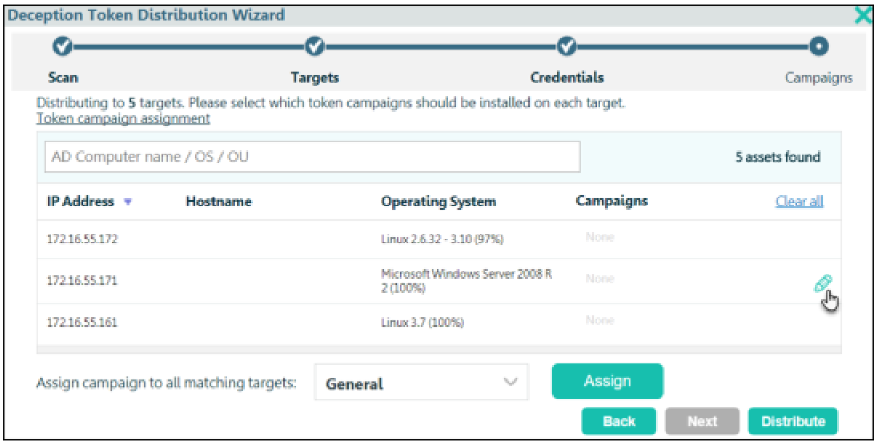
Username: Username

Password: Password

□ Save credentials for future use

If you previously selected any Linux targets, here provide the credentials of **root**, or of a sudoer that is configured with NOPASSWD.

10. In the **Campaigns** page, assign one or more token campaigns to each target endpoint. To assign to a single target, hover over the target row and click 🔗:



Distributing to **5** targets. Please select which token campaigns should be installed on each target.

AD Computer name / OS / OU

5 assets found

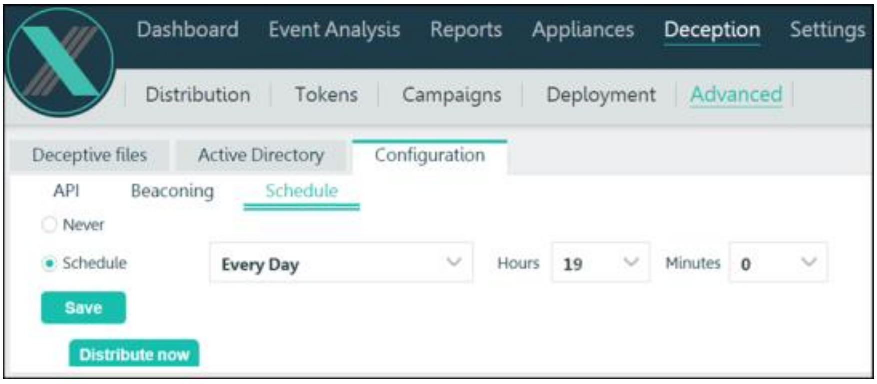
IP Address	Hostname	Operating System	Campaigns
172.16.55.172		Linux 2.6.32 - 3.10 (97%)	None
172.16.55.171		Microsoft Windows Server 2008 R 2 (100%)	None
172.16.55.161		Linux 3.7 (100%)	None

Assign campaign to all matching targets: General

To assign to multiple targets, use the search bar to filter the list, and then below select a campaign and click **Assign**.

For example, if you have **Servers** and **Workstations** campaigns, apply **Workstations** to all relevant workstations and apply **Servers** to all relevant servers.

11. Click **Distribute**.
12. To keep endpoint deception tokens up-to-date with current trap and token configuration, periodically distribute deception tokens according to configuration of last TSOC distribution. Go to **Deception** > **Advanced** > **Configuration** > **Schedule**, set a **Schedule** and/or **Distribute now**:



○ Never

● Schedule Every Day Hours 19 Minutes 0

Save.

If you encounter issues, troubleshooting is available (see Troubleshooting Token Distribution on page 70).

Distribution to Non-Domain Endpoints

To distribute tokens to endpoints outside the organizational domain (for example, tokens for Remote traps, to employees working from home), you can do either of the following (or both, to different endpoint groups):

- **VPN connection script**: For endpoints that connect to the organizational VPN, set up token installation to be run upon connecting to the VPN.

  Download and configure the token installation package as for regular External Distribution (see External Distribution on page 53) (Connected or Disconnected; until step 7), and configure a batch file (.bat file) that will copy the files to devices and then run **endpnt.exe** . Configure a VPN connection script to run the batch file upon connecting to VPN.

Example

The following are examples of a batch script to install tokens, and a VPN connection script to run the batch file.

This batch script is intended to run from a shared network location, where the source files are in the same location, as appropriate to be run from a VPN connection script.

```
SET SourceFolder=%~dp0
SET DestinationFolder=%UserProfile%
if exist "%DestinationFolder%\endpnt.exe" (
REM "Do nothing, another token deployment is already running")
else (
REM "Copy files over to destination folder and run installer"
xcopy /H /K /F /C /Y /I "%SourceFolder%\endpnt.exe" "%DestinationFolder%\endpnt.exe*"
xcopy /H /K /F /C /Y /I "%SourceFolder%\endpnt_conf.json" "%DestinationFolder%\endpnt_conf.json*"
xcopy /H /K /F /C /Y /I "%SourceFolder%\endpnt_data.json" "%DestinationFolder%\endpnt_data.json*"
xcopy /E /S /H /K /F /C /Y /I "%SourceFolder%\globals" "%DestinationFolder%\globals"
start /B /WAIT /MIN "Token" "%DestinationFolder%\endpnt.exe"
)
Exit
```

This VPN connection script is for Forticlient, to run the batch file upon connecting to VPN. You can use EMS to centrally configure users' Forticlient with the script.

```
<on_connect>
<script>
<os>windows</os>
<script><![CDATA[\\cyber.local\NETLOGON\TokenForVPN\Token-Deployment-Computers.bat]]></script>
</script>
</on_connect>
```

- **MDM**: Use an MDM (Mobile Device Management) or other external (not domain) distribution system. For this, TrapX can provide a single-file installer (MSI / EXE) upon special request. Use your distribution system to distribute and execute the installer as a Local Administrator.

Example

If you're using Microsoft Endpoint Manager (Formerly Intune):

1. Request a token installation MSI from TrapX support.

2. Go to **Apps** > **Add App**, and upload the MSI as the **App package file**.

3. In the App **Properties**, set **Command-line arguments** to: **/qn**

4. Assign the App to the relevant **Group**, and make the App **available for enrolled devices**.

The tokens will be installed upon device logon.

Note: Although the installer is automatically deleted after token installation, it remains listed in Windows Add/Remove Programs, which could be detected by attackers. To remove the listing, subsequently Uninstall the MSI.

Installing Active Directory Tokens

Active Directory deception tokens place computer records of DeceptionGrid traps on organizational domain controllers. Active Directory tokens are not installed by the distributed installer. Instead, they need to be manually installed.

Active Directory token installation requires some preconfiguration. Besides per-trap token configuration (see Configuring Deception Tokens on page 50), which for the Active Directory token includes only the option to change the trap's computer password from its default, you need to provide details of the Domain Controller where the token will be installed, and of the Active Directory object in which the trap computer objects should be recorded. For each Appliance whose traps should be recorded in Active Directory, you can provide this configuration in one of two ways:

- **SMB signing configuration** (recommended): For the Appliance, configure SMB Signing support (see the *DeceptionGrid Administration Guide*), and there select to use the information for external AD / SIEM configuration. This will apply to the Active Directory tokens of the Appliance's traps.

  This method makes the tokens stealthy, preventing detection by known trap detection tools.

- **Global AD token configuration**: In TSOC, go to **Deception** > **Advanced** > **Active Directory**, provide the details and click **Apply**. This will apply to the Active Directory tokens of traps of Appliances for which the first method above is **not** configured (that is, of Appliances for which **Use this information for AD tokens** is not selected, whether or not SMB signing is configured).

  Note that these tokens may be detected by known trap detection tools.

If the above configuration includes the domain controller's hostname and/or IP address (required for SMB signing configuration), the Active Directory token can be installed only remotely, from a Windows computer with access to the domain controller - not locally on the DC itself. If the configuration includes neither DC hostname nor DC IP address, the token can be installed only locally on the DC.

To install the Active Directory deception token:

1. According to the method(s) described above, configure DC and AD details.
2. Go to **Deception** > **Distribution** > **External Distribution**, and **Download** the archive.

If (and only if) the above configuration did not include the domain controller's hostname or IP address, copy the downloaded archive to the domain controller.

3. Extract the archive.
4. From the **external_distribution\externals\active_directory\** folder run **add_active_directory.ps1** .You will be prompted for Active Directory administrative credentials; the tokens are then installed.

Troubleshooting Token Distribution

- On some Windows endpoints, UAC may block remote execution. In this case, to disable UAC configure the following registry key with value **1**:

  HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies \System\LocalAccountTokenFilterPolicy

  If the key doesn't exist, create it as **DWORD 32 bit**.

- Some endpoint security products may block token distribution. In this case, configure the security product to trust (whitelist) the following processes, by name or by MD5. You can use standard tools to obtain the MD5s, but be aware that it may change upon TrapX product upgrades.
  - Windows endpoints:
    - **Token.exe**
    - **ffcred.exe**
    - **ffcred32.exe**
    - **ffurlhash.exe**
  - Linux endpoints: **Token.bin**

Supporting Cached Credentials Tokens

The Cached Credentials deception token (see Deception Token Types and Availability on page 45) places the trap's IP address with specified credentials in endpoints' Windows Credential Manager (Vault).

For optimal deception, the credentials you specify in the token configuration (see Configuring Deception Tokens on page 50) should be actually configured in the organizational Active Directory. And, to be alerted on any attempted use of the credentials (not just to connect to traps), your organizational SIEM should be configured to alert upon the credentials being used for any service in the domain. Tools for both these configurations are provided in the token distribution archive, preconfigured with the credentials defined in Cached Credentials tokens.

Configuring Active Directory with Cached Credentials token credentials

To configure your organizational Active Directory with the credentials specified in Cached Credential tokens (for each relevant Appliance):

1. In TSOC, go to **Appliances** > Appliance > **Configuration** > **Settings** > **Configure SMB domain**. Ensure that SMB domain is **Enabled** and properly configured, and select **Use this information for external AD / SIEM configuration**.
2. Go to **Deception** > **Distribution** > **External Distribution**, and **Download** the archive.

If (and only if) the above SMB configuration includes **neither** the domain controller's hostname nor its IP address, copy the downloaded archive to the domain controller.

3. Extract the archive.
4. From the **external_distribution\externals\** folder run **add_cached_credentials.ps1** .You will be prompted for Active Directory administrative credentials; the credentials are then configured.

Configuring a SIEM to alert on use of Cached Credentials token credentials

To configure your organizational SIEM to alert on use of token credentials:

1. In TSOC, go to **Appliances** > Appliance > **Configuration** > **Settings** > **Configure SMB domain**. Ensure that SMB domain is **Enabled** and properly configured, and select **Use this information for external AD / SIEM configuration**.
2. Go to **Deception** > **Distribution** > **External Distribution**, and **Download** and extract the archive.
3. In the archive, the following text file includes regular expression for identifying the token credentials:

   external_distribution\externals\cached_credentials\splunk_regex_for_cached_credentials_events_forwarding.txt

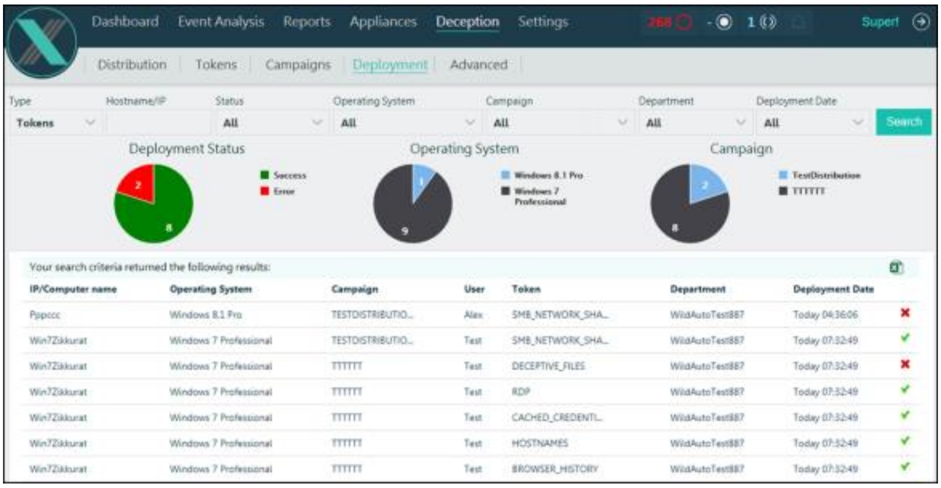
   The expression is in the appropriate format for Splunk; for other SIEMs some adaptation may be needed. For information on configuring the SIEM, see SIEM documentation.

Managing Token Deployment

In This Section

Tracking Token Deployment

Upon distribution by TSOC Distribution (see TSOC Distribution on page 65) or by connected execution (see External Distribution on page 53), tokens report to TSOC on installation status. The current results of that reporting are displayed in TSOC at **Deception** > **Deployment**:



Type: Tokens
Hostname/IP:
Status: All
Operating System: All
Campaign: All
Department: All
Deployment Date: All

Your search criteria returned the following results:

IP/Computer name	Operating System	Campaign	User	Token	Department	Deployment Date	
Pppccc	Windows 8.1 Pro	TESTDISTRIBUTIO...	Alex	SMB_NETWORK_SHA...	WildAutoTest887	Today 04:36:06	✗
Win7Zikkurat	Windows 7 Professional	TESTDISTRIBUTIO...	Test	SMB_NETWORK_SHA...	WildAutoTest887	Today 07:32:49	✓
Win7Zikkurat	Windows 7 Professional	TTTTTT	Test	DECEPTIVE_FILES	WildAutoTest887	Today 07:32:49	✗
Win7Zikkurat	Windows 7 Professional	TTTTTT	Test	RDP	WildAutoTest887	Today 07:32:49	✓
Win7Zikkurat	Windows 7 Professional	TTTTTT	Test	CACHED_CREDENTI...	WildAutoTest887	Today 07:32:49	✓
Win7Zikkurat	Windows 7 Professional	TTTTTT	Test	HOSTNAMES	WildAutoTest887	Today 07:32:49	✓
Win7Zikkurat	Windows 7 Professional	TTTTTT	Test	BROWSER_HISTORY	WildAutoTest887	Today 07:32:49	✓

Only recently-distributed tokens are displayed. After the number of days from distribution defined in **Deception** > **Advanced** > **Configuration** > **Beaconing**, tokens are removed from the list.

Select the display **Type**. **Tokens** means that pie chart numbers and table rows represent an individual tokens on individual endpoints; when **Assets** is selected, table rows represent endpoints (charts are not available).

You can filter the list of represented tokens and endpoints by various parameters.

To export the list to CSV, click .

If no data appears, verify the following:

- The tokens were distributed via TSOC distribution or connected execution.
- Organizational firewalls allow port 8443 (HTTPS) from endpoints to TSOC.

Note: Only the current state is displayed, not deployment history. So, each endpoint's last (recent) distribution is reflected.

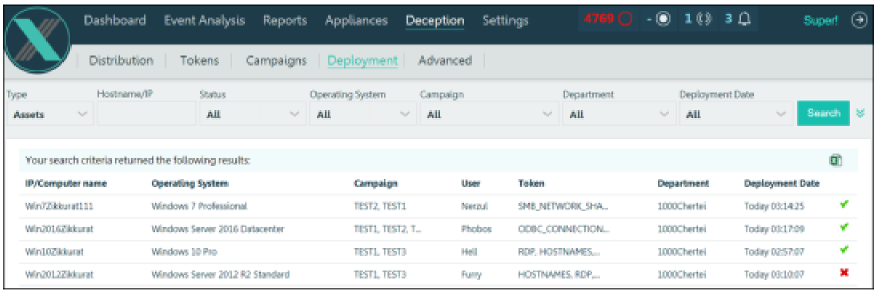
Updating Token Deployment

After initial token deployment to target endpoints via TSOC distribution (see TSOC Distribution on page 65), you can change those endpoints' deployed token campaigns. Upon each change, TSOC immediately redistributes tokens according to the updated campaign list, causing tokens to be added or removed as relevant.

These updates are relevant only for target endpoints to which a TSOC distribution has previously been performed, as long as connectivity and credentials are maintained. When distribution is managed externally (see External Distribution on page 53), you can make changes only by repeated such distributions; to completely remove tokens from an endpoint, configure an empty campaign (see Configuring Token Campaigns on page 48) and distribute it.

To updated endpoints' campaign lists:

1. In TSOC, go to **Deception** > **Deployment**, and by **Type** select **Assets**:



Type	Hostname/IP	Status	Operating System	Campaign	Department	Deployment Date
Assets		All	All	All	All	All

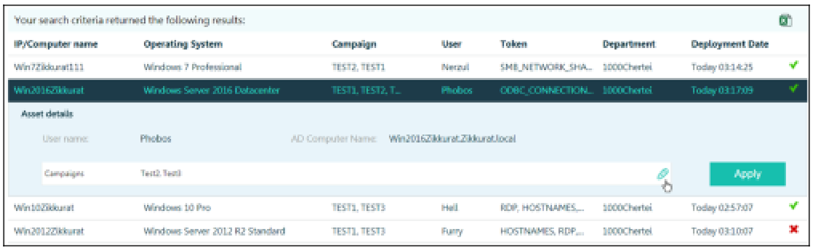
Your search criteria returned the following results:

IP/Computer name	Operating System	Campaign	User	Token	Department	Deployment Date	
Win7Zikkurat111	Windows 7 Professional	TEST2, TEST1	Nerzul	SMB_NETWORK_SHA...	1000Chertei	Today 03:14:25	✓
Win2016Zikkurat	Windows Server 2016 Datacenter	TEST1, TEST2, T...	Phobos	ODBC_CONNECTION...	1000Chertei	Today 03:17:09	✓
Win10Zikkurat	Windows 10 Pro	TEST1, TEST3	Hell	RDP, HOSTNAMES,...	1000Chertei	Today 02:57:07	✓
Win2012Zikkurat	Windows Server 2012 R2 Standard	TEST1, TEST3	Fury	HOSTNAMES, RDP,...	1000Chertei	Today 03:10:07	✗

Displayed rows now represent target endpoints.

2. Do one of the following:

- To update a **single** endpoint's campaign list, select the endpoint, and in its **Campaigns** list click 🖉:



Your search criteria returned the following results:

IP/Computer name	Operating System	Campaign	User	Token	Department	Deployment Date	
Win7Zikkurat111	Windows 7 Professional	TEST2, TEST1	Nerzul	SMB_NETWORK_SHA...	1000Chertei	Today 03:14:25	✓
Win2016Zikkurat	Windows Server 2016 Datacenter	TEST1, TEST2, T...	Phobos	ODBC_CONNECTION...	1000Chertei	Today 03:17:09	✓
Win10Zikkurat	Windows 10 Pro	TEST1, TEST3	Hell	RDP, HOSTNAMES,...	1000Chertei	Today 02:57:07	✓
Win2012Zikkurat	Windows Server 2012 R2 Standard	TEST1, TEST3	Fury	HOSTNAMES, RDP,...	1000Chertei	Today 03:10:07	✗

Asset details

User name: Phobos   AD Computer Name: Win2016Zikkurat.Zikkurat.local

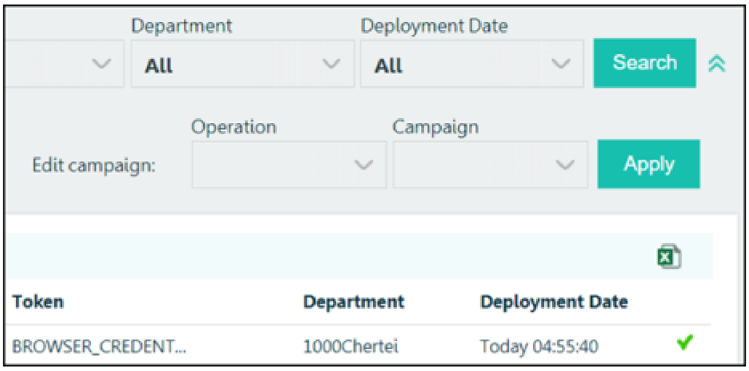
Campaigns: Test2, Test1

Add and/or remove campaigns, and click **Apply**.

- To add a campaign to or remove from multiple endpoints' campaign lists:

  i. Display only the relevant endpoints: set the search fields and click **Search**.

  ii. To the right of the Search bar, click ⩔. Additional fields appear:



Department: All   Deployment Date: All

Edit campaign: Operation: Campaign:

Token	Department	Deployment Date	
BROWSER_CREDENT...	1000Chertei	Today 04:55:40	✓

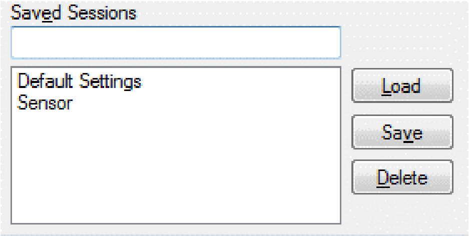
  iii. Select the **Operation** (**Add** / **Remove**) and a **Campaign**, and click **Apply**.

Testing Token Deployment

You can check tokens on endpoints as in the following sections.

Verifying SSH \ PuTTy token deployment

Open PuTTy and make sure the relevant session appears under Saved Sessions:



Saved Sessions

Default Settings
Sensor

Verifying Browser token deployment

- Verify **Bookmarks**: Open the Bookmarks toolbar / Favorites and check that the relevant title appears.
- Verify **History**: Open the browser's navigation history and check that the relevant title appears.
- Verify **Credentials**: Use the program NirSoft WebBrowserPassView (recommended) to check.

Verifying SMB token deployment

1. Run **cmd**
2. Enter:

```
net use
```

3. Check that \\<trapIP>**\data** is mapped.

Verifying ODBC token deployment

In **Control Panel** > **Administrative Tools**, open **Data Sources (ODBC)** and check that your display name appears there.

Verifying Hostname token deployment

Open **C:\Windows\System32\drivers\etc\hosts** (on Linux: **/etc/hosts**), and check that your host record appears.

Verifying Active Directory token deployment

1. On your Domain Controller, open **Active Directory Users and Computers** and check that the configured Active Directory objects are added.
2. Return the command **set-executionpolicy** to its default (typically, **set-executionpolicy unrestricted**).

Support

Support for TrapX products is provided by TrapX or by an authorized TrapX Service Partner. More information and technical support for TrapX products are available at:

- support.trapx.com/portal
- support@trapx.com
- Americas: 1-855-249-4453
  EMEA & Asia Pacific: +44-208-819-9849

Documentation Feedback

TrapX Security continually strives to produce high quality documentation. If you have any comments, please contact Documentation@trapx.com.

About TrapX Security®

TrapX Security is the pioneer and global leader in cyber deception technology, with flagship solution DeceptionGrid effectively detecting, deceiving, and defeating advanced cyber attacks and human attackers in real-time. DeceptionGrid provides automated, highly accurate insight into malicious activity unseen by other types of cyber defenses. Deploying DeceptionGrid sustains a proactive security posture, fundamentally halting the progression of an attack. DeceptionGrid changes cyber-attack economics by shifting the cost to the attacker.

The TrapX Security customer base includes worldwide Forbes Global 2000 commercial and government customers in key industries including defense, healthcare, finance, energy, and consumer products. Learn more at www.trapx.com .

Disclaimer

Product specifications are subject to change without notice. This document is believed to be accurate and reliable at the time of printing. However, due to ongoing product improvements and revisions, TrapX cannot guarantee accuracy of printed material after the Date Published nor can it accept responsibility for errors or omissions. Before consulting this document, check the corresponding Release Notes regarding feature preconditions and/or specific support in this release. In cases where there are discrepancies between this document and the Release Notes, the information in the Release Notes supersedes that in this document. Updates to this document and other documents as well as software files can be obtained by TrapX customers.

Trademarks and Copyright

Updated 17/6/21